UNITI

Introduction to java

Java is an Object-Oriented Language. As a languagethat hastheObject Orientedfeature,Java supports the following fundamental concepts:

- Polymorphism
- Inheritance
- Encapsulation
- Abstraction
- Classes
- Objects
- Instance
- Method
- MessageParsing

Inthischapter, wewilllookintotheconceptsClassesandObjects.

- **Object -** Objects have states and behaviors. Example: A dog has states - color, name, breedaswellasbehaviors-wagging,barking,eating.Anobject isaninstanceofaclass.
- **Class-**Aclasscanbedefinedasatemplate/blue printthatdescribesthebehaviors/states that object of its type support.

ObjectsinJava:

Let usnowlookdeepintowhat areobjects. Ifweconsiderthereal-worldwecanfind many objects around us, Cars, Dogs, Humans, etc. All these objects have a state and behavior.

Ifweconsideradog,thenitsstateis -name,breed,color, andthebehavior is-barking, wagging, running

If youcomparethesoftwareobject witharealworldobject,theyhaveverysimilar characteristics.

Softwareobjectsalso haveastateandbehavior. Asoftwareobject'sstateisstoredinfieldsand behavior is shown via methods.

So insoftwaredevelopment, methodsoperateonthe internalstateofanobject and theobject-to-object communication is done via methods.

ClassesinJava:

Aclass isablueprint fromwhichindividualobjectsarecreated. A

sample of a class is given below:

```
publicclassDog{
   String breed;int
   age;
   Stringcolor;

   voidbarking(){
   }

   voidhungry(){
   }

   voidsleeping(){
   }
}
```

Aclasscancontainanyofthe followingvariable types.

Local variables: Variables defined inside methods, constructors or blocks are called localvariables.Thevariablewillbedeclaredand initializedwithinthe methodandthe variable will be destroyed when the method has completed.

Instance variables: Instance variables are variables within a class but outside any method.Thesevariablesare instantiatedwhentheclassisloaded.Instance variablescan be accessed from inside any method, constructor or blocks of that particular class.

Classvariables: Classvariablesarevariablesdeclaredwithinaclass,outsideany method, with the static keyword.

Aclasscan haveanynumber ofmethodsto accessthevalueofvariouskindsofmethods. Inthe above example, barking(), hungry() and sleeping() are methods.

Belowmentionedaresomeofthe importanttopicsthat needto bediscussedwhenlooking into classes of the Java Language.

Constructors:

Whendiscussingabout classes,oneofthe most important subtopicwould beconstructors.Every class has a constructor. If we do not explicitlywrite a constructor for a class the Java compiler builds a default constructor for that class.

Each time a new object is created, at least one constructor will be invoked. The main rule of constructorsisthattheyshouldhavethesame nameastheclass. Aclasscanhave morethanone constructor.

Exampleofaconstructorisgivenbelow: public

```
class Puppy{
   publicPuppy(){
   }

   publicPuppy(Stringname){
      // Thisconstructorhasoneparameter,name.
   }
}
```

Javaalso supportsSingletonClasseswhere youwouldbeableto createonlyone instanceofa class.

Creatingan Object:

Asmentionedpreviously,aclassprovidestheblueprintsforobjects. So basicallyanobject is created from a class. In Java, the new keyword is used to create new objects.

Therearethreestepswhencreating anobjectfrom a class:

Declaration: Avariabledeclarationwithavariablenamewithanobjecttype.
Instantiation: The 'new'keywordisusedtocreatethe object.
Initialization: The'new'keywordis followedbya callto aconstructor.Thiscall initializes the new object.

Exampleofcreatinganobject isgivenbelow:

```
public class Puppy{

   publicPuppy(Stringname){
      // This constructor has one parameter, name.
      System.out.println("PassedNameis:"+name);
   }
   publicstaticvoid main(String[]args){
      //Followingstatementwouldcreateanobject myPuppy Puppy
      myPuppy = new Puppy( "tommy" );
   }
}
```

Ifwecompileandruntheaboveprogram, thenit wouldproducethe followingresult: Passed

Name is :tommy

AccessingInstanceVariablesand Methods:

Instance variablesand methodsareaccessedvia createdobjects.To accessaninstancevariable the fully qualified path should be as follows:

```
/*Firstcreateanobject*/
```

```
ObjectReference=new Constructor();

/*Nowcallavariableasfollows*/ ObjectReference.variableName;

/*Nowyoucancallaclass methodasfollows*/
ObjectReference.MethodName();
```

Example:

Thisexampleexplainshowtoaccess instancevariablesand methodsofaclass: public

```
class Puppy{

   intpuppyAge;

   publicPuppy(Stringname){
      // This constructor has one parameter, name.
      System.out.println("PassedNameis:"+name);
   }
   publicvoidsetAge(intage){
       puppyAge = age;
   }

   publicint getAge(){
       System.out.println("Puppy'sageis:"+puppyAge);
       return puppyAge;
   }
   publicstaticvoid main(String[]args){
      /*Objectcreation*/
      PuppymyPuppy=newPuppy("tommy");

      /*Callclassmethodto set puppy'sage*/
      myPuppy.setAge( 2 );

      /*Callanotherclassmethodtogetpuppy'sage*/
      myPuppy.getAge( );

      /* You can access instance variable as follows as well */
      System.out.println("VariableValue:"+myPuppy.puppyAge);
   }
}
```

Ifwecompileandruntheaboveprogram, thenit wouldproducethe followingresult: Passed

Name is :tommy

```
Puppy's age is :2
VariableValue:2
```

Sourcefiledeclarationrules:

Asthelast partofthissectionlet'snow lookinto thesourcefiledeclarationrules. Theserulesare essential when declaring classes, *import* statements and *package* statements in a source file.

- There canbe onlyone public class per source file.
- Asourcefilecanhavemultiplenonpublicclasses.
- The public class name should be the name of the source file as well which should be appendedby**.java**attheend. For example:Theclass name is. *publicclassEmployee{}*Then the source file should be as Employee.java.
- Iftheclassisdefined insideapackage,thenthepackagestatement should bethefirst statement in the source file.
- Ifimportstatementsarepresentthentheymust bewrittenbetweenthepackagestatement and the class declaration. If there are no package statements then the import statement should be the first line in the source file.
- Import andpackagestatementswillimplytoalltheclassespresent inthesourcefile. Itis not possible to declare different import and/or package statements to different classes in thesourcefile.

Classeshaveseveralaccess levelsandtherearedifferenttypesofclasses;abstractclasses, final classes, etc. I will be explaining about all these in the access modifiers chapter.

Apart fromtheabove mentionedtypesofclasses,Javaalso hassomespecialclassescalledInner classes and Anonymous classes.

JavaPackage:

Insimple, it isawayofcategorizingtheclassesandinterfaces. Whendevelopingapplications in Java, hundreds ofclasses and interfaces will be written, therefore categorizing these classes is a must as well as makes life much easier.

Importstatements:

InJavaifafullyqualifiedname,whichincludesthepackageandtheclass name,isgiventhen thecompiler caneasilylocatethe sourcecodeorclasses. Import statement isawayofgiving the proper location for the compiler to find that particular class.

For example, thefollowing linewouldaskcompilertoloadalltheclassesavailable indirectory java_installation/java/io :

```
importjava.io.*;
```

ASimpleCaseStudy:

Forourcasestudy,wewillbecreatingtwoclasses.TheyareEmployeeandEmployeeTest.

Firstopennotepadandaddthefollowingcode.Rememberthis istheEmployeeclassandthe class is a public class. Now, save this source file with the name Employee.java.

TheEmployeeclasshas fourinstancevariablesname,age,designationandsalary.Theclasshas one explicitly defined constructor, which takes a parameter.

```
importjava.io.*;
publicclassEmployee{
  String name;
  intage;
  Stringdesignation;
  double salary;

  //This istheconstructoroftheclassEmployee public
  Employee(String name){
    this.name=name;
  }
  //AssigntheageoftheEmployeetothevariableage. public
  void empAge(int empAge){
    age =empAge;
  }
  /*Assignthedesignationtothevariabledesignation.*/
  public void empDesignation(String empDesig){
    designation= empDesig;
  }
  /*Assignthesalarytothevariablesalary.*/
  public void empSalary(double empSalary){
    salary=empSalary;
  }
  /*PrinttheEmployeedetails*/
  public void printEmployee(){
    System.out.println("Name:"+ name );
    System.out.println("Age:" + age );
    System.out.println("Designation:"+designation);
    System.out.println("Salary:" + salary);
  }
}
```

Asmentionedpreviouslyinthistutorial,processingstartsfromthe main method.Thereforein- order for us to run this Employee class there should be main method and objects should be created. We will be creating a separate class for these tasks.

Givenbelow isthe*EmployeeTest*class, whichcreatestwo instancesoftheclassEmployeeand invokes the methods for each object to assign values for each variable.

SavethefollowingcodeinEmployeeTest.javafile import

```
java.io.*;
publicclassEmployeeTest{

  publicstaticvoid main(Stringargs[]){
    /*Createtwoobjectsusing constructor*/
    EmployeeempOne=newEmployee("JamesSmith");
    Employee empTwo = new Employee("MaryAnne");

    // Invoking methods for each object created
    empOne.empAge(26);
    empOne.empDesignation("SeniorSoftwareEngineer");
    empOne.empSalary(1000);
    empOne.printEmployee();

    empTwo.empAge(21);
    empTwo.empDesignation("SoftwareEngineer");
    empTwo.empSalary(500);
    empTwo.printEmployee();
  }
}
```

Now, compile boththeclassesandthenrun*EmployeeTest*to seetheresult asfollows: C :>

```
javac Employee.java
C :>viEmployeeTest.java
C:>javacEmployeeTest.java C
:> java EmployeeTest
Name:James Smith
Age:26
Designation:SeniorSoftwareEngineer
Salary:1000.0
Name:MaryAnne
Age:21
Designation:SoftwareEngineer
Salary:500.0
```

ControlStatements

Theremaybeasituationwhenweneed to executeablock ofcodeseveralnumber oftimes, and is often referred to as a loop.

Javahasveryflexiblethreeloopingmechanisms. Youcanuseoneofthe followingthreeloops: while

Loop
do...whileLoop
for Loop

AsofJava5,the*enhanced forloop*wasintroduced.ThisismainlyusedforArrays.

ThewhileLoop:

Awhileloopisacontrolstructurethatallowsyoutorepeatataskacertainnumberoftimes.

Syntax:

Thesyntaxofawhileloopis:

```
while(Boolean_expression)
{
  //Statements
}
```

Whenexecuting, ifthe*boolean_expression*result istrue,thentheactions insidethe loopwillbe executed. This will continue as long as the expression result is true.

Here,keypointofthe*while*loopisthattheloopmightnoteverrun. Whentheexpressionis testedandtheresult isfalse,theloopbodywillbeskippedandthefirst statement afterthewhile loop will be executed.

Example:

```
publicclassTest{

  publicstaticvoidmain(Stringargs[]){ int
    x = 10;

    while( x < 20 ) {
      System.out.print("valueofx:"+x); x++;
      System.out.print("\n");
    }
  }
}
```

Thiswouldproducethefollowing result:

```
value of x: 10
value of x: 11
value of x: 12
value of x: 13
value of x: 14
value of x: 15
value of x: 16
value of x: 17
value of x: 18
value ofx: 19
```

Thedo...while Loop:

Ado...while loopissimilartoawhile loop,except that ado...while loopisguaranteedtoexecute at least one time.

Syntax:

Thesyntaxofado...whileloopis: do

```
{
  //Statements
}while(Boolean_expression);
```

NoticethattheBooleanexpressionappearsattheendofthe loop, sothe statementsinthe loop execute once before the Boolean is tested.

Ifthe Booleanexpression istrue,theflowofcontroljumpsback up to do,and the statements in the loop execute again. This process repeats until the Boolean expression is false.

Example:

```
publicclassTest{

  publicstaticvoidmain(Stringargs[]){ int
    x = 10;

    do{
      System.out.print("valueofx:"+x); x++;
      System.out.print("\n");
      }while(x<20);
   }
}
```

Thiswouldproducethefollowing result:

```
value of x: 10
value of x: 11
value of x: 12
value of x: 13
value of x: 14
value of x: 15
value of x: 16
value of x: 17
value of x: 18
value ofx: 19
```

The forLoop:

Afor loopisarepetitioncontrolstructurethat allows you to efficientlywritealoopthat needsto execute a specific number of times.

Aforloopisusefulwhenyou knowhowmanytimesatask isto berepeated.

Syntax:

The syntaxofa forloopis:

```
for(initialization;Boolean_expression;update)
{
   //Statements
}
```

Hereistheflowofcontrolinaforloop:

The initialization step is executed first, and onlyonce. This step allows you to declare and initializeanyloopcontrolvariables. Youarenotrequiredtoput astatement here, as long as a semicolon appears.

Next,theBooleanexpressionisevaluated. Ifit istrue,thebodyofthe loopisexecuted. If it is false, the bodyofthe loop does not execute and flow ofcontrol jumps to the next statement past the for loop.

After the bodyof the for loop executes, the flow of control jumps back up to the update statement.Thisstatement allowsyouto updateanyloopcontrolvariables.Thisstatement can be left blank, as long as a semicolon appears after the Boolean expression.

- The Boolean expression is now evaluated again. If it is true, the loop executes and the processrepeatsitself(bodyofloop,thenupdate step,thenBooleanexpression). Afterthe Boolean expression is false, the for loop terminates.

Example:

```
publicclassTest{

   publicstaticvoidmain(Stringargs[]){

      for(int x = 10; x < 20; x = x+1) {
         System.out.print("valueofx:"+x);
         System.out.print("\n");
      }
   }
}
```

Thiswouldproducethefollowing result:

```
value of x: 10
value of x: 11
value of x: 12
value of x: 13
value of x: 14
value of x: 15
value of x: 16
value of x: 17
value of x: 18
value ofx: 19
```

Enhancedforloopin Java:

AsofJava5,theenhancedforloopwasintroduced.ThisismainlyusedforArrays.

Syntax:

Thesyntaxofenhancedforloopis:

```
for(declaration : expression)
{
   //Statements
}
```

- **Declaration:** Thenewlydeclaredblock variable, whichisofatypecompatiblewiththe elements of the array you are accessing. The variable will be available within the for block and its value would be the same as the current arrayelement.

- **Expression:**Thisevaluatestothearrayyouneedto loopthrough.Theexpressioncanbe an array variable or method call that returns an array.

Example:

```
publicclassTest{

  public static void main(String args[]){
    int []numbers={10,20,30,40,50};

    for(int x : numbers ){
      System.out.print(x);
      System.out.print(",");
    }
    System.out.print("\n");
    String[]names={"James","Larry","Tom","Lacy"};
    for( String name : names ) {
      System.out.print(name);
      System.out.print(",");
    }
  }
}
```

Thiswould producethefollowing result:

```
10,20,30,40,50,
James,Larry,Tom,Lacy,
```

ThebreakKeyword:

The*break*keywordisusedto stoptheentire loop.Thebreakkeyword must beused insideany loop or a switch statement.

Thebreakkeywordwillstoptheexecutionofthe innermost loopandstart executingthenext line of code after the block.

Syntax:

Thesyntaxofabreakis asinglestatement insideanyloop: break;

Example:

```
publicclassTest{
  public static void main(Stringargs[]) {
    int []numbers={10,20,30,40,50};

    for(intx:numbers){ if( x
      == 30 ) {
              break;
      }
      System.out.print( x );
      System.out.print("\n");
    }
  }
}
```

Thiswouldproducethefollowing result:

```
10
20
```

ThecontinueKeyword:

The*continue*keyword canbeused inanyofthe loopcontrolstructures. It causesthe loopto immediately jump to the next iteration of the loop.

- Inaforloop,thecontinuekeyword causes flow ofcontrolto immediatelyjump to the update statement.
- Inawhile loopordo/while loop, flowofcontrolimmediatelyjumpstotheBoolean expression.

Syntax:

Thesyntaxofacontinue isasinglestatement insideanyloop:

```
continue;
```

Example:

```
publicclassTest{

  public static void main(Stringargs[]) {
    int []numbers={10,20,30,40,50};

    for(intx:numbers){ if( x
      == 30 ) {
              continue;
      }
      System.out.print(x );
```

```
        System.out.print("\n");
      }
   }
}
```

Thiswouldproducethefollowing result:

```
10
20
40
50
```

Inheritance

Inheritancecanbedefinedastheprocesswhereoneobject acquiresthepropertiesofanother. With the use of inheritance the information is made manageable in a hierarchical order.

When we talk about inheritance, the most commonlyused keyword would be **extends** and **implements**. Thesewordswould determinewhetheroneobject IS-Atypeofanother. Byusing these keywords we can make one object acquire the properties of another object.

IS-A Relationship:

IS-Ais awayofsaying:Thisobject isatypeofthatobject. Letusseehowthe **extends**keyword is used to achieve inheritance.

```
publicclassAnimal{
}

publicclassMammalextendsAnimal{
}

publicclassReptileextendsAnimal{
}

publicclassDogextendsMammal{
}
```

Now,basedontheaboveexample,InObject Orientedterms,thefollowingaretrue: Animal is

- the superclass of Mammal class.
- AnimalisthesuperclassofReptile class.
- Mammal and Reptile are subclasses of Animal class.
- DogisthesubclassofbothMammaland Animalclasses.

Now, ifweconsider theIS-Arelationship,wecansay:

- MammalIS-AAnimal
- Reptile IS-A Animal
- Dog IS-A Mammal
- Hence :DogIS-AAnimalas well

Withuse ofthe extendskeyword thesubclasseswillbeableto inherit alltheproperties ofthe superclass except for the private properties of the superclass.

Wecanassure thatMammalisactuallyanAnimalwiththe useoftheinstance operator.

Example:

```
public class Dog extends Mammal{

   publicstaticvoidmain(Stringargs[]){

      Animal a = new Animal();
      Mammalm=newMammal();
      Dog d = new Dog();

      System.out.println(m instanceof Animal);
      System.out.println(dinstanceofMammal);
      System.out.println(d instanceof Animal);
   }
}
```

Thiswouldproducethefollowing result:

```
true
true
true
```

Sincewehaveagood understanding ofthe**extends**keyword let uslook into how the **implements**keywordisusedtogettheIS-Arelationship.

The**implements**keywordisused byclassesbyinherit frominterfaces. Interfacescannever be extended by the classes.

Example:

```
publicinterface Animal{}

publicclassMammalimplements Animal{
}
```

```
publicclassDogextendsMammal{
}
```

TheinstanceofKeyword:

Let ususethe**instanceof**operatorto checkdeterminewhetherMammalisactuallyanAnimal, and dog is actually an Animal

```
interfaceAnimal{}

classMammalimplementsAnimal{}

public class Dog extends Mammal{
  publicstaticvoid main(String args[]){

    Mammalm=newMammal();
    Dog d = new Dog();

    System.out.println(m instanceof Animal);
    System.out.println(dinstanceofMammal);
    System.out.println(d instanceof Animal);
  }
}
```

Thiswouldproducethefollowing result:

```
true
true
true
```

HAS-A relationship:

Theserelationshipsare mainlybasedontheusage. Thisdetermineswhether acertainclass **HAS- A** certain thing. This relationship helps to reduce duplication of code as well as bugs.

Letsuslookintoanexample: public

```
class Vehicle{}
publicclassSpeed{}
publicclassVanextendsVehicle{
          private Speed sp;
}
```

Thisshowsthat classVanHAS-ASpeed. Byhavingaseparateclass for Speed, wedo nothave to put the entire code that belongs to speed inside the Van class., which makes it possible to reuse the Speed class in multiple applications.

InObject-Oriented feature, the users do not need to bother about whichobject is doing the real work. To achieve this, the Vanclass hidesthe implementationdetails fromthe usersofthe Van class. So basicallywhat happens is the users would asktheVanclass to do a certainactionand the Van class will either do the work byitself or ask another class to performthe action.

Averyimportantfact toremember isthat Javaonlysupportsonlysingle inheritance. This means that a class cannot extend more than one class. Therefore following is illegal:

publicclassextendsAnimal,Mammal{}

However,aclasscanimplementoneormoreinterfaces.Thishasmade Javagetridofthe impossibility of multiple inheritance.

Packages

Packages are used in Java in order to prevent naming conflicts, to controlaccess, to make searching/locatingandusageofclasses, interfaces, enumerationsandannotationseasier,etc.

APackagecanbedefinedasagroupingofrelatedtypes(classes, interfaces,enumerationsand annotations) providing access protection and name space management.

SomeoftheexistingpackagesinJavaare::

- **java.lang**-bundlesthefundamentalclasses
- **java.io**-classesforinput,outputfunctionsarebundledinthispackage

Programmerscandefinetheirownpackagesto bundlegroupofclasses/interfaces,etc. It isa good practice to group related classes implemented by you so that a programmer can easily determine that the classes, interfaces, enumerations, annotations are related.

Sincethepackagecreatesanewnamespacetherewon't beanynameconflictswithnames in other packages. Using packages, it is easier to provide access controland it is also easier to locate the related classes.

Creatingapackage:

When creating a package, you should choose a name for the package and put a **package** statement withthat nameatthetopofeverysourcefilethat containstheclasses, interfaces, enumerations, and annotation types that you want to include in the package.

The**package**statement should bethefirst line inthesourcefile.Therecanbeonlyonepackage statement in each source file, and it applies to alltypes in the file.

Ifapackagestatement is not usedthentheclass, interfaces,enumerations, andannotationtypes will be put into an unnamed package.

Example:

Let uslookat anexamplethat createsapackagecalled **animals**.Itiscommonpracticeto use lowercased names of packages to avoid anyconflicts with the names of classes, interfaces.

Putaninterfaceinthepackage*animals*:

```
/*Filename:Animal.java*/
package animals;

interface Animal {
  public void eat();
  publicvoidtravel();
}
```

Now,putanimplementationinthesamepackage *animals*: package

animals;

```
/*Filename:MammalInt.java*/
publicclassMammalInt implementsAnimal{

  public void eat(){
    System.out.println("Mammaleats");
  }

  public void travel(){
    System.out.println("Mammaltravels");
  }

  publicintnoOfLegs(){
    return 0;
  }

  public static void main(String args[]){
    MammalInt m = new MammalInt();
    m.eat();
    m.travel();
  }
}
```

Now, you compilethesetwo filesand put themin asub-directorycalled **animals**and tryto run as follows:

```
$mkdiranimals
$cpAnimal.classMammalInt.classanimals
$javaanimals/MammalInt
Mammal eats
Mammaltravels
```

TheimportKeyword:

Ifaclasswantstouseanother class inthesamepackage,thepackagenamedoesnotneedto be used. Classes in the same package find each other without any special syntax.

Example:

Here, aclass named Boss is added to thepayroll packagethat alreadycontains Employee. The Bosscanthenrefer to the Employee class without using the payrollprefix, as demonstrated by the following Boss class.

```
package  payroll;

publicclassBoss
{
  publicvoidpayEmployee(Employee e)
  {
    e.mailCheck();
  }
}
```

What happens ifBoss is notinthepayrollpackage?TheBossclass mustthenuseoneofthe following techniques for referring to a class in a different package.

- Thefullyqualifiednameoftheclasscanbeused. Forexample:

```
payroll.Employee
```

- Thepackagecanbe importedusingtheimportkeywordandthewildcard(*). For example:

```
importpayroll.*;
```

- Theclassitselfcanbe importedusingtheimportkeyword.Forexample:

```
import payroll.Employee;
```

Note: Aclass filecancontainanynumberofimport statements.The import statements must appear after the package statement and before the class declaration.

TheDirectoryStructureofPackages:

Twomajor resultsoccurwhenaclassisplacedina package:

- Thenameofthepackagebecomesapartofthenameoftheclass, aswe just discussed in the previous section.
- Thenameofthepackagemust matchthedirectorystructurewherethecorresponding bytecode resides.

Hereis simplewayofmanagingyourfiles inJava:

Putthesourcecodeforaclass, interface,enumeration,orannotationtype inatext filewhose name is the simple name of the type and whose extension is **.java**. For example:

```
//FileName:Car.java

package vehicle;

publicclassCar{
   //Classimplementation.
}
```

Now, putthesourcefile inadirectorywhosename reflectsthenameofthepackagetowhichthe class belongs:

```
....\vehicle\Car.java
```

Now,thequalifiedclassnameandpathnamewouldbeasbelow: Class

- name -> vehicle.Car
- Pathname->vehicle\Car.java(inwindows)

Ingeneral,acompanyusesitsreversed Internet domainnameforitspackage names.Example:A company's Internet domain name is apple.com, then all its package names would start with com.apple. Each component of the package name corresponds to a subdirectory.

Example:Thecompanyhadacom.apple.computerspackagethatcontainedaDell.javasource file, it would be contained in a series of subdirectories like this:

```
....\com\apple\computers\Dell.java
```

At thetimeofcompilation, thecompiler creates adifferent output file for eachclass, interface andenumerationdefined in it. Thebase nameoftheoutput file is the nameofthetype, and its extension is **.class**

Forexample:

```
//FileName: Dell.java

packagecom.apple.computers;
public class Dell{

}
classUps{

}
```

Now,compilethis fileas followsusing-d option:

```
$javac-d. Dell.java
```

Thiswouldputcompiledfilesas follows:

```
.\com\apple\computers\Dell.class
.\com\apple\computers\Ups.class
```

Youcanimportalltheclassesor interfacesdefinedin\com\apple\computers\asfollows: import

com.apple.computers.*;

Likethe.javasourcefiles,thecompiled.class filesshould be inaseriesofdirectoriesthat reflect the package name. However, the path to the .class files does not have to be the same as the path to the .java source files. You can arrange your source and class directories separately, as:

```
<path-one>\sources\com\apple\computers\Dell.java

<path-two>\classes\com\apple\computers\Dell.class
```

Bydoing this, it is possible to give the classes directoryto other programmerswithout revealing your sources. Youalso needto managesourceand class files inthis manner sothatthecompiler and the Java Virtual Machine (JVM) can find all the types your program uses.

Thefullpathtotheclassesdirectory,<path-two>\classes, iscalledtheclasspath, and isset with the CLASSPATH system variable. Both the compiler and the JVM construct the path to your .classfilesbyadding the packagenametothe class path.

Say<path-two>\classes istheclasspath,andthepackagename iscom.apple.computers,thenthe compiler and JVM will look for .class files in <path-two>\classes\com\apple\compters.

A class path may include several paths. Multiple paths should be separated by a semicolon (Windows) or colon(Unix). Bydefault, the compiler andthe JVM searchthe current directory andtheJAR filecontainingtheJavaplatformclassessothatthesedirectoriesareautomatically in the class path.

SetCLASSPATHSystemVariable:

Todisplaythecurrent CLASSPATHvariable,use thefollowingcommandsinWindowsand UNIX (Bourne shell):

- InWindows->C:\>setCLASSPATH In
- UNIX -> % echo $CLASSPATH

Todeletethecurrent contentsoftheCLASSPATH variable,use: In

- Windows -> C:\> set CLASSPATH=
- InUNIX->%unsetCLASSPATH;exportCLASSPATH To

set the CLASSPATH variable:

- InWindows->set CLASSPATH=C:\users\jack\java\classes
- InUNIX->%CLASSPATH=/home/jack/java/classes;exportCLASSPATH

Abstraction

Abstraction refers to the abilityto make a class abstract in OOP. An abstract class is one that cannotbe instantiated. Allother functionalityoftheclassstillexists, and its fields, methods, and constructors are allaccessed in the same manner. You just cannot create an instance of the abstract class.

Ifa class is abstract and cannot be instantiated, the class does not have much use unless it is subclass. This is typically how abstract classes come about during the design phase. A parent classcontainsthecommonfunctionalityofacollectionofchildclasses, buttheparentclass itself is too abstract to be used on its own.

AbstractClass:

Usethe**abstract** keywordtodeclareaclassabstract.Thekeywordappearsintheclass declaration somewhere before the class keyword.

```
/*Filename:Employee.java*/
public abstract class Employee
{
  private String name;
  privateStringaddress;
  private int number;
  publicEmployee(Stringname,Stringaddress,intnumber)
  {
    System.out.println("ConstructinganEmployee");
```

```
     this.name=name;
     this.address=address;
     this.number=number;
   }
   publicdoublecomputePay()
   {
    System.out.println("InsideEmployeecomputePay");
    return 0.0;
   }
   publicvoidmailCheck()
   {
     System.out.println("Mailingacheckto"+this.name
      +"" +this.address);
   }
   publicString toString()
   {
     returnname+""+address+ ""+number;
   }
   publicString getName()
   {
     returnname;
   }
   publicString getAddress()
   {
     returnaddress;
   }
   publicvoidsetAddress(StringnewAddress)
   {
     address= newAddress;
   }
   publicint getNumber()
   {
    return number;
   }
}
```

Noticethat nothing isdifferent inthisEmployeeclass.Theclassisnowabstract,but it stillhas three fields, seven methods, and one constructor.

Nowifyouwouldtryasfollows:

```
/*Filename:AbstractDemo.java*/ public
class AbstractDemo
{
   publicstaticvoidmain(String[] args)
   {
      /*Followingisnotallowedandwouldraiseerror*/
      Employeee=newEmployee("GeorgeW.", "Houston,TX",43);

      System.out.println("\nCallmailCheckusingEmployeereference--"); e.mailCheck();
    }
}
```

Whenyouwouldcompileaboveclassthen youwouldgetthefollowingerror:

```
Employee.java:46: Employee is abstract; cannot be instantiated
      Employeee=newEmployee("GeorgeW.", "Houston,TX", 43);
                   ^
1error
```

ExtendingAbstractClass:

WecanextendEmployeeclassin normalwayasfollows:

```
/*Filename:Salary.java*/
public classSalaryextendsEmployee
{
   privatedoublesalary;//Annualsalary
   publicSalary(Stringname,Stringaddress,intnumber,double
      salary)
   {
       super(name,address,number);
       setSalary(salary);
   }
   publicvoidmailCheck()
   {
       System.out.println("Within mailCheckofSalaryclass");
       System.out.println("Mailing check to " + getName()
       +" withsalary" +salary);
   }
   publicdoublegetSalary()
   {
       returnsalary;
   }
   publicvoidsetSalary(double newSalary)
   {
       if(newSalary>=0.0)
       {
          salary=newSalary;
       }
   }
```

```
  publicdoublecomputePay()
  {
    System.out.println("Computingsalarypayfor"+getName()); return
    salary/52;
  }
}
```

Here, wecannot instantiateanewEmployee, but if weinstantiateanewSalaryobject,theSalary object will inherit the three fields and seven methods from Employee.

```
/*Filename:AbstractDemo.java*/ public
class AbstractDemo
{
  publicstaticvoidmain(String[] args)
  {
    Salarys=newSalary("MohdMohtashim","Ambehta,UP",3,3600.00);
    Employee e = new Salary("John Adams", "Boston, MA", 2, 2400.00);

    System.out.println("CallmailCheckusingSalaryreference--"); s.mailCheck();

    System.out.println("\nCallmailCheckusingEmployeereference--"); e.mailCheck();
   }
}
```

Thiswouldproducethefollowing result:

```
ConstructinganEmployee
ConstructinganEmployee
CallmailCheckusingSalaryreference-- Within
mailCheck of Salary class
MailingchecktoMohd Mohtashimwithsalary3600.0

CallmailCheckusingEmployeereference--
Within mailCheck of Salary class
Mailing check toJohnAdamswithsalary2400.
```

AbstractMethods:

If youwant aclasstocontainaparticular methodbut youwanttheactualimplementationofthat method to be determined bychild classes, you can declare the method in the parent class as abstract.

Theabstract keywordis also usedto declareamethodasabstract.Anabstractmethodconsistsof a method signature, but no method body.

Abstract methodwouldhaveno definition,and itssignatureisfollowed byasemicolon,not curly braces as follows:

```
publicabstract classEmployee
{
  private String name;
  privateStringaddress;
  private int number;

  publicabstract doublecomputePay();

  //Remainderofclassdefinition
}
```

Declaringamethodasabstracthastwo results:

- Theclass must also bedeclaredabstract. Ifaclasscontainsanabstract method,theclass must be abstract as well.
- Anychildclassmusteitheroverridetheabstractmethodordeclareitselfabstract.

Achildclassthat inherits anabstractmethod must override it. Iftheydo not,theymust be abstract and any of their children must override it.

Eventually, adescendant classhastoimplementtheabstractmethod;otherwise, youwouldhave a hierarchyof abstract classes that cannot be instantiated.

IfSalaryisextendingEmployeeclass,thenit isrequiredtoimplement computePay() methodas follows:

```
/*Filename:Salary.java*/
public classSalaryextendsEmployee
{
  privatedoublesalary;//Annualsalary

  publicdouble computePay()
  {
    System.out.println("Computingsalarypayfor"+getName()); return
    salary/52;
  }

  //Remainderofclassdefinition
}
```

Interface

Aninterface isacollectionofabstract methods. Aclassimplementsaninterface, thereby inheriting the abstract methods of the interface.

Aninterface isnot aclass.Writinganinterface is similartowritingaclass, buttheyaretwo different concepts. A class describes the attributes and behaviors ofan object. An interface contains behaviors that a class implements.

Unlesstheclassthat implementstheinterface isabstract, allthe methodsofthe interface needto be defined in the class.

An interface is similar to a class in the following ways:

- Aninterfacecancontainanynumberofmethods.
- Aninterface iswrittenin afilewitha**.java**extension, withthenameofthe interface matching the name of the file.
- Thebytecodeofaninterfaceappearsina**.class**file.
- Interfacesappear inpackages,andtheircorrespondingbytecodefilemust beina directory structure that matches the package name.

However,aninterface isdifferent fromaclassinseveralways, including: You

- cannot instantiate an interface.
- Aninterfacedoesnot containanyconstructors.
- Allofthe methods in an interface are abstract.
- Aninterfacecannotcontaininstancefields. Theonlyfieldsthat canappear inaninterface must be declared both static and final.
- Aninterface isnot extendedbyaclass;it isimplementedbyaclass. An
- interface can extend multiple interfaces.

DeclaringInterfaces:

The**interface**keywordisusedtodeclareaninterface. Hereisasimpleexampletodeclarean interface:

Example:

Letuslookatanexamplethatdepictsencapsulation:

```
/*Filename:NameOfInterface.java*/
import java.lang.*;
//Anynumberofimport statements

publicinterfaceNameOfInterface
{
  //Anynumberoffinal,staticfields
  //Anynumberofabstract method declarations\
```

}

Interfaceshavethefollowingproperties:

- Aninterface is implicitlyabstract.Youdonotneedtousethe**abstract**keywordwhen declaring an interface.
- Eachmethodinaninterface isalso implicitlyabstract,sotheabstract keywordisnot needed.
- Methodsinaninterfaceareimplicitlypublic.

Example:

```
/*Filename:Animal.java*/
interface Animal {

  public void eat();
  publicvoidtravel();
}
```

ImplementingInterfaces:

Whena class implements an interface, you canthink ofthe class as signing a contract, agreeing toperformthespecific behaviorsofthe interface. Ifaclassdoesnot performallthe behaviorsof the interface, the class must declare itself as abstract.

Aclassusesthe**implements**keywordto implement aninterface. The implementskeyword appears in the class declaration following the extends portion of the declaration.

```
/*Filename:MammalInt.java*/
publicclassMammalInt implementsAnimal{

  public void eat(){
    System.out.println("Mammaleats");
  }

  public void travel(){
    System.out.println("Mammaltravels");
  }

  publicintnoOfLegs(){
    return 0;
  }

  public static void main(Stringargs[]){
    MammalInt m = new MammalInt();
    m.eat();
    m.travel();
  }
}
```

Thiswouldproducethefollowing result:

```
Mammal eats
Mammaltravels
```

Whenoverridingmethodsdefinedininterfacesthereareseveralrulestobe followed:

- Checkedexceptionsshouldnot bedeclaredonimplementationmethodsotherthanthe ones declared bythe interface method or subclasses ofthose declared bythe interface method.
- Thesignatureofthe interface methodandthesamereturntypeorsubtype should be maintained when overriding the methods.
- Animplementationclassitselfcanbeabstractandifsointerfacemethodsneednotbe implemented.

Whenimplementationinterfacesthereareseveralrules:

- Aclasscanimplementmorethanone interface atatime.
- Aclasscanextendonlyoneclass,butimplementmanyinterfaces.
- Aninterfacecanextendanother interface, similarlytothewaythat aclasscanextend another class.

ExtendingInterfaces:

An interface canextend another interface, similarlyto the waythat a classcanextend another class. The**extends** keyword is used to extend an interface, and thechild interface inherits the methods of the parent interface.

Thefollowing SportsinterfaceisextendedbyHockeyandFootballinterfaces.

```
//Filename:Sports.java
public interface Sports
{
  public void setHomeTeam(String name);
  publicvoidsetVisitingTeam(Stringname);
}

//Filename:Football.java
publicinterfaceFootballextendsSports
{
  public void homeTeamScored(int points);
  publicvoidvisitingTeamScored(intpoints);
```

```
   publicvoidendOfQuarter(intquarter);
}

//Filename:Hockey.java
publicinterfaceHockeyextendsSports
{
   public void homeGoalScored();
   public void visitingGoalScored();
   publicvoidendOfPeriod(intperiod);
   public void overtimePeriod(int ot);
}
```

The Hockeyinterface has four methods, but it inherits two fromSports; thus, a class that implementsHockeyneedsto implement allsix methods. Similarly, aclassthat implements Footballneedstodefinethethreemethods fromFootballandthetwo methodsfromSports.

ExtendingMultipleInterfaces:

AJavaclasscanonlyextendoneparent class.Multiple inheritance isnot allowed.Interfacesare not classes, however, and an interface can extend more than one parent interface.

Theextendskeywordis usedonce, andtheparent interfacesaredeclared inacomma-separated list.

Forexample, iftheHockeyinterfaceextended bothSportsandEvent, it would bedeclaredas: public

interface Hockey extends Sports, Event

TaggingInterfaces:

The most commonuseofextending interfacesoccurswhentheparent interfacedoesnotcontain anymethods. For example, the MouseListener interface inthe java.awt.event package extended java.util.EventListener, which is defined as:

```
packagejava.util;
publicinterfaceEventListener
{}
```

Aninterfacewithno methodsinit isreferredto asa**tagging**interface.Therearetwo basic design purposes of tagging interfaces:

Createsacommonparent: AswiththeEventListener interface, whichisextended bydozensof other interfaces in the Java API, you can use a tagging interface to create a common parent among a group of interfaces. For example, when an interface extends EventListener, the JVM knows that this particular interface is going to be used in an event delegation scenario.

Adds a data type to a class: This situation is where the termtagging comes from. A class that implementsatagging interfacedoesnot needtodefineanymethods(sincethe interfacedoesnot have any), but the class becomes an interface type through polymorphism.

Exception

Anexception isa problemthat arisesduring the executionofaprogram. Anexceptioncanoccur for many different reasons, including the following:

- Auser hasenteredinvalid data.
- Afilethatneedstobeopenedcannotbefound.
- Anetworkconnectionhas beenlost inthe middle ofcommunicationsortheJVMhasrun out of memory.

Someoftheseexceptionsarecaused byuser error,othersbyprogrammer error,and othersby physical resources that have failed in some manner.

Tounderstandhowexceptionhandlingworksin Java, youneedto understandthethree categories of exceptions:

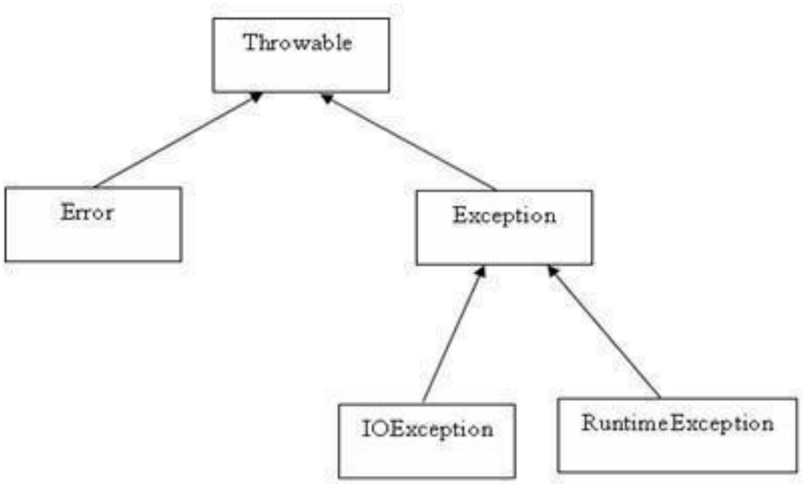
- **Checkedexceptions:** Acheckedexceptionisanexceptionthat istypicallyauser erroror a problemthat cannot be foreseen bythe programmer. For example, if a file is to be opened, but the file cannot be found, an exception occurs. These exceptions cannotsimply be ignored at the time of compilation.
- **Runtime exceptions:** A runtime exception is an exception that occurs that probably couldhavebeenavoided bytheprogrammer.Asopposedtocheckedexceptions,runtime exceptions are ignored at the time of compilation.
- **Errors:**Thesearenotexceptionsat all, but problemsthat arisebeyondthecontrolofthe user ortheprogrammer. Errorsaretypicallyignoredin your codebecause youcanrarely do anything about an error. For example, if a stack overflow occurs, an error will arise. Theyarealsoignoredat thetimeofcompilation.

ExceptionHierarchy:

All exception classes are subtypes of the java.lang.Exception class. The exception class is a subclassoftheThrowableclass. Otherthantheexceptionclassthere isanother subclasscalled Error which is derived from the Throwable class.

Errorsarenot normallytrappedformtheJavaprograms. Theseconditions normallyhappenin case of severe failures, which are not handled bythe java programs. Errors are generatedto indicate errors generated bythe runtime environment. Example : JVM is out of Memory. Normallyprogramscannotrecover fromerrors.

TheExceptionclasshastwomainsubclasses:IOExceptionclassandRuntimeExceptionClass.



HereisalistofmostcommoncheckedanduncheckedJava'sBuilt-inExceptions.

ExceptionsMethods:

Followingisthe listofimportantmedthodsavailable intheThrowableclass.

SN	MethodswithDescription
1	**publicString getMessage()** Returnsadetailed messageabouttheexceptionthathasoccurred.Thismessage isinitialized in the Throwable constructor.
2	**publicThrowablegetCause()** Returnsthe causeoftheexceptionasrepresentedbyaThrowable object.
3	**publicString toString()** ReturnsthenameoftheclassconcatenatedwiththeresultofgetMessage()
4	**publicvoidprintStackTrace()** PrintstheresultoftoString()along withthestacktraceto System.err,the erroroutputstream.
5	**publicStackTraceElement[] getStackTrace()** Returns an arraycontaining each element on the stack trace. The element at index 0 representsthetopofthecallstack, andthe last element inthearrayrepresentsthe method at the bottom of the call stack.
6	**publicThrowablefillInStackTrace()** FillsthestacktraceofthisThrowableobject withthecurrent stacktrace, adding to any previous information in the stack trace.

CatchingExceptions:

A method catches anexceptionusing a combinationofthe **try** and **catch** keywords. Atry/catch block is placedaround thecodethat might generateanexception. Codewithinatry/catchblock is referred to as protected code, and the syntax for using try/catch looks like the following:

```
try
{
  //Protectedcode
}catch(ExceptionNamee1)
{
  //Catchblock
}
```

A catch statement involves declaring the type of exception you are trying to catch. If an exceptionoccursinprotectedcode,thecatchblock (orblocks)that followsthetryischecked. If the type ofexceptionthat occurred is listed ina catchblock,the exception ispassed to the catch block much as an argument is passed into a method parameter.

Example:

Thefollowing isanarrayisdeclared with2 elements. Thenthe codetriesto accessthe3rd element of the array which throws an exception.

```
//FileName:ExcepTest.java
import java.io.*;
publicclassExcepTest{

  publicstaticvoidmain(Stringargs[]){
    try{
      inta[] =newint[2];
      System.out.println("Accesselementthree:"+a[3]);
    }catch(ArrayIndexOutOfBoundsException e){
      System.out.println("Exceptionthrown:"+e);
    }
    System.out.println("Outofthe block");
  }
}
```

Thiswouldproducethefollowing result:

```
Exceptionthrown:java.lang.ArrayIndexOutOfBoundsException:3 Out
of the block
```

MultiplecatchBlocks:

Atryblockcanbe followed bymultiplecatchblocks. Thesyntaxfor multiplecatchblocks looks like the following:

```
try
{
  //Protectedcode
}catch(ExceptionType1e1)
{
  //Catchblock
}catch(ExceptionType2e2)
{
  //Catchblock
}catch(ExceptionType3e3)
{
  //Catchblock
}
```

The previous statements demonstratethree catch blocks, but you can have anynumber of them after asingletry. Ifanexceptionoccursintheprotected code,theexception isthrowntothe first catch block in the list. Ifthe datatype ofthe exception thrown matches ExceptionType1, it gets caught there. If not, the exception passes down to the second catch statement. This continues untilthe exceptioneither is caught or falls throughallcatches, inwhichcase the current method stops execution and the exception is thrown down to the previous method on the call stack.

Example:

Hereiscodesegment showinghowto usemultiple try/catchstatements. try

```
{
  file=newFileInputStream(fileName); x
  = (byte) file.read();
}catch(IOExceptioni)
{
  i.printStackTrace();
  return -1;
}catch(FileNotFoundExceptionf)//Not valid!
{
  f.printStackTrace();
  return -1;
}
```

Thethrows/throwKeywords:

Ifamethoddoesnothandleacheckedexception,themethodmustdeclareitusingthe**throws** keyword.Thethrowskeyword appearsattheend ofamethod'ssignature.

Youcanthrowanexception, either anewlyinstantiatedoneoranexceptionthat youjust caught, by using the **throw** keyword. Tryto understand the different in throws and throw keywords.

Thefollowing methoddeclaresthat itthrowsaRemoteException: import

```
java.io.*;
publicclassclassName
{
  publicvoiddeposit(doubleamount)throwsRemoteException
  {
    //Methodimplementation
    thrownewRemoteException();
  }
  //Remainderofclassdefinition
}
```

Amethodcandeclarethat it throwsmorethanone exception, inwhichcasetheexceptionsare declared in a list separated bycommas. For example, the following method declares that it throwsaRemoteExceptionandanInsufficientFundsException:

```
import
java.io.*;publicclasscl
assName
{
  publicvoidwithdraw(doubleamount)throwsRemoteException,
                         InsufficientFundsException
  {
     //Methodimplementation
  }
  //Remainderofclassdefinition
}
```

ThefinallyKeyword

Thefinallykeyword isused to createablock ofcodethat followsa tryblock. A finallyblock of code always executes, whether or not an exception has occurred.

Usinga finallyblockallows youtorunanycleanup-typestatementsthat youwant toexecute, no matter what happens in the protected code.

Afinallyblockappearsat theendofthecatchblocksandhasthe following syntax: try

```
{
  //Protectedcode
}catch(ExceptionType1e1)
{
  //Catchblock
}catch(ExceptionType2e2)
{
  //Catchblock
}catch(ExceptionType3e3)
{
  //Catchblock
}finally
{
  //Thefinallyblockalwaysexecutes.
}
```

Example:

```
publicclassExcepTest{

  publicstaticvoidmain(Stringargs[]){ int
    a[] = new int[2];
    try{
      System.out.println("Accesselementthree:"+a[3]);
    }catch(ArrayIndexOutOfBoundsException e){
      System.out.println("Exceptionthrown:"+e);
    }
    finally{
      a[0]=6;
      System.out.println("First element value: " +a[0]);
      System.out.println("Thefinallystatementisexecuted");
    }
  }
}
```

Thiswouldproducethefollowing result:

```
Exceptionthrown:java.lang.ArrayIndexOutOfBoundsException:3 First
element value: 6
Thefinallystatementisexecuted
```

Note the following:

- Acatchclause cannotexistwithoutatrystatement.
- It isnot compulsoryto havefinallyclauseswhenever atry/catchblock is present. The
- try block cannot be present without either catch clause or finally clause.
- Anycodecannotbepresentinbetweenthetry, catch,finallyblocks.

DeclaringyouownException:

Youcancreateyourownexceptions inJava. Keep thefollowingpoints inmindwhenwriting your own exception classes:

- Allexceptionsmustbea childofThrowable.
- If youwant to writeacheckedexceptionthat isautomaticallyenforcedbytheHandleor Declare Rule, you need to extend the Exception class.
- Ifyouwantto writearuntimeexception,youneedtoextendtheRuntimeExceptionclass.

WecandefineourownExceptionclassas below:

```
classMyExceptionextendsException{
}
```

You just need to extend the Exception class to create your own Exception class. These are considered to be checked exceptions. The following InsufficientFundsException class is a user-definedexceptionthat extendstheExceptionclass, making it acheckedexception.Anexception class is like any other class, containing useful fields and methods.

Example:

```
//FileNameInsufficientFundsException.java
import java.io.*;

publicclassInsufficientFundsExceptionextendsException
{
  privatedoubleamount;
  publicInsufficientFundsException(doubleamount)
  {
    this.amount= amount;
  }
  publicdoublegetAmount()
  {
    returnamount;
  }
}
```

Todemonstrateusingouruser-definedexception,the followingCheckingAccount classcontains a withdraw() method that throws an InsufficientFundsException.

```
//FileNameCheckingAccount.java
import java.io.*;

publicclassCheckingAccount
{
  privatedoublebalance;
  private int number;
  publicCheckingAccount(intnumber)
  {
    this.number=number;
  }
  publicvoiddeposit(doubleamount)
  {
    balance+=amount;
  }
     publicvoidwithdraw(doubleamount)throws
                      InsufficientFundsException
  {
    if(amount<=balance)
    {
      balance-=amount;
    }
    else
    {
      doubleneeds=amount- balance;
      thrownewInsufficientFundsException(needs);
    }
  }
  publicdoublegetBalance()
  {
    return balance;
  }
  publicint getNumber()
  {
    return number;
  }
}
```

ThefollowingBankDemo programdemonstratesinvokingthedeposit()andwithdraw()methods of CheckingAccount.

```
//FileNameBankDemo.java public
class BankDemo
{
  publicstaticvoidmain(String[] args)
  {
    CheckingAccountc=newCheckingAccount(101);
    System.out.println("Depositing $500...");
    c.deposit(500.00);
    try
    {
      System.out.println("\nWithdrawing$100...");
      c.withdraw(100.00);
```

```
        System.out.println("\nWithdrawing$600...");
        c.withdraw(600.00);
      }catch(InsufficientFundsExceptione)
      {
        System.out.println("Sorry,butyouareshort$"
                                  +e.getAmount());
        e.printStackTrace();
      }
    }
}
```

Compilealltheabovethreefilesand runBankDemo, thiswouldproducethefollowingresult:

```
Depositing

$500...

Withdrawing$100...

Withdrawing$600...
Sorry,but youareshort$200.0
InsufficientFundsException
       atCheckingAccount.withdraw(CheckingAccount.java:25)
       at BankDemo.main(BankDemo.java:13)
```

CommonExceptions:

InJava,itispossibletodefinetwocatergoriesofExceptionsand Errors.

- **JVMExceptions:** -Theseareexceptions/errorsthat areexclusivelyor logicallythrown by the JVM. Examples : NullPointerException, ArrayIndexOutOfBoundsException, ClassCastException,
- **Programmaticexceptions:** - Theseexceptionsarethrownexplicitlybytheapplication or the API programmers Examples: IllegalArgumentException, IllegalStateException.

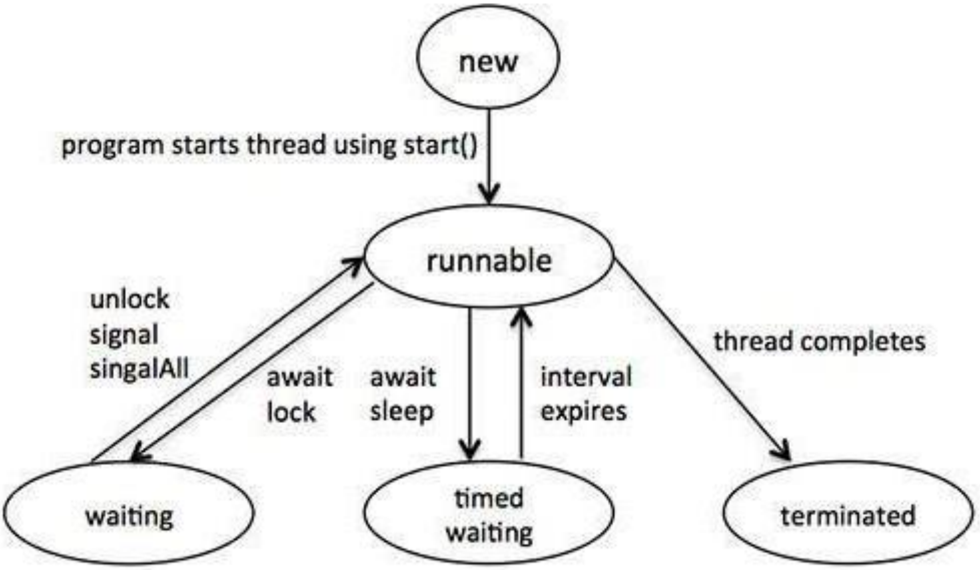
Java is a*multithreaded programming language* which means we can develop multithreaded program using Java. A multithreaded program contains two or more parts that can run concurrentlyandeachpart canhandledifferent taskatthesametime makingoptimaluseofthe available resources specially when your computer has multiple CPUs.

Bydefinitionmultitasking iswhenmultipleprocessessharecommonprocessingresourcessuch as a CPU. Multithreading extends the idea of multitasking into applications where you can subdivide specific operations within a single application into individual threads. Each of the threadscanruninparallel.TheOSdividesprocessingtime notonlyamongdifferent applications, but also among each thread within an application.

Multithreadingenables youtowriteinawaywheremultipleactivitiescanproceedconcurrently in the same program.

LifeCycleofa Thread:

Athreadgoesthroughvariousstages inits lifecycle. Forexample, athread is born, started,runs, and then dies. Following diagram shows complete life cycle of a thread.



Above-mentionedstagesareexplained here:

- **New:**Anewthreadbegins its lifecycle inthenewstate.Itremains inthisstateuntilthe program starts the thread. It is also referred to as a born thread.
- **Runnable:**Afteranewlybornthreadisstarted,thethreadbecomesrunnable. Athreadin this state is considered to be executing its task.
- **Waiting:** Sometimes, a threadtransitions to the waiting state while the thread waits for anotherthreadto performatask.Athreadtransitions back to therunnable stateonlywhen another thread signals the waiting thread to continue executing.
- **Timedwaiting:**Arunnablethreadcanenter thetimedwaiting stateforaspecified intervaloftime. Athread inthisstatetransitions backto therunnablestatewhenthat time interval expires or when the event it is waiting for occurs.
- **Terminated:**Arunnablethreadentersthe terminatedstatewhenit completesitstaskor otherwise terminates.

Thread Priorities:

EveryJavathread hasapriority that helpstheoperating systemdetermine theorder inwhich threads are scheduled.

Javathreadprioritiesare intherangebetweenMIN_PRIORITY(aconstantof1) and MAX_PRIORITY (a constant of 10). Bydefault, everythread is given priority NORM_PRIORITY (a constant of 5).

Threadswithhigher priorityare moreimportanttoaprogramandshouldbe allocatedprocessor time before lower-priority threads. However, thread priorities cannot guarantee the order in which threads execute and very much platform dependentant.

CreateThreadbyImplementingRunnableInterface:

Ifyourclassis intendedtobeexecutedasathreadthenyoucanachievethisbyimplementing **Runnable**interface.Youwillneedtofollowthreebasicsteps:

Step1:

As a first step you need to implement a run() method provided by**Runnable** interface. This methodprovidesentrypoint forthethreadand you willput youcompletebusinesslogic inside this method. Following is simple syntax of run() method:

publicvoidrun()

Step2:

At secondstepyouwillinstantiatea**Thread**objectusingthefollowingconstructor: Thread(Runnable

threadObj, String threadName);

Where,*threadObj*isaninstanceofaclassthatimplementsthe**Runnable**interfaceand **threadName**isthenamegiventothenew thread.

Step3

OnceThreadobject iscreated, youcanstart it bycalling**start()**method, whichexecutesacall to run( ) method. Following is simple syntax of start() method:

void start();

Example:

Hereisanexamplethat createsanewthreadandstartsit running: class

```
RunnableDemo implements Runnable {
  privateThread t;
  privateStringthreadName;

  RunnableDemo(Stringname){threadName= name;
     System.out.println("Creating " +
     threadName);
  }
  publicvoidrun() {
    System.out.println("Running"+threadName);
    try{
      for(inti=4; i>0; i--){
        System.out.println("Thread:"+threadName+","+i);
        //Letthethreadsleepforawhile.
        Thread.sleep(50);
      }
   }catch(InterruptedExceptione){
      System.out.println("Thread"+threadName+"interrupted.");
   }
   System.out.println("Thread"+threadName+" exiting.");
  }

  publicvoidstart()
  {
    System.out.println("Starting"+threadName); if
    (t == null)
    {
      t=newThread(this,threadName);
      t.start ();
    }
  }

}

publicclassTestThread{
  publicstaticvoidmain(Stringargs[]) {

    RunnableDemoR1=newRunnableDemo("Thread-1");
    R1.start();

    RunnableDemoR2=newRunnableDemo("Thread-2");
    R2.start();
  }
}
```

Thiswouldproducethefollowingresult:

```
Creating Thread-1
StartingThread-1
Creating Thread-2
Starting Thread-2
Running Thread-1
Thread:Thread-1,4
Running Thread-2
Thread:Thread-2,4
```

```
Thread:Thread-1,3
Thread:Thread-2,3
Thread:Thread-1,2
Thread:Thread-2,2
Thread:Thread-1,1
Thread: Thread-2, 1
ThreadThread-1exiting.
Thread Thread-2exiting.
```

CreateThreadbyExtendingThread Class:

The second wayto create a thread is to create a new class that extends **Thread** class using the followingtwo simplesteps.Thisapproachprovidesmoreflexibilityinhandling multiplethreads created using available methods in Thread class.

Step1

You will need to override **run( )** method available in Thread class. This method provides entry point for thethreadand youwill put youcompletebusiness logic insidethis method. Following is simple syntax of run() method:

```
publicvoidrun()
```

Step2

OnceThreadobject iscreated, youcanstart it bycalling**start()**method, whichexecutesacall to run( ) method. Following is simple syntax of start() method:

```
void start();
```

Example:

Hereistheprecedingprogramrewrittentoextend Thread:

```
class ThreadDemo extends Thread {
   privateThread t;
   privateStringthreadName;

   ThreadDemo( Stringname){threadName=name;
       System.out.println("Creating"+threadName);
   }
   publicvoidrun() {
      System.out.println("Running"+threadName);
      try{
         for(inti=4; i>0; i--){
```

```
        System.out.println("Thread:"+threadName+","+i);
        //Letthethreadsleepforawhile.
        Thread.sleep(50);
      }
   }catch(InterruptedExceptione){
      System.out.println("Thread"+threadName+"interrupted.");
   }
   System.out.println("Thread"+threadName+" exiting.");
  }

  publicvoidstart()
  {
    System.out.println("Starting"+threadName); if
    (t == null)
    {
      t=newThread(this,threadName);
      t.start ();
    }
  }

}

publicclassTestThread{
  publicstaticvoidmain(Stringargs[]) {

    ThreadDemoT1=newThreadDemo("Thread-1"); T1.start();

    ThreadDemoT2=newThreadDemo("Thread-2"); T2.start();
  }
}
```

Thiswouldproducethefollowingresult:

```
Creating Thread-1
StartingThread-1
Creating Thread-2
Starting Thread-2
Running Thread-1
Thread:Thread-1,4
Running Thread-2
Thread:Thread-2,4
Thread:Thread-1,3
Thread:Thread-2,3
Thread:Thread-1,2
```

```
Thread:Thread-2,2
Thread:Thread-1,1
Thread: Thread-2, 1
ThreadThread-1exiting.
Thread Thread-2exiting.
```

ThreadMethods:

FollowingisthelistofimportantmethodsavailableintheThread class.

SN	Methodswith Description
1	**publicvoidstart()** Startsthethread inaseparate pathofexecution, then invokestherun() methodonthis Thread object.
2	**publicvoidrun()** IfthisThreadobject was instantiatedusingaseparateRunnabletarget,therun()methodis invoked on that Runnable object.
3	**publicfinalvoidsetName(Stringname)** ChangesthenameoftheThreadobject.Thereisalso agetName() methodforretrievingthe name.
4	**publicfinalvoidsetPriority(intpriority)** Setsthe priorityofthisThreadobject.Thepossiblevaluesarebetween1 and10.
5	**publicfinalvoidsetDaemon(booleanon)** AparameteroftruedenotesthisThreadasadaemonthread.
6	**publicfinalvoid join(longmillisec)** Thecurrentthreadinvokesthis methodonasecondthread, causingthecurrentthreadto block untilthe second thread terminatesorthe specified number ofmillisecondspasses.
7	**publicvoidinterrupt()** Interruptsthisthread,causingittocontinueexecutionifitwasblockedforanyreason.
8	**publicfinalbooleanisAlive()** Returnstrue ifthe threadisalive,whichisanytimeafterthethread hasbeenstarted but

beforeitrunstocompletion.

Thepreviousmethodsareinvokedonaparticular Threadobject.Thefollowing methodsinthe Thread class are static. Invoking one of the static methods performs the operation onthe currently running thread.

SN	Methodswith Description
1	**publicstaticvoidyield()** Causesthecurrentlyrunning threadto yield to anyotherthreadsofthe sameprioritythat are waiting to be scheduled.
2	**publicstaticvoidsleep(longmillisec)** Causesthecurrentlyrunningthreadtoblock foratleastthespecified numberof milliseconds.
3	**publicstaticbooleanholdsLock(Objectx)** Returnstrue ifthe currentthreadholdsthelockonthegivenObject.
4	**publicstaticThreadcurrentThread()** Returnsareferencetothecurrentlyrunningthread, whichisthethreadthat invokesthis method.
5	**publicstaticvoiddumpStack()** Printsthestacktraceforthe currentlyrunning thread, which isusefulwhendebugging a multithreaded application.

Example:

ThefollowingThreadClassDemo programdemonstratessomeofthese methodsoftheThread class. Consider a class **DisplayMessage** which implements **Runnable**:

```
//FileName:DisplayMessage.java
//Createathreadto implementRunnable
publicclassDisplayMessage implementsRunnable
{
  privateStringmessage;
  publicDisplayMessage(Stringmessage)
  {
    this.message=message;
  }
  publicvoidrun()
  {
    while(true)
    {
      System.out.println(message);
    }
  }
}
```

Followingisanotherclasswhichextends Thread class:

```
//FileName:GuessANumber.java
//Create athreadtoextentdThread
publicclassGuessANumberextendsThread
{
  privateintnumber;
  publicGuessANumber(int number)
  {
    this.number=number;
  }
  publicvoidrun()
  {
    intcounter=0;
    int guess = 0;
    do
    {
       guess=(int) (Math.random()*100+1);
       System.out.println(this.getName()
                 +"guesses"+guess); counter++;
    }while(guess!=number);
    System.out.println("**Correct!"+this.getName()
                 + "in "+ counter+"guesses.**");
  }
}
```

Followingisthemainprogramwhichmakesuseofabovedefinedclasses:

```
//FileName:ThreadClassDemo.java
public class ThreadClassDemo
{
  publicstaticvoidmain(String[] args)
  {
    Runnablehello=newDisplayMessage("Hello"); Thread
    thread1 = new Thread(hello);
    thread1.setDaemon(true);
    thread1.setName("hello");
    System.out.println("Startinghellothread...");
    thread1.start();

    Runnablebye=newDisplayMessage("Goodbye");
    Thread thread2 = new Thread(bye);
    thread2.setPriority(Thread.MIN_PRIORITY);
    thread2.setDaemon(true);
    System.out.println("Startinggoodbyethread...");
    thread2.start();

    System.out.println("Starting thread3...");
    Threadthread3=newGuessANumber(27);
    thread3.start();
    try
    {
      thread3.join();
    }catch(InterruptedExceptione)
    {
      System.out.println("Threadinterrupted.");
    }
    System.out.println("Starting thread4...");
    Threadthread4=newGuessANumber(75);

            thread4.start();
    System.out.println("main()isending...");
  }
}
```

Thiswouldproducethefollowing result. You cantrythisexampleagainandagainand you would get different result every time.

```
Startinghello thread...
Startinggoodbyethread...
Hello
Hello
Hello
Hello
Hello
Hello
Goodbye
Goodbye
Goodbye
Goodbye
Goodbye
.......
```

IOStream

The java.io packagecontainsnearlyeveryclass youmight everneedto performinput andoutput (I/O) in Java. All these streams representan input source and an output destination. The stream in the java.io package supports manydata such as primitives, Object, localized characters, etc.

Astreamcanbedefined asasequenceofdata.The InputStreamisused to readdatafroma source and the OutputStream is used for writing data to a destination.

Java provides strong but flexible support for I/Orelatedto Files and networks but this tutorial coversverybasic functionalityrelatedto streams andI/O. Wewouldsee most commonlyused example one by one:

ByteStreams

Java byte streams are used to perform input and output of8-bit bytes. Though there are many classesrelatedto bytestreams but the most frequentlyusedclassesare, **FileInputStream**and **FileOutputStream**. Following isanexamplewhichmakesuseofthesetwo classestocopyan input file into an output file:

```
importjava.io.*;

publicclassCopyFile{
  publicstaticvoidmain(Stringargs[])throwsIOException
  {
    FileInputStream in = null;
    FileOutputStreamout=null;

    try{
      in=new FileInputStream("input.txt");
      out=newFileOutputStream("output.txt");

      intc;
      while((c=in.read())!=-1){
        out.write(c);
      }
    }finally{
      if(in !=null){
        in.close();
      }
      if(out!=null){
        out.close();
      }
    }
  }
}
```

Nowlet'shaveafile**input.txt**withthefollowingcontent:

```
Thisistestforcopyfile.
```

Asanext step,compileaboveprogramandexecuteit,whichwillresult increatingoutput.txt file with the same content as we have in input.txt. So let's put above code in CopyFile.java file and do the following:

```
$javacCopyFile.java
$javaCopyFile
```

CharacterStreams

Java**Byte**streamsareusedto performinput andoutputof8-bit bytes, whereasJava**Character** streamsare used to performinput and output for16-bit unicode. Thoughthere are manyclasses related to character streams but the most frequentlyused classes are , **FileReader** and **FileWriter.**. Though internally FileReader uses FileInputStream and FileWriter uses FileOutputStreambut here major difference is that FileReader reads two bytes at a time and FileWriter writes two bytes at a time.

Wecanre-writeaboveexamplewhichmakesuse ofthesetwo classesto copyaninput file (having unicode characters) into an output file:

```
importjava.io.*;

publicclassCopyFile{
  publicstaticvoidmain(Stringargs[])throwsIOException
  {
    FileReader in = null;
    FileWriterout=null;

    try{
      in = new FileReader("input.txt");
      out=newFileWriter("output.txt");

      intc;
      while((c=in.read())!=-1){
        out.write(c);
      }
    }finally{
      if(in !=null){
        in.close();
      }
      if(out!=null){
        out.close();
      }
      }
   }
}
```

Nowlet'shaveafile**input.txt**withthefollowingcontent:

Thisistest forcopyfile.

Asanext step,compileaboveprogramandexecuteit,whichwillresult increatingoutput.txt file with the same content as we have in input.txt. So let's put above code in CopyFile.java file and do the following:

```
$javacCopyFile.java
$javaCopyFile
```

StandardStreams

Alltheprogramming languagesprovide support for standard I/Owhereuser'sprogramcantake input froma keyboard and thenproduceoutput onthecomputer screen. If you areawareifCor C++ programming languages, then you must be aware of three standard devices STDIN, STDOUT and STDERR. Similar way Java provides following three standard streams

- **Standard Input:**This isusedto feedthedatato user'sprogramandusuallyakeyboardis used as standard input stream and represented as **System.in**.
- **Standard Output:**This isusedtooutput thedataproducedbytheuser'sprogramand usuallya computer screen is used to standard output stream and represented as **System.out**.
- **StandardError:** This isusedtooutput theerrordataproducedbytheuser's program and usually a computer screen is used to standard error stream and represented as **System.err**.

Following isasimpleprogramwhichcreates**InputStreamReader**toreadstandard input stream until the user types a "q":

```
importjava.io.*;

publicclassReadConsole{
   publicstaticvoidmain(Stringargs[])throwsIOException
   {
      InputStreamReadercin= null;

      try{
         cin = new InputStreamReader(System.in);
         System.out.println("Entercharacters,'q'toquit."); char
         c;
         do{
            c=(char)cin.read(); System.out.print(c);
         }while(c!='q');
      }finally{
         if(cin!=null){
            cin.close();
         }
      }
   }
}
```

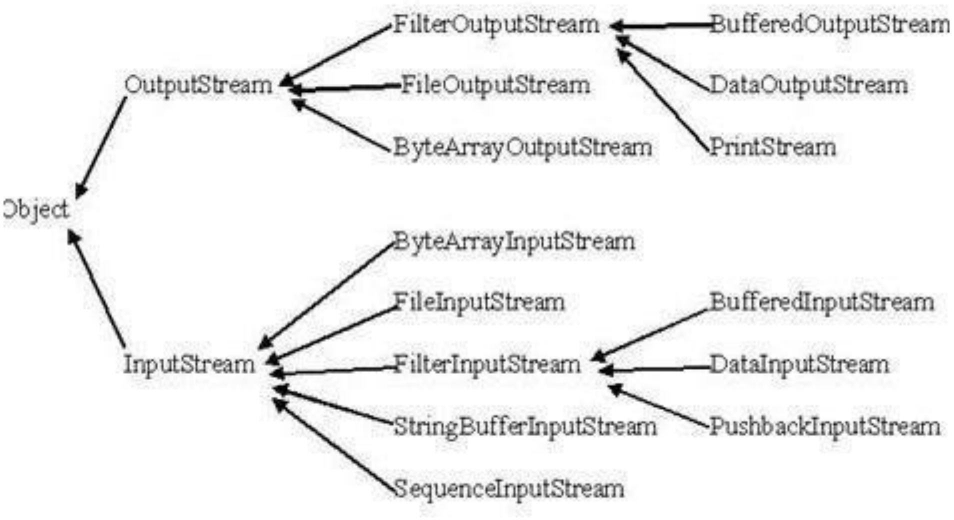
Let'skeepabovecodeinReadConsole.java file andtrytocompileandexecute it asbelow.This program continues reading and outputting same character until we press 'q':

```
$javacReadConsole.java
$javaReadConsole
Entercharacters,'q'toquit. 1
1
e
e
q
q
```

Reading and WritingFiles:

Asdescribedearlier, Astreamcanbedefinedasa sequenceofdata.The**InputStream** isusedto read data froma source and the **OutputStream** is used for writing data to a destination.

Hereisahierarchyofclassestodealwith InputandOutputstreams.



Thetwoimportantstreamsare **FileInputStream**and**FileOutputStream**,whichwouldbe discussed in this tutorial:

FileInputStream:

Thisstreamisused for readingdatafromthe files. Objectscanbecreatedusingthekeywordnew and there are several types of constructors available.

Followingconstructortakesafile nameasastringtocreateaninput streamobject toreadthe file.:

```
InputStreamf=new FileInputStream("C:/java/hello");
```

Followingconstructortakesafileobject tocreateaninput streamobject toreadthe file. First we create a file object using File() method as follows:

```
File f = new File("C:/java/hello");
InputStreamf=newFileInputStream(f);
```

Once youhave*InputStream*object inhand,thenthereisa list ofhelper methodswhichcanbe used to read to stream or to do other operations on the stream.

SN	MethodswithDescription
1	**publicvoidclose()throwsIOException{}** This methodclosesthefileoutputstream. Releasesanysystemresourcesassociatedwiththe file. Throws an IOException
2	**protectedvoidfinalize()throwsIOException{}** This methodcleansuptheconnectiontothefile.Ensuresthattheclose methodofthisfile output stream is called when there are no more references to this stream. Throws an IOException.
3	**publicintread(intr)throwsIOException{}** This methodreadsthespecified byteofdatafromtheInputStream. Returnsanint. Returns the next byte of data and -1 will be returned if it's end of file.
4	**publicintread(byte[]r)throwsIOException{}** This methodreadsr.lengthbytes fromthe input streaminto anarray. Returnsthetotal number of bytes read. If end of file -1 will be returned.
5	**publicintavailable()throwsIOException{}** Givesthenumberofbytesthatcanbereadfromthisfileinputstream.Returnsanint.

Thereareotherimportant input streamsavailable,formoredetailyoucanrefertothefollowing links:

- ByteArrayInputStreamDat
- aInputStream

FileOutputStream:

FileOutputStreamisusedtocreateafileandwrite datainto it.Thestreamwouldcreateafile, if it doesn't already exist, before opening it for output.

Herearetwoconstructorswhichcanbeusedtocreatea FileOutputStreamobject.

Followingconstructortakesafile nameasastringtocreateaninput streamobject towritethe file:

```
OutputStreamf=newFileOutputStream("C:/java/hello")
```

Followingconstructortakesafileobject tocreateanoutput streamobjectto writethefile. First, we create a file object using File() method as follows:

```
File f = new File("C:/java/hello");
OutputStreamf=newFileOutputStream(f);
```

Once youhave*OutputStream*object inhand,thenthereisa list ofhelper methods,whichcanbe used to write to stream or to do other operations on the stream.

SN	MethodswithDescription
1	**publicvoidclose()throwsIOException{}** This methodclosesthefileoutputstream. Releasesanysystemresourcesassociatedwiththe file. Throws an IOException
2	**protectedvoidfinalize()throwsIOException{}** This methodcleansuptheconnectiontothefile.Ensuresthattheclose methodofthisfile output stream is called when there are no more references to this stream. Throws an IOException.
3	**publicvoidwrite(intw)throwsIOException{}** Thismethodwritesthespecifiedbytetotheoutputstream.
4	**publicvoidwrite(byte[]w)** Writesw.lengthbytesfromthementionedbytearraytotheOutputStream.

Thereareotherimportantoutputstreamsavailable,formoredetail youcanrefertothefollowing links:

- ByteArrayOutputStreamDataOutputStr
- eam

Example:

FollowingistheexampletodemonstrateInputStreamandOutputStream: import

```
java.io.*;

publicclass fileStreamTest{

  publicstaticvoidmain(Stringargs[]){

  try{
    bytebWrite[]= {11,21,3,40,5};
    OutputStreamos=newFileOutputStream("test.txt");
    for(int x=0; x < bWrite.length ; x++){
      os.write(bWrite[x]);//writesthebytes
    }
    os.close();

    InputStreamis=newFileInputStream("test.txt"); int
    size = is.available();

    for(int i=0; i< size; i++){
      System.out.print((char)is.read()+"");
    }
    is.close();
  }catch(IOException e){
    System.out.print("Exception");
  }
  }
}
```

Theabovecodewouldcreatefiletest.txtandwouldwritegivennumbersin binaryformat. Same would be output on the stdout screen.

FileNavigation and I/O:

Thereareseveralother classesthat wewouldbegoingthroughto getto knowthebasicsofFile Navigation and I/O.

- FileClass

- FileReaderClass
- FileWriter Class

Directoriesin Java:

Adirectoryisa Filewhichcancontainsa list ofother files and directories. You use **File**object to createdirectories, to list downfiles available in a directory. For completedetailcheck a list ofall the methods which you can callon File object and what are related to directories.

CreatingDirectories:

Therearetwo useful**File**utilitymethods,whichcanbe used tocreatedirectories:

- The**mkdir()** methodcreatesadirectory,returning trueonsuccessandfalseon failure. Failureindicatesthatthepathspecified intheFile object alreadyexists, orthat the directorycannot be created because the entire path does not exist yet.
- The**mkdirs()**methodcreatesbothadirectoryand allthe parentsofthe directory.

Followingexamplecreates"/tmp/user/java/bin"directory:

```
import java.io.File;

publicclassCreateDir{
  public static void main(String args[]) {
    Stringdirname="/tmp/user/java/bin";
    File d = new File(dirname);
    //Createdirectorynow.
    d.mkdirs();
  }
}
```

Compileandexecute above codetocreate"/tmp/user/java/bin".

Note: Java automatically takes care of path separators on UNIX and Windows as per conventions. If youusea forwardslash(/)onaWindowsversionofJava, thepathwillstill resolve correctly.

ListingDirectories:

Youcanuse**list( )** methodprovidedby**File**object to list downallthe filesanddirectories available in a directory as follows:

```
import java.io.File;

publicclassReadDir{
  publicstaticvoidmain(String[]args){
    Filefile=null;
    String[] paths;

    try{
      //createnew fileobject
      file=newFile("/tmp");

      //arrayoffilesanddirectory paths =
      file.list();

      //foreachname inthepatharray for(String
      path:paths)
      {
        //printsfilenameanddirectoryname
        System.out.println(path);
      }
    }catch(Exceptione){
      //ifanyerroroccurs
      e.printStackTrace();
    }
  }
}
```

Thiswouldproducefollowingresultbasedonthedirectoriesandfilesavailableinyour**/tmp** directory:

```
test1.txt
test2.txt
ReadDir.java
ReadDir.class
```

Applet

Anapplet isaJavaprogramthatruns ina Webbrowser. Anapplet canbeafullyfunctionalJava application because it has the entire Java API at its disposal.

Therearesomeimportant differencesbetweenanapplet andastandalone Javaapplication, including the following:

- AnappletisaJavaclassthatextendsthejava.applet.Appletclass.
- Amain() methodisnot invokedonanapplet, and anapplet classwillnotdefine main().
- Applets are designed to be embedded within an HTML page.
- Whenauser viewsanHTMLpagethat containsanapplet,thecodefortheapplet is downloaded to the user's machine.

- AJVM isrequiredtoviewanapplet. The JVMcanbeeither aplug-inoftheWeb browser or a separate runtime environment.
- TheJVMontheuser's machinecreatesaninstanceoftheapplet classand invokes various methods during the applet's lifetime.
- Applets have strict securityrules that are enforced bythe Web browser. The securityof anapplet isoftenreferredto assandboxsecurity,comparingtheapplettoachildplaying in a sandbox with various rules that must be followed.
- Otherclassesthattheapplet needscanbedownloadedinasingleJava Archive(JAR) file.

LifeCycleofanApplet:

FourmethodsintheAppletclassgiveyoutheframeworkonwhichyoubuildanyserious applet:

- **init:**Thismethodisintendedforwhateverinitializationisneededfor yourapplet.Itis called after the param tags inside the applet tag have been processed.
- **start:** This method is automaticallycalled after the browser calls the init method. It is also calledwhenevertheuserreturnstothepagecontainingtheapplet afterhavinggone off to other pages.
- **stop:**This method isautomaticallycalledwhentheuser movesoffthepageonwhichthe applet sits. It can, therefore, be called repeatedly in the same applet.
- **destroy:** This method isonlycalledwhenthebrowsershutsdownnormally. Because appletsare meant to liveonanHTMLpage, you should not normallyleave resources behind after a user leaves the page that contains the applet.
- **paint:** Invoked immediatelyafter the start() method, and also anytime the applet needstorepaint itselfinthe browser.Thepaint()methodisactuallyinherited fromthe java.awt.

A"Hello,World"Applet:

ThefollowingisasimpleappletnamedHelloWorldApplet.java: import

```
java.applet.*;
importjava.awt.*;

publicclassHelloWorldAppletextendsApplet
{
  publicvoidpaint(Graphics g)
  {
    g.drawString("HelloWorld",25,50);
  }
}
```

These importstatementsbringtheclassesintothescopeofourapplet class: java.applet.Applet.

java.awt.Graphics.

Withoutthoseimportstatements,theJavacompilerwouldnotrecognizetheclassesApplet and Graphics, which the applet class refers to.

TheApplet CLASS:

Everyapplet is an extension of the *java.applet.Applet class*. The base Applet class provides methodsthataderived Applet class maycalltoobtaininformationandservices fromthebrowser context.

Theseincludemethodsthatdothefollowing: Get

- applet parameters
- GetthenetworklocationoftheHTMLfilethat containstheapplet Get
- the network location of the applet class directory
- Printastatusmessageinthebrowser
- Fetch an image
- Fetchanaudioclip
- Play an audio clip
- Resize the applet

Additionally, theApplet classprovidesaninterfacebywhichtheviewer orbrowserobtains information about the applet and controls the applet's execution. The viewer may:

- requestinformationabouttheauthor,versionandcopyrightoftheapplet
- request a description of the parameters the applet recognizes
- initializetheapplet
- destroy the applet
- starttheapplet'sexecution
- stoptheapplet'sexecution

TheApplet classprovidesdefault implementationsofeachofthese methods.Those implementations may be overridden as necessary.

The"Hello, World"applet iscompleteasit stands. Theonlymethodoverriddenisthepaint method.

Invokingan Applet:

Anapplet maybe invokedbyembeddingdirectives inanHTMLfileandviewingthefile through an applet viewer or Java-enabled browser.

The<applet>tagisthebasis for embeddinganapplet inanHTMLfile. Below isanexamplethat invokes the "Hello, World" applet:

```
<html>
<title>TheHello,WorldApplet</title>
```

```
<hr>
<appletcode="HelloWorldApplet.class"width="320"height="120"> If
your browser was Java-enabled, a "Hello, World"
messagewouldappearhere.
</applet>
<hr>
</html>
```

Basedontheaboveexamples, hereistheliveappletexample:AppletExample.

Note: Youcanreferto HTMLAppletTagtounderstandmoreaboutcallingappletfromHTML.

The code attribute ofthe <applet>tag is required. It specifiesthe Applet class to run. Widthand height arealso requiredtospecifythe initialsizeofthepanelinwhichanapplet runs. Theapplet directive must be closed with a </applet> tag.

Ifanapplet takesparameters, values maybepassedfortheparametersbyadding<param>tags between <applet> and </applet>. The browser ignores text and other tags between the applet tags.

Non-Java-enabled browsersdonot process<applet>and</applet>.Therefore,anythingthat appears between the tags, not related to the applet, is visible in non-Java-enabled browsers.

Theviewerorbrowserlooksforthecompiled Java codeatthelocationofthedocument.To specifyotherwise, use the codebase attribute of the <applet> tag as shown:

```
<applet codebase="http://amrood.com/applets"
code="HelloWorldApplet.class"width="320"height="120">
```

Ifanapplet residesinapackageotherthanthedefault,theholdingpackage must bespecified in the code attribute using the period character (.) to separate package/class components. For example:

```
<appletcode="mypackage.subpackage.TestApplet.class"
        width="320" height="120">
```

GettingAppletParameters:

The following example demonstrates how to make an applet respond to setup parameters specified inthedocument. Thisapplet displaysacheckerboardpatternofblackandasecond color.

Thesecondcolor andthesizeofeachsquare maybespecifiedasparameterstotheapplet within the document.

CheckerApplet getsitsparametersinthe init()method. Itmayalso get itsparametersinthe paint() method. However, getting the values and saving the settings once at the start ofthe applet, instead of at everyrefresh, is convenient and efficient.

Theapplet viewerorbrowsercallsthe init() methodofeachapplet it runs.Theviewercallsinit() once, immediatelyafter loading the applet. (Applet.init() is implemented to do nothing.) Override the default implementation to insert custom initialization code.

TheApplet.getParameter()methodfetchesaparametergiventheparameter's name(thevalueof a parameter is always a string). If the value is numeric or other non-character data, the string must be parsed.

Thefollowing isaskeletonofCheckerApplet.java:

```
import java.applet.*;
importjava.awt.*;
publicclassCheckerAppletextends Applet
{
  intsquareSize=50;//initializedtodefaultsize public
  void init () {}
  privatevoidparseSquareSize(Stringparam){}
  private Color parseColor (String param) {}
  public void paint (Graphics g) {}
}
```

HereareCheckerApplet'sinit()andprivateparseSquareSize()methods: public

```
void init ()
{
  StringsquareSizeParam=getParameter("squareSize");
  parseSquareSize (squareSizeParam);
  StringcolorParam=getParameter("color");
  Color fg = parseColor (colorParam);
  setBackground (Color.black);
  setForeground (fg);
}
privatevoidparseSquareSize(Stringparam)
{
  if(param==null)return;
  try{
    squareSize=Integer.parseInt(param);
  }
  catch(Exceptione){
   //Letdefaultvalueremain
  }
}
```

Theapplet callsparseSquareSize()to parsethesquareSizeparameter.parseSquareSize()callsthe librarymethod Integer.parseInt(), which parses a string and returns an integer. Integer.parseInt() throws an exception whenever its argument is invalid.

Therefore,parseSquareSize()catchesexceptions, ratherthanallowingtheapplet tofailonbad input.

Theapplet calls parseColor()to parsethecolor parameter into aColor value. parseColor() doesa series ofstring comparisonsto matchthe parameter value to the name ofa predefined color. You need to implement these methods to make this applet works.

SpecifyingApplet Parameters:

Thefollowing isanexampleofanHTML filewithaCheckerApplet embedded init. TheHTML file specifies both parameters to the applet by means of the <param> tag.

```
<html>
<title>CheckerboardApplet</title>
<hr>
<appletcode="CheckerApplet.class"width="480"height="320">
<paramname="color" value="blue">
<paramname="squaresize" value="30">
</applet>
<hr>
</html>
```

Note: Parameter namesarenotcase sensitive.

ApplicationConversiontoApplets:

It iseasyto convert agraphicalJavaapplication(that is, anapplicationthat usestheAWT and that you canstart withthe java programlauncher) into anapplet that you canembed ina web page.

Herearethespecificstepsforconverting anapplicationtoanapplet.

- MakeanHTMLpage withthe appropriatetagtoloadthe appletcode.
- SupplyasubclassoftheJApplet class. Makethisclasspublic. Otherwise, theapplet cannot be loaded.
- Eliminatethe mainmethod intheapplication.Do notconstruct aframewindowforthe application. Your application will be displayed inside the browser.
- Moveanyinitializationcodefromthe framewindowconstructortothe init methodofthe applet. You don't need to explicitlyconstruct the applet object.the browser instantiates it for you and calls the init method.
- Removethecallto setSize;forapplets,sizing isdonewiththewidthand height parameters in the HTML file.

- RemovethecalltosetDefaultCloseOperation.Anapplet cannotbeclosed;it terminates when the browser exits.
- Iftheapplicationcalls setTitle, eliminatethecalltothe method. Appletscannothavetitle bars. (You can, of course, title the web page itself, using the HTML title tag.)
- Don'tcallsetVisible(true).Theapplet isdisplayedautomatically.

EventHandling:

Applets inherit agroupofevent-handling methodsfromtheContainerclass.TheContainerclass defines several methods, such as processKeyEvent and processMouseEvent, for handling particular types of events, and then one catch-all method called processEvent.

Inorderto reactanevent,anapplet mustoverridetheappropriateevent-specific method. import

```
java.awt.event.MouseListener;
importjava.awt.event.MouseEvent;
import java.applet.Applet;
importjava.awt.Graphics;

publicclassExampleEventHandlingextendsApplet
                                        implementsMouseListener{

   StringBufferstrBuffer;

   public void init() {
            addMouseListener(this);
            strBuffer=newStringBuffer();
      addItem("initializingtheapple");
   }

   publicvoid start(){
      addItem("startingtheapplet");
   }

   publicvoid stop(){
      addItem("stoppingtheapplet");
   }

   publicvoiddestroy(){
      addItem("unloadingtheapplet");
   }

   voidaddItem(Stringword){
      System.out.println(word);
      strBuffer.append(word);
      repaint();
```

```
    }

    publicvoidpaint(Graphicsg){
              //DrawaRectanglearoundtheapplet'sdisplayarea.
        g.drawRect(0, 0,
                              getWidth()-1,
                              getHeight()-1);

              //displaythestringinsidetherectangle.
        g.drawString(strBuffer.toString(), 10, 20);
    }


    publicvoidmouseEntered(MouseEventevent){
    }
    publicvoidmouseExited(MouseEventevent){
    }
    publicvoidmousePressed(MouseEventevent){
    }
    publicvoidmouseReleased(MouseEventevent){
    }

    publicvoidmouseClicked(MouseEventevent){
              addItem("mouse clicked! ");
    }
}
```

Now, letuscallthisappletasfollows:

```
<html>
<title>EventHandling</title>
<hr>
<appletcode="ExampleEventHandling.class"
width="300" height="300">
</applet>
<hr>
</html>
```

Initially,theapplet willdisplay"initializingtheapplet. Startingtheapplet."Thenonce youclick inside the rectangle "mouse clicked" will be displayed as well.

Basedontheaboveexamples,hereistheliveappletexample:AppletExample.

DisplayingImages:

Anapplet candisplayimagesoftheformat GIF, JPEG, BMP,andothers.To displayanimage within the applet, you use the drawImage() method found in the java.awt.Graphics class.

Following istheexampleshowingallthestepsto showimages:

```
import java.applet.*;
importjava.awt.*;
importjava.net.*;
publicclassImageDemoextendsApplet
{
  privateImageimage;
  privateAppletContextcontext;
  public void init()
  {
      context=this.getAppletContext();
      StringimageURL=this.getParameter("image");
      if(imageURL == null)
      {
         imageURL="java.jpg";
      }
      try
      {
         URLurl=newURL(this.getDocumentBase(),imageURL);
         image = context.getImage(url);
      }catch(MalformedURLExceptione)
      {
         e.printStackTrace();
         // Display in browser status bar
         context.showStatus("Couldnotloadimage!");
      }
   }
   publicvoidpaint(Graphicsg)
   {
      context.showStatus("Displayingimage");
      g.drawImage(image, 0, 0, 200, 84, null);
      g.drawString("www.javalicense.com",35,100);
   }
}
```

Now, letuscallthisappletasfollows:

```
<html>
<title>TheImageDemoapplet</title>
<hr>
```

```
<appletcode="ImageDemo.class"width="300"height="200">
<paramname="image"value="java.jpg">
</applet>
<hr>
</html>
```

Basedontheaboveexamples, hereistheliveappletexample:AppletExample.

PlayingAudio:

Anapplet canplayanaudio filerepresentedbytheAudioClip interface inthe java.applet package. The AudioClip interface has three methods, including:

- **publicvoidplay():**Playstheaudio cliponetime,fromthebeginning.
- **public void loop():** Causes the audio clip to replay continually.**public**
- **void stop():** Stops playing the audio clip.

ToobtainanAudioClipobject,you must invokethegetAudioClip()methodoftheApplet class. The getAudioClip() method returns immediately, whether or not the URL resolves to an actual audio file. The audio file is not downloaded until an attempt is made to playthe audio clip.

Following istheexample showingallthestepstoplayanaudio:

```
import java.applet.*;
importjava.awt.*;
importjava.net.*;
publicclassAudioDemoextendsApplet
{
  privateAudioClipclip;
  privateAppletContextcontext;
  public void init()
  {
    context=this.getAppletContext();
    StringaudioURL=this.getParameter("audio");
    if(audioURL == null)
    {
      audioURL="default.au";
    }
    try
    {
      URLurl=newURL(this.getDocumentBase(),audioURL); clip
      = context.getAudioClip(url);
    }catch(MalformedURLExceptione)
    {
      e.printStackTrace();
      context.showStatus("Couldnotloadaudio file!");
```

```
          }
 }
 publicvoidstart()
 {
   if(clip!=null)
    {
       clip.loop();
    }
 }
 publicvoidstop()
 {
  if(clip!=null)
  {
       clip.stop();
    }
```

Now, letuscallthisappletasfollows:

```
<html>
<title>TheImageDemoapplet</title>
<hr>
<appletcode="ImageDemo.class"width="0"height="0">
<paramname="audio" value="test.wav">
</applet>
<hr>
</html>
```

Unit–II

UnderstandingInternetBasics

You canprogramforthe Web, using your skillsas a VisualBasic programmer, no matterwhat your levelofexperience with Internet technology. If you are new to the Internet or unfamiliar withits technology, VisualBasic allows youto quicklyandeasilyproducefunctionalapplications. If you are more experienced with Internet technology, you can work at a more advanced level.

Fromoneperspective,Internettechnologysimply providesanotherareaforyourdevelopmentefforts.When youdeploy Internetapplicationson theWeb,youmay goaboutitdifferently —incorporatingHTML pages withyourVisual Basiccode,providingsecurity features,andsoon—butyou'restillcallingmethods,setting properties,andhandlingevents.In thisway,allofyourknowledgeasaVisual Basicdevelopercanbecarried intotheInternetarena.

Fromanotherperspective, applying Internet technologyenablesyou to extendyour development skills in excitingnewways. For example, writing VisualBasic codethatmanipulatesHTMLpagesallowsyouto decreasedeploymentcosts,reduceclientmaintenanceproblems,andreach thebroadaudienceof theInternet.

InternetClientsandServers

A common wayto think aboutInternetdevelopmentis in terms of client/server relationships. In this case, the clientis the browser, and the server is the Web server. Mostinteractions on the Internet or an intranetcan be thought of interms ofrequests and responses.The browser makes a request to the Webserver (usuallyto displaya page the user wantstosee) and the Web server returns a response (usuallyanHTMLpage,an element, or an image) to the browser.

Internetvs.Intranet

TheInternetencompassestwocategories:theInternetandtheintranet.TheInternetisaglobal,distributed networkofcomputersoperatingonaprotocolcalledTCP/IP.Anintranetisalsoanetworkofcomputers operatingontheTCP/IPprotocol,butitisnotglobal.Generally,intranetsarerestrictedtoaparticularsetof usersand arenot accessible bytheoutside world.For example,manycorporationsusea corporateintranet to provideinformationtotheiremployees,andrunanotherInternetsiteforexternalusers.Userswithinthe companycan access both the intranetsites and the Internet, but users outside the company can access only the company'sInternetsites.

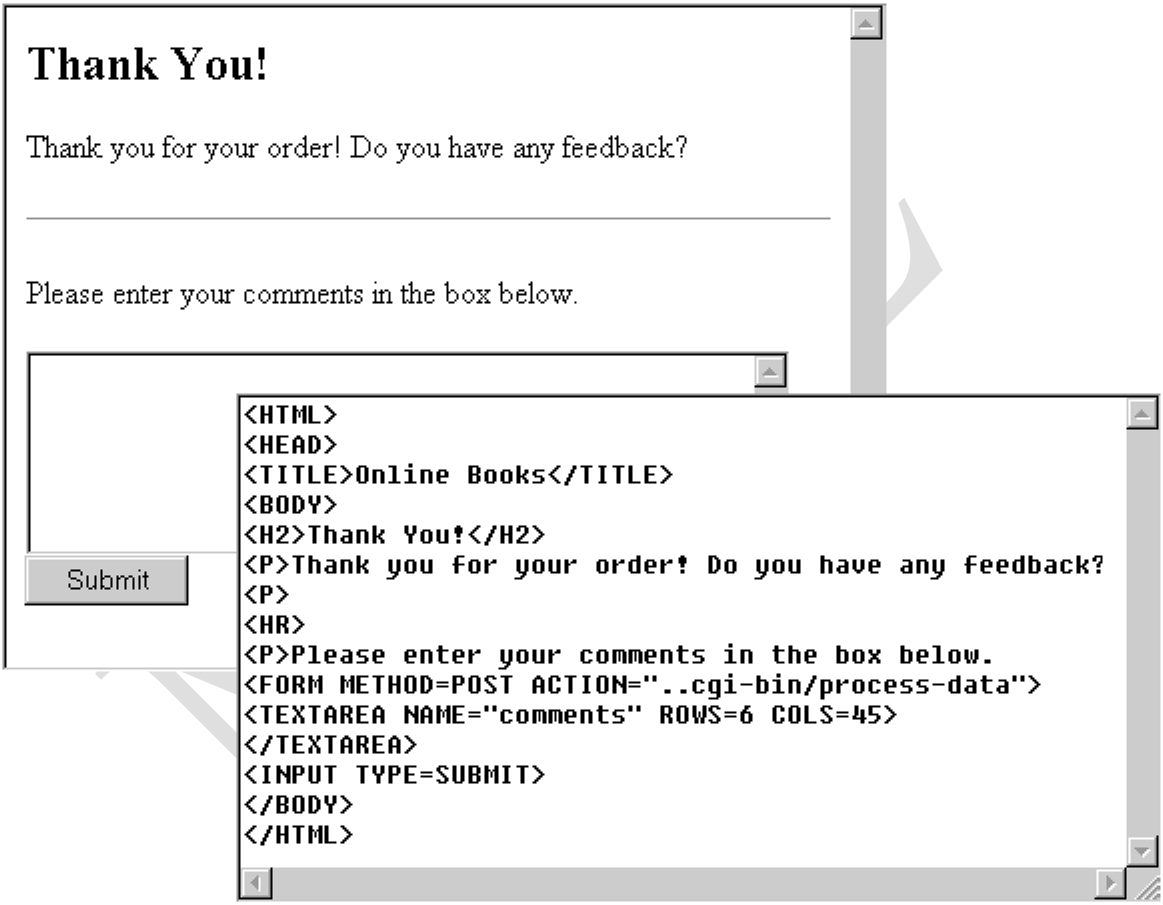
HTMLPages

HTML(HyperText MarkupLanguage) isa languagethat allows youtodisplaydocumentsina Webbrowser. YouuseHTMLto create.htmfilesthat aredisplayed inabrowser.WhenyoucreateanInternet applicationin VisualBasic,youruser interfaceisusuallymadeupofHTMLpagesratherthanforms. Inmanyways,an.htm file (whichallows youto displayHTMLpages) is similar to aVisualBasic .frm file (which allows youto displayaVisualBasicform).

NoteWhile the user interface is generallymade upofHTMLpages, it canalso containa mix ofVisualBasic formsandHTMLpages.

An.htmfile is a text document that contains a series oftagsthattellthe browser how to displaythe file. These HTMLtagssupplyinformationaboutthepage'sstructure,appearance,andcontent.Thefollowingfigureshows the relationship between page in the browser and its HTML tags:

HTMLPageandSourceHTML



Thank You!

Thank you for your order! Do you have any feedback?

Please enter your comments in the box below.

```
<HTML>
<HEAD>
<TITLE>Online Books</TITLE>
<BODY>
<H2>Thank You!</H2>
<P>Thank you for your order! Do you have any feedback?
<P>
<HR>
<P>Please enter your comments in the box below.
<FORM METHOD=POST ACTION="..cgi-bin/process-data">
<TEXTAREA NAME="comments" ROWS=6 COLS=45>
</TEXTAREA>
<INPUT TYPE=SUBMIT>
</BODY>
</HTML>
```

In addition todescribingthestructural relationships amongpageelements,someHTML tags alsocontain attributes.Attributesprovidedetailsaboutaparticulartag.Forexample,thetagthatinsertsanimageontoa pagecontainsan attribute thatspecifies thenameof thefiletoinsert. The tag isshownbelow.

HTMLTagsandAttributes

The image part with relationship ID rId31 was not found in the file.

InternetObjectModels

Youusetheconceptsofobject-orientedprogramminginyourVisualBasicInternetapplicationsjustasyoudo informs-basedVisualBasicapplications.InVisualBasicInternetapplications,youuseInternet-relatedobject modelstoaccessandmanipulateinformationandcontrolsonyourHTMLpages.

There aretwo types ofVisual Basic Internet applications: IIS applications and DHTMLapplications. In IIS applications, you make use ofthe Active Server Pages (ASP) object model to retrieve information fromthe user, send information to the browser, and maintain information about the current session. In DHTML applications, youusetheDynamicHTML(DHTML) object modelto manipulatetheelementsonanHTML page.

The important pointto remember isthat youaccesstheinformationon yourHTMLpagesthroughobjects, regardlessofwhethertheobjectsthemselvesareASPorDHTML.Theobjectmodelsareexplainedinmuch greater detail in the chapters describing each type ofapplication.

ForMoreInformation See "A History of Development on the Internet" for more information on the differences between IIS and DHTML applications. See the "Developing DHTML Applications" chapter for more information on using Dynamic HTML objects. See the "Developing IIS Applications with Webclasses" chapter for more informationonusing ASP objects. Seethe MSDN™ Website at http://msdn.microsoft.com/fordetailsonusingHTMLandInternettechnologies.

Webpage

Adocumentdisplayableinawebbrowser

Website

Acollectionofwebpages

Web Server

Acomputerthathostsawebsite

SearchEngine

Awebsitethathelpsyoufindwebpages

HTMLBasics

WelcometoHTMLBasics.ThisworkshopleadsyouthroughthebasicsofHyperText MarkupLanguage (HTML). HTML is the buildingblockfor web pages. You will learn touse HTML toauthor an HTMLpagetodisplayinawebbrowser.

Objectives:

Bythe endofthisworkshop,you willbe ableto:

- Usea texteditortoauthoranHTMLdocument.
- Beableto usebasictagstodenoteparagraphs,emphasisorspecialtype.
- Createhyperlinks tootherdocuments.
- Create anemaillink.
- Addimagestoyourdocument.
- Useatableforlayout.
- ApplycolorstoyourHTMLdocument.

Prerequisites:

Youwill needatexteditor,suchasNotepadandanInternetbrowser,suchasInternet Explorer orNetscape.

Q:WhatisNotepadandwheredoIgetit?

A:Notepadisthedefault Windowstexteditor.OnmostWindowssystems,click yourStart buttonand choose Programs then Accessories. It should be a little bluenotebook.

MacUsers: SimpleTextisthedefaulttexteditorontheMac. InOSXuseTextEditand change the following preferences: Select (in the preferences window) Plain text instead of Rich textand then select Ignorerichtext commandsinHTMLfiles.Thisisveryimportantbecause if youdon't dothis HTML codes probably won't work.

Onethingyoushouldavoidusingisawordprocessor(likeMicrosoftWord)forauthoringyour HTMLdocuments.

WhatisanhtmlFile?

HTML is a format that tells a computer how to display a web page. The documents themselves are plain text fileswithspecial"tags" or codesthata webbrowser usestointerpretanddisplayinformationon your computer screen.

- HTMLstandsforHyperTextMarkupLanguage
- AnHTMLfileisatextfilecontainingsmallmarkuptags
- ThemarkuptagstelltheWebbrowserhowtodisplaythepage
- AnHTMLfilemusthaveanhtmor htmlfileextension

```
 <html>
 <head>
 <title>MyFirstWebpage</title>
 </head>
<body>
This is myfirsthomepage.<b>Thistextis bold</b>
</body>
</html>
```

Savethefileas**mypage.html**.StartyourInternetbrowser.Select**Open**(orOpenPage)inthe **File** menuof ourbrowser.Adialogboxwillappear.Select **Browse**(orChooseFile)and locatethehtmlfile you just reated- **mypage.html**-selectitandclick**Open**.

HTMorHTMLExtension?

When you save an HTML file, you can use either the .htmor the .html extension. The .htmextension comes fromthe past when some ofthe commonlyused software onlyallowed three letter extensions.It is perfectly safe to use either .html or .htm, but be consistent. **mypage.htm** and mypage.htmlare treated as different files bythebrowser.

HowtoViewHTMLSource

AgoodwaytolearnHTMListolookathowotherpeoplehavecodedtheirhtmlpages. Tofindout,simply clickontheViewoptionin yourbrowserstoolbarandselect SourceorPageSource.Thiswillopena window thatshows youtheactualHTMLofthe page.Goaheadand view the source html for thispage.

HTMLTags

- HTMLtagsareusedtomark-upHTMLelements
- HTMLtagsare surroundedbythetwocharacters< and>
- Thesurroundingcharactersarecalledangle brackets
- HTMLtagsnormallycomeinpairslike <b>and</b>
- Thefirsttaginapairisthestart tag,thesecond tagistheendtag
- Thetextbetweenthestartand endtagsistheelement content

- HTMLtagsarenotcasesensitive, <b>meansthesameas<B>

Logicalvs.PhysicalTags

In HTMLthere are both logicaltags and physical tags. Logicaltags are designed to describe (tothe browser) the enclosed text's meaning. An exampleofa logicaltagis the<strong></strong>tag. Byplacing textin betweenthese tags you aretelling the browser that thetext has somegreater importance. Bydefault all browsers make the text appear bold when in between the <strong>and </strong>tags.

Physicaltagsontheother hand provide specific instructionsonhow to displaythe text theeynclose. Examples ofphysicaltagsinclude:

- <b>:Makesthe textbold.
- <big>:Makesthetextusuallyonesizebigger thanwhat'saround it.
- <i>:Makestextitalic.

Physicaltagswereinventedtoadd style to HTMLpages because style sheets were not around, thoughthe originalintentionofHTMLwastonothavephysicaltags.Ratherthanusephysicaltagstostyle your HTMLpages,youshouldusestylesheets.

HTMLElements

RemembertheHTMLexamplefromthepreviouspage:

```
<html>
<head>
<title>MyFirstWebpage</title>
</head>
<body>
Thisismyfirsthomepage.<b>Thistextisbold</b>
</body>
</html>
```

ThisisanHTMLelement:

<b>**Thistextisbold**</b>

TheHTMLelementbeginswithastart tag:<b>
ThecontentoftheHTMLelement is:Thistext isboldThe
HTML element ends with an end tag:</b>

Thepurposeofthe<b>tagistodefineanHTMLelementthatshouldbedisplayedasbold. This is also an HTML element:

<body>
Thisismyfirsthomepage. <b>**Thistext isbold**</b>

</body>

ThisHTMLelementstartswiththestarttag<body>,andendswiththeendtag</body>.Thepurposeofthe <body>tagistodefinetheHTMLelementthatcontainsthebodyoftheHTMLdocument.

NestedTags

Youmayhavenoticedinthe exampleabove,the<body>tagalsocontainsothertags,likethe<b>tab.When youencloseanelementinwithmultipletags,thelasttagopenedshouldbethefirsttagclosed.Forexample:

<p><b><em>***ThisisNOTtheproperwaytoclosenestedtags***.</p></em></b>

<p><b><em>***Thisisthe properwaytoclosenestedtags.***</em></b></p>

Note: Itdoesn'tmatterwhichtagisfirst,buttheymustbeclosedintheproperorder.

WhyUseLowercaseTags?

Youmaynoticewe'veusedlowercasetagseventhoughIsaidthat HTMLtagsarenotcasesensitive. <B>meansthesameas<b>.TheWorldWideWebConsortium(W3C),thegroupresponsiblefor developingwebstandards,recommendslowercasetagsintheirHTML4recommendation,andXHTML (the next generation HTML) requires lowercase tags.

TagAttributes

Tagscanhaveattributes.AttributescanprovideadditionalinformationabouttheHTMLelementsonyour page.The<tag>tellsthebrowsertodosomething,whiletheattributetellsthebrowserhowtodoit.For instance,ifweaddthebgcolorattribute,wecantellthebrowserthatthebackgroundcolorofyourpage should beblue, likethis:<bodybgcolor="blue">
ThistagdefinesanHTMLtable:<table>.Withanaddedborderattribute,youcantellthebrowserthatthe tableshouldhavenoborders:<tableborder="0">.Attributesalwayscomeinname/valuepairslikethis: name="value". Attributes are always added to the start tag of an HTML element and the value is surrounded byquotes.

BasicHTML Tags

ThemostimportanttagsinHTMLaretagsthatdefineheadings,paragraphsandlinebreaks.

BasicHTML Tags

Tag	Description
<html>	DefinesanHTMLdocument

<body>	Definesthedocument'sbody
<h1>to<h6>	Definesheader1toheader6
<p>	Definesaparagraph
<br>	Insertsasingle linebreak
<hr>	Definesahorizontalrule
<!-->	Definesacomment

Headings

Headingsaredefinedwiththe<h1>to<h6>tags.<h1>definesthelargestheadingwhile<h6>definesthe smallest.

<h1>Thisisaheading</h1>

<h2>Thisisaheading</h2>

<h3>Thisisaheading</h3>

<h4>Thisisaheading</h4>

<h5>Thisisaheading</h5>

<h6>Thisisaheading</h6>

HTMLautomaticallyaddsanextrablanklinebeforeandafteraheading.Ausefulheadingattributeisalign.

<h5align="left">I canalignheadings</h5>

<h5align="center">Thisisacenteredheading</h5>

<h5align="right">Thisisa headingalignedtotheright</h5>

Paragraphs

Paragraphsaredefined withthe<p>tag.Think ofaparagraphasablockoftext.Youcanusethealign attribute with a paragraph tag aswell.

<palign="left">This isaparagraph</p>

<palign="center">thisisanotherparagraph</p>

Important: You must indicate paragraphswith <p>elements.A browser ignores any indentationsor blank linesin the sourcetext. Without <p>elements,the documentbecomes onelargeparagraph.HTMLautomaticallyaddsanextrablanklinebeforeandafteraparagrap

LineBreaks

The<br>tagisusedwhenyouwanttostartanewline,butdon'twanttostartanewparagraph. The <br>tagforcesalinebreakwhereveryouplaceit. Itissimilartosinglespacinginadocument.

ThisCode	WouldDisplay
<p>This<br>isapara<br>graphwithline breaks</p>	This<br>isapara<br>graphwithlinebreaks

The<br>taghasnoclosingtag.

HorizontalRule

The<hr>elementisusedforhorizontalrulesthatactasdividersbetweensections,likethis:

Thehorizontalruledoesnothaveaclosingtag.Ittakesattributessuchasalignandwidth.Forinstance:

ThisCode	WouldDisplay
<hrwidth="50%"align="center">	

Commentsin HTML

The comment tag is used to insert a comment in the HTML source code. A comment can beplaced anywhere in the document and the browser will ignore everything inside the brackets. You canuse commentstowritenotestoyourself,orwriteahelpfulmessagetosomeonelookingatyoursourcecode.

ThisCode	WouldDisplay
<p>Thishtmlcomment would<!--Thisisa comment --> be displayed like this.</p>	ThisHTMLcommentwouldbedisplayedlike this.

Notice you don't seethetext betweenthetags<!--and-->. Ifyou look atthesourcecode, you wouldseethe comment. To view the source code for this page, in your browser window, select **View** and then select **Source.**

Note: Youneedanexclamationpointaftertheopeningbracket<!––butnotbeforetheclosing bracket––>.

HTML automaticallyadds an extra blanklinebeforeandaftersome elements,likebeforeandaftera paragraph,andbeforeandafteraheading.If youwant toinsertblanklinesintoyour document,usethe <br>tag.

OtherHTMLTags

Asmentionedbefore,therearelogicalstylesthatdescribewhatthetextshouldbeandphysicalstyleswhich actuallyprovide physical formatting. It is recommended to use the logical tags and usestyle sheets to style thetext inthosetags.

LogicalTags

Tag	Description
<abbr>	Definesanabbreviation
<acronym>	Definesanacronym
<address>	Definesanaddresselement
<cite>	Definesa*citation*
<code>	Definescomputercodetext
<blockquot	e>Definesalongquotation
<del>	Definestext
<dfn>	Definesa*definition*term
<em>	Defines*emphasized*text
<ins>	Definesinsertedtext
<kbd>	Defineskeyboardtext
<pre>	Definespreformattedtext
<q>	Definesashortquotation
<samp>	Definessamplecomputercode
<strong>	Defines**strong**text
<var>	Definesa*variable*

PhysicalTags

Tag	Description
<b>	Defines**bold**text
<big>	Definesbigtext
<i> <small>	Defines*italic*text Definessmalltext
<sup>	Defines$^{\text{superscripted}}$text
<sub>	Definessubscriptedtext
<tt>	Definesteletypetext
<u>	Deprecated.Usestylesinstead

Character tags like <strong>and <em>produce the same physical display as <b>and <i>but are more uniformlysupportedacross differentbrowsers.

HTMLCharacterEntities

SomecharactershaveaspecialmeaninginHTML,likethelessthansign(<)thatdefinesthestartofan HTMLtag.Ifwewant the browsertoactuallydisplaythesecharacterswemustinsertcharacterentitiesin placeoftheactualcharactersthemselves.

MostCommonCharacterEntities:

sult	Description	EntityName	EntityNumber
	non-breaking space	&nbsp;	&#160;
<	lessthan	&lt;	&#60;
>	greaterthan	&gt;	&#62;
&	ampersand	&amp;	&#38;
"	quotationmark	&quot;	&#34;
'	apostrophe	&apos;(doesnotworkinIE)	&#39;

Acharacterentityhasthreeparts:anampersand(&),anentityname oranentitynumber,andfinallya icolon(;).The& meanswearebeginningaspecialcharacter,the**;** meansendingaspecialcharacter andthelettersinbetweenaresortofanabbreviationforwhatit'sfor.Todisplayalessthansigninan HTMLdocumentwemustwrite:**&lt;**or**&#60;**Theadvantageofusinganameinsteadofanumberistha anameiseasiertoremember.Thedisadvantageisthatnotallbrowserssupportthenewestentitynames, whilethesupportforentitynumbersisverygoodinalmostallbrowsers.

Note: Entitiesarecasesensitive.

Non-breakingSpace

Themostcommoncharacterentityin HTMListhenon-breakingspace**&nbsp;**.NormallyHTMLwill truncatespaces inyourtext. Ifyou add 10 spaces in your text,HTMLwillremove9 ofthem. To add spaces toyourtext,usethe&nbsp;character entity.

ThisCode	WouldDisplay
This code wouldappear his.</p>	Thiscodewouldappearasthis.

ThisCode	WouldDisplay
<p>Thiscode&nbsp;&nbsp;&nbsp;would appear with three extra spaces.</p>	Thiscode wouldappearwiththreeextra spaces.

HTMLFonts

The<font>taginHTMLisdeprecated.TheWorldWideWebConsortium(W3C)hasremovedthe <font>tag from its recommendations. In future versions ofHTML, style sheets (CSS) will be used to define thelayoutanddisplaypropertiesofHTMLelements.The<font>TagShould**NOT**beused.

HTMLBackgrounds

Backgrounds

The<body>taghastwoattributeswhereyou canspecifybackgrounds.The background canbeacolororan mage.

Bgcolor

The bgcolor attribute specifies a background-color for an HTML page. The value of this attribute can be a hexadecimalnumber,anRGBvalue,oracolorname:

```
<bodybgcolor="#000000">
<bodybgcolor="rgb(0,0,0)">
<bodybgcolor="black">
```

Thelinesaboveallsetthebackground-colortoblack.

Background

Thebackgroundattributecanalso specifya background-image foranHTMLpage.Thevalueofthis attributeistheURLoftheimageyouwanttouse.Iftheimageissmallerthanthebrowserwindow,the imagewillrepeatitselfuntilitfillstheentirebrowserwindow.

```
<bodybackground="clouds.gif">
<bodybackground="http://profdevtrain.austincc.edu/html/graphics/clouds.gif">
```

TheURLcanberelative(as inthefirst lineabove)orabsolute(asinthesecond lineabove). If you want to use a background image, you should keep in mind:

- Willthebackgroundimageincreasetheloadingtimetoomuch?
- Willthebackgroundimagelook goodwithotherimagesonthepage?
- Willthebackgroundimagelookgoodwiththetextcolorsonthepage?
- Willthebackgroundimagelook goodwhenitisrepeatedonthepage?
- Willthe backgroundimage takeawaythefocusfromthetext?

```
<html>
<head>
<title>MyFirstWebpage</title>
</head>
<body
background="http://profdevtrain.austincc.edu/html/graphics/clouds.gif"bgcolor="#EDDD9E">
<h1align="center">MyFirstWebpage</h1>
<p>Welcome to my<strong>first</strong> webpage. I amwriting this page using atexteditor and plain
oldhtml.</p>
<p>Bylearninghtml,I'llbe able tocreate webpageslike a<del>beginner</del> pro……… <br>
whichI amofcourse.</p>
</body>
</html>
```

Saveyourpageas**mypage3.html**andviewitin yourbrowser.Toviewhowthepageshouldlook,visit thiswebpage:**http://profdevtrain.austincc.edu/html/mypage3.html**

Noticewegaveourpageabackgroundcoloraswellasabackgroundimage.Ifforsomereasontheweb page isunable to find the picture,it willdisplayour backgroundcolor.

LColors

Values

rs are defined using a hexadecimal notation forthe combinationofred, green, and bluecolorvalues (RGB).Thelowest valuethat canbegiventoonelight sourceis0(hex#00).Thehighestvalue is255 (hex #FF).Thistableshowstheresultofcombiningred,green,and blue:

	ColorHEX	**Color RGB**
⬛	#000000	rgb(0,0,0)
🟥	#FF0000	rgb(255,0,0)
🟩	#00FF00	rgb(0,255,0)
🟦	#0000FF	rgb(0,0,255)
🟨	#FFFF00	rgb(255,255,0)
🟦	#00FFFF	rgb(0,255,255)
🟪	#FF00FF	rgb(255,0,255)
▒	#C0C0C0	rgb(192,192,192)
⬜	#FFFFFF	rgb(255,255,255)

Names

A collection ofcolor names is supported bymost browsers. To view a table ofcolor names that are supportedbymost browsers visit this webpage: **http://profdevtrain.austincc.edu/html/color_names.htm**

Note: Only16 **colornames** are supportedbytheW3C HTML 4.0 standard(aqua,black,blue, fuchsia,gray,green,lime,maroon,navy,olive,purple,red,silver,teal,white,andyellow).For all other colors you should use the **Color HEX**value.

	ColorHEX	ColorName
⬜	#F0F8FF	AliceBlue
⬜	#FAEBD7	AntiqueWhite
🟩	#7FFFD4	Aquamarine
⬛	#000000	Black
🟦	#0000FF	Blue
🟪	#8A2BE2	BlueViolet
🟫	#A52A2A	Brown

Web SafeColors

Afewyearsago,whenmostcomputerssupportedonly256differentcolors,alistof216 Web SafeColors was suggested as a Web standard. The reasonfor this was that the Microsoft and Macoperating systemused 40 different"reserved"fixed systemcolors (about20each).This 216cross platformwebsafecolorpalette wasoriginallycreatedtoensure that allcomputers would displayallcolorscorrectlywhenrunning a 256 colorpalette.Toview the216 CrossPlatformColorsvisitthiswebpage:
http://profdevtrain.austincc.edu/html/216.html

16MillionDifferentColors

ThecombinationofRed,GreenandBlue valuesfrom0to255givesa totalofmorethan16 milliondifferentcolorstoplaywith(256x256x256).Mostmodernmonitorsarecapableofdisplaying atleast16,384differentcolors.Toassistyouinusingcolorschemes, checkout **http://wellstyled.com/tools/colorscheme2/index-en.html**.Thissiteletsyoutestdifferent color schemesfor page backgrounds, textand links

HTMLLists

HTMLprovidesasimplewaytoshowunorderedlists(bulletlists)ororderedlists(numberedlists).

UnorderedLists

An unordered listis a list of items marked with bullets (typically small black circles). An unordered list starts withthe<ul>tag. Eachlistitemstartswiththe<li>tag.

ThisCode	WouldDisplay
<li>Coffee</li> <li>Milk</li> </ul>	▪ Coffee ▪ Milk

OrderedLists

Anorderedlistisalsoalistofitems.Thelistitemsaremarkedwithnumbers.Anorderedliststartswiththe <ol>tag.Eachlistitemstartswiththe<li>tag.

ThisCode	WouldDisplay
<ol> <li>Coffee</li> <li>Milk</li>	1. Coffee 2. Milk

Insidealistitemyoucanput paragraphs,line breaks,images,links,other lists,

DefinitionLists

Definitionlistsconsistoftwoparts:a**term**anda**description**. Tomarkupadefinitionlist,you needthreeHTMLelements;acontainer<dl>,adefinitionterm<dt>,andadefinitiondescription <dd>.

ThisCode	WouldDisplay
<dl> <dt>CascadingStyleSheets</dt> <dd>Style sheets are used to provide presentationalsuggestionsfordocuments marked up in HTML. </dd>	CascadingStyleSheets Style sheets are used to provide presentational suggestions for documentsmarkedupinHTML.

Insidea definition-listdefinition(the<dd>tag)youcanputparagraphs,linebreaks,images,links,otherlists, etc

```
<html>
<head>
<title>MyFirstWebpage</title>
</head>
<bodybgcolor="#EDDD9E">
<h1align="center">MyFirstWebpage</h1>
<p>Welcome to my <strong>first</strong> webpage. I am writing this page using atext editor and plain old
html.</p>
<p>Bylearninghtml,I'llbeabletocreateweb pageslikeapro............. <br>whichIamof
course.</p>
```

```
        Here'swhatI'velearned:
        <ul>
    <li>HowtouseHTMLtags</li>
    <li>HowtouseHTMLcolors</li>
    <li>HowtocreateLists</li>

  </ul>
  </body>
  </html>
```

HTMLLinks

HTMLusesthe<a>anchortagtocreatealink toanotherdocument orwebpage.

TheAnchorTag andtheHrefAttribute

Ananchorcanpointtoanyresourceonthe Web:anHTMLpage,animage,asoundfile,amovie,etc.The syntax of creating ananchor:
<ahref="url">Texttobedisplayed</a>
The<a>tagisusedtocreateananchortolink from,thehrefattributeisusedtotelltheaddressofthe documentorpagewearelinkingto,andthewordsbetweentheopenandcloseoftheanchortagwillbe displayed as ahyperlink.

ThisCode	WouldDisplay
<ahref="http://www.austincc.edu/">VisitACC!</a>	VisitACC!

TheTargetAttribute

Withthetarget attribute, youcandefine **where**the linkeddocument willbeopened. Bydefault,

thelinkwillopeninthecurrentwindow.Thecodebelowwillopenthedocumentinanewbrowserwindow:

<ahref=http://www.austincc.edu/target="_blank">VisitACC!</a>

EmailLinks

createanemaillink,youwillusemailto:plusyouremailaddress. HereisalinktoACC'sHelpDesk:
<ahref="mailto:helpdesk@austincc.edu">EmailHelpDesk</a>
Toaddasubjectfortheemailmessage,youwouldadd?subject=aftertheemailaddress.Forexample:
<ahref="mailto:helpdesk@austincc.edu?subject=EmailAssistance">EmailHelpDesk</a>

TheAnchorTagandtheNameAttribute

Thenameattributeisusedtocreateanamedanchor.Whenusingnamedanchorswecancreatelinkst canjumpdirectlytoaspecificsection on apage,insteadoflettingtheuserscrollaroundtofindwhathe/sheislookingfor.Unlikeananchorthat useshref,anamedanchordoesn't change theappearanceofthetext (unlessyousetstylesforthatanchor)orindicateinanywaythatthereisanythingspecialaboutthetext

Belowisthesyntaxofanamedanchor:

```
<aname="top">Texttobedisplayed</a>
```

Tolinkdirectlytothetopsection,adda#signandthenameoftheanchortotheendofaURL, like this:

ThisCode	WouldDisplay
<a href="http://profdevtrain.austincc.edu/html /10links.html#top">Backtotopofpage</a>	Backtotopofpage
Ahyperlinktothetopofthepagefromwithinthefile 10links.htmlwilllooklikethis: <ahref="#top">Backtotopofpage</a>	Backtotopofpage

Note: Always add a trailingslash to subfolder references. If you link likethis: href="http://profdevtrain.austincc.edu/html", you will generate two HTTP requests tothe server,becausetheserverwill addaslashtotheaddressandcreatea newrequestlikethis: href="http://profdevtrain.austincc.edu/html/"

Namedanchorsareoftenusedtocreate"tableofcontents"atthebeginningofalargedocument.Each chapterwithinthedocument isgivenanamedanchor,and linksto eachoftheseanchorsareputatthetop ofthedocument.Ifabrowsercannotfindanamedanchorthathasbeenspecified,itgoestothetopofthe document.No erroroccurs.

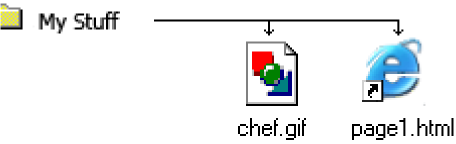
HTMLImages

TheImageTagand theSrcAttribute

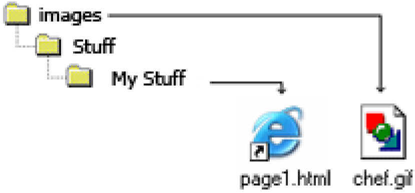
The<img>tagisempty,whichmeansthatitcontainsattributesonlyandithasnoclosingtag.Todisplay animageonapage,youneedtousethesrcattribute.Srcstandsfor"source".Thevalueofthesrcattribute istheURLoftheimageyouwanttodisplayonyourpage.Thesyntaxofdefininganimage:

ThisCode	WouldDisplay
<imgsrc="graphics/chef.gif">	

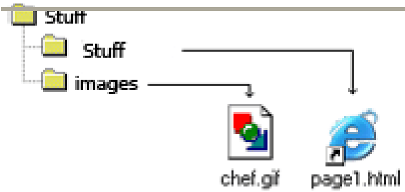
Notonlydoesthesourceattributespecifywhatimagetouse,butwheretheimageislocated.The aboveimage,graphics/chef.gif,meansthatthebrowserwilllookfortheimagename**chef.gif**ina **graphics** folder in the same folder as the html documentitself.



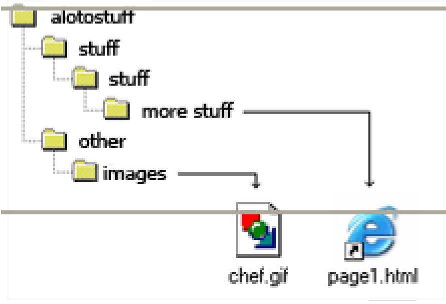
src="chef.gif"meansthattheimageisinthesame folder as the html document calling forit.



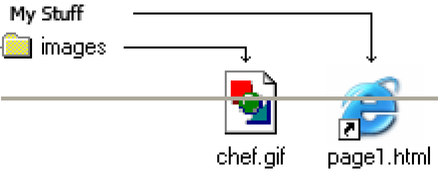
src="images/chef.gif"means that the image is one folderdownfromthehtmldocumentthatcalled for it. This can go on down as many layers as necessary.



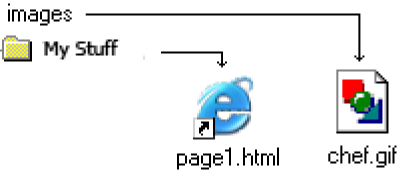
src="../chef.gif"means that the image is in one folderupfromthehtmldocumentthatcalledforit.



src="../../chef.gif"means that the image is two foldersupfromthehtmldocument thatcalledforit.



src="../images/chef.gif"means thattheimage is one folderupandthenanotherfolderdownintheimages directory.



src="../../../other/images/chef.gif"meansthisgoe multiple layers up.

Thebrowserputstheimagewheretheimagetagoccursinthedocument.Ifyouput animagetagbetween twoparagraphs,thebrowser showsthe firstparagraph, thenthe image, and thenthe secondparagraph.

TheAltAttribute

Thealt attributeisusedtodefineanalternatetextforanimage.Thevalueofthealt attributeis author-defined text:

```
<imgsrc="graphics/chef.gif"alt="SmilingHappy"Chef>
```

Thealtattributetellsthereaderwhatheorsheismissingonapageifthebrowsercan'tloadimages.The browserwillthendisplaythealternatetextinsteadoftheimage.Itisagoodpracticetoincludethealt attributeforeachimageonapage,toimprovethedisplayandusefulnessofyourdocumentforpeoplewh havetext-onlybrowsersor usescreen

readers.

eDimensions

n you have an image, the browser usually figuresout howbig the image is all byitself. If youput in magedimensions inpixels however, thebrowsersimplyreservesaspacefortheimage,thenloadsthe fthe page. Once the entire page is loads it can go back and fill in the images.Without dimensions, itrunsinto animage,thebrowserhasto pause loadingthepage,loadtheimage,thencontinue ng the page. The chef image would thenbe:

src="graphics/chef.gif"width="130"height="101"alt="SmilingHappyChef">

the file **mypage2.html** in your text editor and add code highlighted inbold:

```
<html>
<head>
<title>MyFirstWebpage</title>
</head>
<body>
<h1align="center">MyFirstWebpage</h1>
<p>Welcometomyfirstwebpage. Iamwritingthispageusingatexteditorand plainoldhtml.</p>
<p>Bylearninghtml,I'llbeabletocreatewebpages like a pro....<br>whichI am of
course.</p>
<!--Whowould haveguessedhow easythiswould be :)-->
<p><imgsrc="graphics/chef.gif"width="130"height="101"alt="SmilingHappyChef"align="ce
nter"></p>
<p align="center">This is my Chef</p>
</body>
</html>
```

Tables

Tablesaredefinedwiththe<table>tag.Atable isdividedinto rows(withthe<tr>tag),and eachrow is dividedintodatacells(withthe<td>tag).Theletterstdstandsfortabledata,whichisthe contentofadata cell.Adata cellcancontaintext, images,lists, paragraphs, forms, horizontalrules,tables, etc.

Code	WouldDisplay

```
<table>
<tr>
<td>row1,cell1</td>
<td>row1,cell2</td>
</tr>
<tr>
<td>row2,cell1</td>
 <td>row2,cell2</td>
 </tr>
```

row1,cell1 row1,cell2

row2,cell1 row2,cell2

TablesandtheBorderAttribute

Todisplayatablewithborders,youwillusetheborderattribute.

Code	WouldDisplay
<tableborder="1" > <tr> <td>Row1,cell1< /td> <td>Row1, cell2< /td> </tr> </table>	row1,cell1row 1,cell2

..

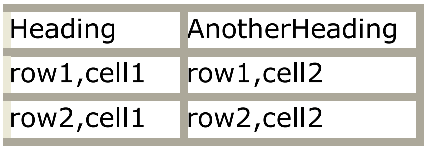
Code	WouldDisplay
<tableborder="5"> <tr> <td>Row1,cell1</td> <td>Row1,cell2</td> </tr>	row1,cell1row 1,cell2

Openupyourtexteditor.Typeinyour<html>,<head>and<body>tags.Fromhere onIwillonlybewriting whatgoesbetweenthe<body>tags.Typeinthefollowing:

```
<tableborder="1">
<tr>
<td>Tablescanbeusedtolayoutinformation</td>
<td>&nbsp;<imgsrc="http://profdevtrain.austincc.edu/html/graphics/chef.gif">&nbsp;
  </td>
  </tr>
  </table>
```

Headings inaTable

Headingsinatablearedefinedwiththe<th>tag.

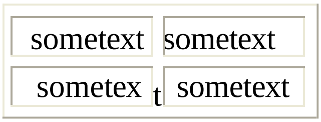
Thiscode	WouldDisplay
<tableborder="1"> <tr> <th>Heading</th> <th>AnotherHeading</th> </tr> <tr> <td>row1,cell1</td> <td>row1,cell2</td> </tr> <tr> <td>row2,cell1</td> <td>row2,cell2</td>	 Heading AnotherHeading row1,cell1 row1,cell2 row2,cell1 row2,cell2

CellPaddingandSpacing

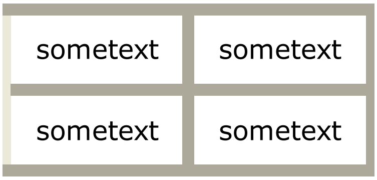
The<table>taghastwoattributesknownascellspacingandcellpadding. Hereisatableexamplewithout these properties. These properties may be used separately or together.

Code	WouldDisplay
<tableborder="1"> <tr> <td>sometext</td> <td>sometext</td> <tr> <td>sometext</td> <td>sometext</td> </tr>	sometextsometext sometextsometext

Cellspacingis thepixelwidth between theindividualdatacellsinthetable(Thethicknessofthelines makingthe tablegrid).Thedefaultiszero.Iftheborderissetat0,thecellspacinglineswillbeinvisible.

ThisCode	WouldDisplay
<tableborder="1"cellspacing="5"> <tr> <td>sometext</td> <td>sometext</td> </tr><tr> <td>sometext</td> <td>sometext</td> </tr>	 sometext sometext sometext sometext

Cellpaddingisthepixelspacebetweenthecellcontentsandthecellborder.Thedefaultfor thispropertyisalsozero.Thisfeatureisnotusedoften,butsometimescomesinhandywhenyouhaveyour bordersturnedonand youwantthecontentsto be awayfrom the borderabitforeasyviewing. Cellpadding is invisible, even with the border propertyturned on. Cellpadding can be handled in a style sheet.

ThisCode	WouldDisplay
<tableborder="1"cellpadding="10"> <tr> <td>sometext</td> <td>sometext</td> </tr><tr> <td>sometext</td> <td>sometext</td> </tr>	 sometext sometext sometext sometext

TableTags

Tag	Description
<table>	Definesatable
<th>	Definesatableheader
<tr>	Definesatablerow
<td>	Definesatablecell
<caption>	Definesatablecaption
<colgroup>	Definesgroupsoftablecolumns
<col>	Definestheattributevaluesforoneormorecolumns intaable

TableSize

TableWidth

Thewidthattributecanbeusedtodefinethewidthofyourtable.Itcanbedefinedasafixedwidthora relativewidth.Afixedtablewidthisonewherethewidthofthetableisspecifiedinpixels.For example, thiscode,<tablewidth="550">,willproduceatablethatis550pixelswide.Arelativetablewidthis specifiedasapercentageofthewidthofthevisitor'sviewingwindow.Hencethiscode,<table width="80%">,willproduceatablethatoccupies80percentofthe screen.

Thistablewidthis250pixels

Thistablewidthis50%

Therearearguments infavorofgiving yourtablesarelativewidthbecausesuchtablewidthsyieldpages that workregardless ofthe visitor's screenresolution. Forexample, a table widthof100%willalways span theentirewidthofthebrowserwindowwhetherthevisitorhasa800x600displayora1024x768display (etc).Yourvisitorneverneedstoscrollhorizontallytoreadyourpage,somethingthatisregardedbymost people as being veryannoying.

HTMLLayout-UsingTables

OneverycommonpracticewithHTML,isto useHTMLtablestoformatthelayoutofan HTMLpage.

A part of this page is formatted with two columns.Asyoucanseeonthispage,thereis a left column and a right column.

Thistextisdisplayed inthe leftcolumn.

AnHTML<table> isusedto divideapartof this Web page into two columns.

Thetrickistouseatablewithoutborders, and maybe a little extracell-padding.

No matter how much text you add to this page,it willstayinsideitscolumnborders.

```
  <html>
  <head>
  <title>MyFirstWebPage</title>
  </head>
 <body>
 <tablewidth="90%"cellpadding="5"cellspacing="0">
 <trbgcolor="#EDDD9E">
     <tdwidth="200"valign="top"><imgsrc="graphics/contact.gif"width="100"height="100"></td>
      <tdvalign="top"><h1align="right">JanetDoeson</h1>
      <h3align="right">TechnicalSpecialist</h3></td>
       </tr>
      <tr>
      <tdwidth="200">
          <h3>Menu</h3>
          <ul>
        <li><ahref="home.html">Home</a></li>
        <li><ahref="faq.html">FAQ</a></li>
        <li><ahref="contact.html">Contact</a></li>
        <li><ahref="http://www.austincc.edu">Links</a></li>
 </ul></td>
 <tdvalign="top"><h2align="center">Welcome!</h2>
 <p>Welcometomyfirst webpage. Icreatedthiswebpagewithout theassistanceofawebpageeditor.
Justmylittletext editorandakeenunderstanding ofhtml.</p>
       <p>Lookaround.NoticeI'mabletouseparagraphs,listsandheadings.Youmaynotbeabletotell,but layout is
done with a table. I'm very clever.</p>
 <blockquote>
     <p>Ialwayswantedtobesomebody, butnowIrealizeIshouldhavebeenmore specific.</p>
     <cite>LilyTomlin</cite></blockquote>                    </td>
     </tr>
 </table>
 <hrwidth="90%"align="left">
 <address>
       JanetDoeson<br>Technical
       Specialist<br>512.555.5555
  </address>
  <p>Contact meat<ahref="mailto:jdoeson@acme.com">jdoeson@acme.com</a></p>
      </body>
   </html>
```

CascadingStyleSheets(CSS)

CascadingStyleSheets, fondlyreferredto asCSS, isasimpledesignlanguage intended to simplify the process of making web pages presentable.

CSS handlesthe look andfeelpartofawebpage.Using CSS, you cancontrolthe color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, as well as a varietyof other effects. CSS is easyto learn and understand but it provides powerful control over the presentationofanHTMLdocument.Most commonly,CSS iscombinedwiththe markup languages HTML or XHTML.
AdvantagesofCSS:
nwriteCSSonceandthenreusesamesheetinmultipleHTML
pages. Youcandefineastyle for eachHTMLelement andapplyit toasmanyWeb pages as you want.
If youareusingCSS, youdo notneedtowriteHTMLtagattributeseverytime. Just writeone CSS rule ofa tag and applyto allthe occurrences ofthat tag. So less code means faster download times.

- Easymaintenance

Tomakeaglobalchange, simplychangethestyle, andallelements inalltheweb pages will be updated automatically.

- SuperiorstylestoHTML

CSS hasa muchwider arrayofattributesthanHTMLso youcangive far better lookto your HTML page in comparison of HTML

- attributes.MultipleDevice
  Compatibility

Style sheetsallow content to beoptimized for more thanone type ofdevice. Byusing the same HTMLdocument, different versionsofawebsitecanbepresentedforhandhelddevicessuchas PDAs and cell phones or for
printing.Globalweb
standards

NowHTMLattributesarebeingdeprecatedandit is beingrecommendedtouseCSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

Introductionto JavaScript

JavaScript is a programming language that can be included on web pages to make them more interactive. You can use it to check or modifythe contents of forms, change images, open new windows and write dynamic page content. You can even use it with CSS to make DHTML (Dynamic HyperText Markup Language). This allows you to make parts of your web pages appearordisappear ormovearoundonthepage. JavaScriptsonlyexecute onthepage(s)that are on your browser window at anyset time. Whenthe user stopsviewing that page, anyscripts that were running on it are immediatelystopped. The onlyexceptions are cookies or various client side storage APIs, which can be used by many pages to store and pass information between them,evenafterthepageshavebeenclosed.

Before we go anyfurther, let me say; JavaScript has nothing to do with Java. Ifwe are honest, JavaScript,originallynicknamed LiveWireandthenLiveScript whenit wascreatedbyNetscape, should in fact be called ECMAscript as it was renamed when Netscape passed it to the ECMAfor standardisation.

JavaScript isaclient side, interpreted,objectoriented,highlevelscripting language,while Java is a client side, compiled, object oriented high level language. Now after that mouthful, here's what it means.

Clientside

Programsarepassed to thecomputer thatthebrowser ison, andthat computer runsthem. The alternative is server side, where the program is run on the server and onlythe results are passed to the computer that the browser is on. Examples of this would be PHP, Perl, ASP, JSP etc.

Interpreted

Theprogramispassedassourcecodewithalltheprogramminglanguagevisible.Itis then converted into machine code as it is being used. Compiled languages are converted into machine code first then passed around, so you never get to see the original programming language. Java is actuallydual half compiled, meaning it is half compiled (to'bytecode') before it ispassed,thenexecutedinavirtualmachinewhichconvertsitto fullycompiled code just before use, in order to execute it on the computer's processor. Interpreted languages are generally less fussyabout syntax and if you have made mistakes inaparttheyneveruse, themistakeusuallywillnot cause youanyproblems.

Scripting

This is a little harder to define. Scripting languages are often used for performing repetitive tasks. Although they may be complete programming languages, they do not usually go into the depths of complex programs, such as thread and memory management. Theymayuse another programto dothe work and simplytell it what to do. Theyoften do not create their own user interfaces, and instead will relyon the other programs to create aninterface forthem. This isquite accurate for JavaScript. We do not have to tellthe browser exactlywhat to put on the screen for everypixel (thoughthere isa relativelynew API known as canvas that makes this possible if needed), we just tell it that we want it to change the document, and it does it. The browser willalso takecareof thememorymanagement andthreadmanagement,leavingJavaScript freetogetonwith the things it wants to do.

High level

Writteninwordsthat areasclosetoenglishaspossible. Thecontrastwouldbewith assemblycode, where each command can be directlytranslated into machine code.

Objectoriented

HowisJavaScriptconstructed

The basic part of a script is a variable, literalor object. A variable is a word that represents a pieceoftext, anumber, abooleantrueor false valueor anobject.Aliteralistheactualnumber or piece of text or boolean value that the variable represents. An object is a collection of variablesheldtogether byaparentvariable, oradocumentcomponent.

Thenext most importantpartofascript isanoperator.Operatorsassignliteralvaluesto variables or say what type of tests to perform.

Thenext most importantpartofascript isacontrolstructure. Controlstructuressaywhat scripts should be run if a test is satisfied.

Functionscollect controlstructures,actionsandassignmentstogether andcanbetoldtorun those pieces of script as and when necessary.

The most obvious parts of a script are the actions it performs. Some of these are done with operatorsbutmost aredoneusingmethods. Methodsareaspecialkindoffunctionand maydo things like submitting forms, writing pages or displaying messages.

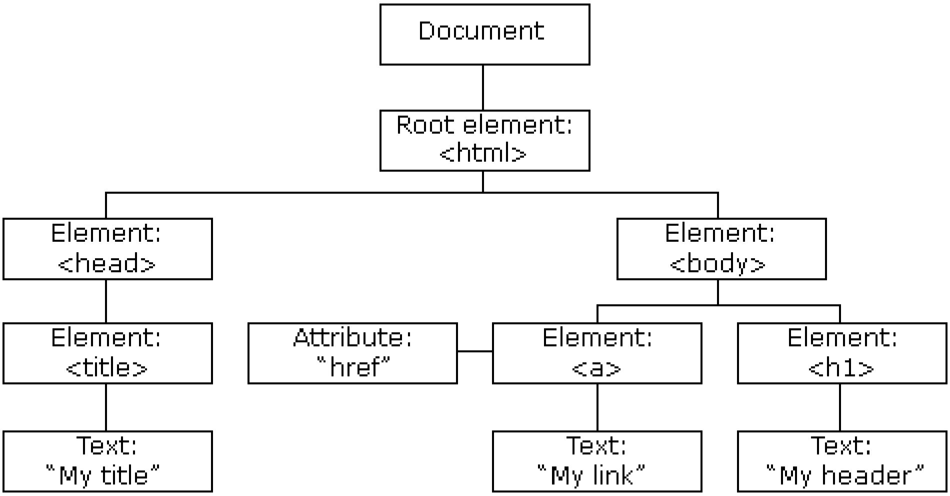
Eventscanbeusedtodetectactions, usuallycreatedbytheuser, suchas movingorclickingthe mouse, pressing a keyor resetting a form. Whentriggered, eventscan be used to runfunctions.

Lastlyandnot quitesoobviousisreferencing.This isaboutworkingoutwhatto writetoaccess the contents of objects or even the objects themselves.

As an example, think ofthe following situation. A person clicks a submit button on a form. Whentheyclickthebutton, wewanttocheck iftheyhave filledouttheir name inatext box and ifthey have, we want to submit the form. So, wetellthe formto detect the submit event. When the event is triggered, wetell it to run the function that holds together the tests and actions. The function contains a control structure that uses a comparison operator to test the text box to see that it is not empty. Ofcourse we have to work out how to reference the text box first. The text box is an object. One ofthe variables it holds is the text that is written in the text box. The text written in it is a literal. Ifthe text box is not empty, a method is used that submits the form.

TheHTMLDOM(DocumentObjectModel)
Whenawebpage is loaded,thebrowsercreatesa**D**ocument **O**bject **M**odelofthepage. The **HTML DOM** model is constructed as a tree of **Objects**:

TheHTMLDOMTreeofObjects



Withtheobject model,JavaScript getsallthepowerit needsto createdynamicHTML: JavaScript

can change all the HTML elements in the page
JavaScript can change all the HTML attributes in the page
JavaScript can change all the CSS styles in the pageJavaScript
canremoveexistingHTMLelementsand attributes JavaScript
can add new HTML elements and attributes JavaScript can
react to all existing HTML events in the page JavaScript can
create new HTML events in the page

RegularExpression

Regular expressions are patterns used to matchcharacter combinations in strings. In JavaScript, regular expressions are also objects. These patterns are used with the exec and test methods of RegExp,andwiththe match,replace,search,and split methodsofString.Thischapterdescribes JavaScript regular expressions.

Creatingaregularexpression

Youconstructaregularexpressioninoneoftwoways:

Usingaregularexpressionliteral,as follows:

varre= /ab+c/;

Regularexpressionliteralsprovidecompilationoftheregularexpressionwhenthescript is loaded. When the regular expression will remain constant, use this for better performance.

OrcallingtheconstructorfunctionoftheRegExp object,asfollows: var re

= new RegExp("ab+c");

Usingtheconstructorfunctionprovidesruntimecompilationoftheregular expression. Usethe constructor function when you know the regular expression pattern will be changing, or you don't know the pattern and are getting it from another source, such as user input.

Writingaregularexpression pattern

Aregular expressionpatterniscomposed ofsimplecharacters, suchas/abc/, oracombinationof simple and special characters, such as /ab*c/ or /Chapter (\d+)\.\d*/. The last example includes parentheses which are used as a memorydevice. The match made with this part of the pattern is remembered for later use, as described in Using parenthesized substring matches.

Usingsimplepatterns

Simple patterns are constructed of characters for which you want to find a direct match. For example, the pattern /abc/ matches character combinations in strings only when exactly the characters 'abc' occurtogether and in that order. Such a match would succeed in the strings "Hi, do youknow yourabc's?"and"The latest airplane designsevolved fromslabcraft."Inbothcases the match is with the substring 'abc'. There is no match in the string 'Grab crab' because while it contains the substring 'ab c', it does not contain the exact substring 'abc'.

Usingspecialcharacters

Whenthesearchfor amatchrequiressomething morethanadirect match, suchas findingoneor more b's, or finding white space,the patternincludesspecialcharacters. For example, the pattern /ab*c/ matches anycharacter combination in whichasingle 'a' is followed byzero or more'b's(* means 0 or more occurrences ofthe preceding item) and then immediately followed by 'c'. In the string "cbbabbbbcdebc," the pattern matches the substring 'abbbbc'.

The following tableprovidesacomplete list and descriptionofthe special charactersthat canbe used in regular expressions.

Specialcharactersinregularexpressions.

Character	**Meaning**
\	Matchesaccordingtothefollowingrules: Abackslashthat precedesanon-specialcharacter indicatesthatthenext character is specialand isnottobe interpretedliterally. For example,a'b'withoutapreceding '\' generally matches lowercase 'b's wherever theyoccur. But a '\b' by itself doesn't match any character; it forms the special word boundary character. A backslash that precedes a special character indicates that the next character is not specialand should be interpretedliterally. For example, the pattern/a*/ reliesonthe specialcharacter '*'to match0or morea's. Bycontrast,thepattern/a*/removesthe specialness of the '*' to enable matches with strings like 'a*'. Do notforgetto escape \itselfwhileusingtheRegExp("pattern")notationbecause \ is also an escape character in strings.
^	Matchesbeginningofinput.Ifthe multiline flagissettotrue,also matches immediately after a line break character. Forexample,/^A/doesnotmatchthe'A'in"anA",butdoesmatchthe'A'in "An E". The'^'hasadifferent meaningwhenit appearsasthe first characterinacharacterset pattern. See complemented character sets for details and an example.
$	Matchesendofinput.Ifthe multiline flagissettotrue,also matchesimmediately before a line break character. For example, /t$/doesnotmatchthe't'in"eater",butdoesmatchitin"eat".
*	Matchesthepreceding character0ormoretimes.Equivalentto{0,}.

Specialcharactersinregularexpressions.

Character	Meaning
	Forexample,/bo*/matches'boooo'in"Aghost booooed"and'b'in"A bird warbled", but nothing in "A goat grunted".
+	Matchesthepreceding character1or moretimes. Equivalentto{1,}. Forexample,/a+/matchesthe'a'in"candy"and allthea'sin "caaaaaaandy", but nothing in "cndy".
?	Matchesthepreceding character0or1time.Equivalent to{0,1}. Forexample,/e?le?/ matchesthe'el'in"angel"andthe'le'in"angle"andalso the 'l' in "oslo". If used immediately after any of the quantifiers *, +, ?, or {}, makes the quantifier non-greedy (matching the fewest possible characters), as opposed tothedefault, whichisgreedy(matchingas manycharactersaspossible). For example, applying /\d+/to"123abc"matches"123".Butapplying/\d+?/tothatsamestring matches only the "1". Alsousedinlookaheadassertions,asdescribedinthex(?=y)andx(?!y) entries of this table.
.	(Thedecimalpoint) matchesanysinglecharacterexceptthenewline character. Forexample,/.n/matches'an'and'on'in"nay,anappleisonthetree",butnot
(x)	Matches'x'andremembersthematch,asthe followingexampleshows. The parentheses are called *capturing parentheses*. The '(foo)'and '(bar)' inthe pattern/(foo)(bar) \1 \2/ matchand remember the first two words in the string "foo bar foo bar". The \1 and \2 in the pattern matchthe string's last two words. Notethat \1, \2, \nare used inthe matching partoftheregex. Inthereplacement partofaregexthesyntax $1, $2, $nmust be used, e.g.: 'bar foo'.replace(/(...) (...)/, '$2 $1').
(?:x)	Matches 'x' but doesnot remember the match. The parenthesesare called *non-capturingparentheses*,andlet youdefinesubexpressionsforregular expression operators to work with. Consider the sample expression /(?:foo){1,2}/.Iftheexpressionwas/foo{1,2}/,the{1,2}characterswould apply only to the last 'o' in 'foo'.Withthenon-capturingparentheses, the{1,2}appliestotheentireword

Specialcharactersinregularexpressions.

Character	Meaning
x(?=y)	Matches'x'onlyif'x' isfollowedby'y'.Thisiscalled alookahead. Forexample,/Jack(?=Sprat)/matches'Jack'onlyifitis followedby'Sprat'. /Jack(?=Sprat|Frost)/ matches'Jack'onlyif it is followed by'Sprat'or 'Frost'.However,neither 'Sprat'nor'Frost'ispartofthematchresults.
x(?!y)	Matches'x'onlyif'x'isnotfollowed by'y'.Thisiscalledanegated lookahead. Forexample,/\d+(?!\.)/ matchesanumberonlyifit isnot followed byadecimal point. The regular expression /\d+(?!\.)/.exec("3.141") matches '141' but not
x|y	Matcheseither'x'or'y'. Forexample, /green|red/matches 'green'in"greenapple"and'red'in"red apple."
{n}	Matchesexactlynoccurrencesoftheprecedingcharacter. Nmust bea positive integer. For example, /a{2}/doesn't matchthe 'a'in"candy,"but it doesmatchallofthea's in "caandy,"andthefirsttwoa'sin"caaandy."
{n,m}	Wherenand marepositive integersandn<= m. Matchesat least nandat most moccurrences ofthe preceding character. When m is omitted, it's treated as ∞. For example, /a{1,3}/ matches nothing in "cndy", the 'a' in "candy," the first two a'sin"caandy,"andthefirstthreea'sin"caaaaaaandy".Noticethatwhenmatching "caaaaaaandy", the match is "aaa", even thoughthe original string had more a's in
[xyz]	Character set. This pattern type matches anyone of the characters in the brackets, including escape sequences. Special characters like the dot(.) and asterisk (*) are notspecialinsideacharacterset, sotheydon't needto beescaped.Youcanspecify a range of characters byusing a hyphen, as the following examples illustrate. Thepattern[a-d],whichperformsthesame matchas[abcd], matchesthe'b'in "brisket" and the 'c' in "city". The patterns /[a-z.]+/ and /[\w.]+/ match the entire string "test.i.ng".
[^xyz]	Anegatedorcomplementedcharacterset.Thatis,it matchesanythingthat isnot enclosed in the brackets. You can specifya range of characters byusing a hyphen. Everything that works in the normal character set also works here. For example, [^abc] isthesameas[^a-c].Theyinitiallymatch'r'in"brisket"and 'h' in "chop."

Specialcharactersinregularexpressions.

Character	**Meaning**
[\b]	Matchesabackspace(U+0008).Youneedtousesquarebracketsifyouwant to match a literal backspace character. (Not to be confused with \b.)
\b	Matches a word boundary. A word boundary matches the position where a word character is not followed or preceeded byanother word-character. Note that a matchedwordboundaryis not included inthe match. Inother words,thelengthof a matched word boundary is zero. (Not to be confused with [\b].) Examples: /\bm/matchesthe'm'in"moon"; /oo\b/doesnotmatchthe'oo'in"moon",because 'oo'isfollowedby'n'whichis a word character; /oon\b/ matchesthe'oon'in"moon", because 'oon'istheendofthestring,thus not followed by a word character; /\w\b\w/willnever matchanything,becauseawordcharactercanneverbe followed by both a non-word and a word character. **Note:** JavaScript's regular expression engine defines a specific set of characters to be"word"characters. Anycharacternotinthat setisconsideredawordbreak.This set of characters is fairlylimited: it consists solelyof the Roman alphabet in both upper- and lower-case, decimal digits, and the underscore character. Accented characters, such as "é" or "ü" are, unfortunately, treated as word breaks.
\B	Matchesanon-wordboundary.This matchesapositionwherethepreviousand next characterareofthesametype:Eitherbothmust bewords,orbothmust be non-words. The beginning and end of a string are considered non-words. Forexample,/\B../matches'oo'in"noonday", and/y\B./matches'ye'in "possibly yesterday."
\c*X*	Where*X*isacharacterrangingfromAtoZ.Matchesacontrolcharacterin astring. Forexample, /\cM/matchescontrol-M(U+000D) ina string.
\d	Matchesadigitcharacter.Equivalentto[0-9]. Forexample,/\d/or/[0-9]/ matches'2'in"B2isthesuitenumber."
\D	Matchesanynon-digitcharacter.Equivalentto[^0-9]. Forexample,/\D/or/[^0-9]/matches'B'in"B2is thesuite number."
\f	Matchesaformfeed (U+000C).

Specialcharactersinregularexpressions.

Character	**Meaning**
\n	Matchesa linefeed(U+000A).
\r	Matchesacarriage return(U+000D).
\s	Matchesasinglewhitespacecharacter,includingspace,tab, formfeed, line feed. Equivalent to [\f\n\r\t\v\u00a0\u1680\u180e\u2000-\u200a \u2028\u2029\u202f\u205f\u3000]. Forexample, /\s\w*/matches'bar'in"foobar."
\S	Matchesasinglecharacterotherthanwhitespace.Equivalentto[^ \f\n\r\t\v \u00a0\u1680\u180e\u2000-\u200a\u2028\u2029\u202f\u205f\u3000]. For example, /\S\w*/matches'foo'in"foobar."
\t	Matchesa tab(U+0009).
\v	Matchesaverticaltab(U+000B).
\w	Matchesanyalphanumericcharacterincludingtheunderscore.Equivalentto [A- Za-z0-9_]. For example,/\w/matches'a'in"apple,"'5'in"$5.28,"and'3'in"3D."
\W	Matchesanynon-wordcharacter.Equivalentto[^A-Za-z0-9_]. Forexample,/\W/or/[^A-Za-z0-9_]/matches'%'in"50%."
n	Where*n*isapositiveinteger,abackreferencetothelastsubstringmatchingthe*n* parentheticalintheregularexpression(countingleftparentheses). Forexample,/apple(,)\sorange\1/matches'apple,orange,'in"apple,orange, cherry, peach."
\0	MatchesaNULL(U+0000)character. Do notfollowthiswithanotherdigit, because \0<digits>isanoctalescapesequence.
\xhh	Matchesthecharacterwiththecodehh(two hexadecimaldigits)
\uhhhh	Matchesthecharacterwiththecodehhhh(fourhexadecimaldigits).

Escapinguser inputto betreatedasaliteralstringwithinaregularexpressioncanbe accomplished by simple replacement:

```
functionescapeRegExp(string){
  returnstring.replace(/[.*+?^${}()|[\]\\]/g,"\\$&");
}
```

Using parentheses

Parentheses around any part of the regular expression pattern cause that part of the matched substringto beremembered.Onceremembered,thesubstringcanberecalled forotheruse,as described in Using Parenthesized Substring Matches.

For example, the pattern /Chapter(\d+)\.\d*/illustratesadditionalescaped and specialcharacters and indicates that part of the pattern should be remembered. It matches preciselythe characters 'Chapter ' followed byone or more numeric characters (\d means any numeric character and + means 1 or moretimes), followed bya decimal point (which in itself is a special character; preceding the decimalpoint with \ means the pattern must look for the literalcharacter '.'), followed byanynumeric character 0 or more times (\d means numeric character, * means 0 or moretimes).Inaddition, parenthesesareusedtorememberthe first matchednumericcharacters.

Thispatternis found in"OpenChapter4.3,paragraph6"and '4'isremembered. Thepatternis not found in "Chapter 3 and 4", because that string does not have a period after the '3'.

Tomatchasubstringwithoutcausingthe matched parttoberemembered, withintheparentheses preface the pattern with ?:. For example, (?:\d+) matches one or more numeric characters but does not remember the matched characters.

Workingwithregularexpressions

Regular expressions are used with the RegExp methods test and exec and with the String methodsmatch,replace,search,andsplit.These methodsareexplained indetailintheJavaScript reference.

Methodsthatuseregular expressions	
Method	**Description**
exec	ARegExp methodthat executesasearchfor amatchinastring. It returnsanarrayof information.
test	ARegExpmethodthattestsfora match ina string.Itreturnstrueor false.
match	AStringmethodthatexecutesasearchforamatch inastring.Itreturnsanarrayof

Methodsthatuseregular expressions	
Method	**Description**
	informationornullonamismatch.
search	AString methodthattestsforamatchinastring. Itreturnsthe indexofthe match, or-1 if the search fails.
replace	AString methodthat executesasearchforamatchinastring,andreplacesthe matched substring with a replacement substring.
split	AString methodthat usesaregularexpressionorafixedstringto breakastring into an array of substrings.

Whenyouwantto knowwhether apatternis foundinastring, usethetestorsearchmethod;for more information (but slower execution) use the exec or match methods. If you use exec or match and if the match succeeds, these methods return an arrayand update properties of the associated regular expression object and also of the predefined regular expression object, RegExp. If the match fails, the exec method returns null (which coerces to false).

Inthe followingexample,thescript usestheexec methodto finda matchinastring. var

```
myRe = /d(b+)d/g;
varmyArray=myRe.exec("cdbbdbsbz");
```

If youdonot needto accessthepropertiesoftheregular expression, analternativewayof creating myArray is with this script:

```
varmyArray=/d(b+)d/g.exec("cdbbdbsbz");
```

Ifyouwantto constructtheregularexpressionfromastring, yet anotheralternative isthisscript: var

```
myRe = new RegExp("d(b+)d", "g");
varmyArray=myRe.exec("cdbbdbsbz");
```

Withthese scripts,the matchsucceedsand returnsthearrayand updatesthepropertiesshown in the following table.

Property orindex	Description	In this example
	The matchedstringandallremembered substrings.	["dbbd", "bb"]
index	The0-basedindexofthematchintheinputstring.	1
input	Theoriginalstring.	"cdbbdbsbz
[0]	Thelastmatchedcharacters.	"dbbd"
lastIndex	The index at which to start the next match. (This propertyis set only if the regularexpressionusesthegoption,describedin AdvancedSearchingWith Flags.)	5
source	Thetextofthepattern. Updatedatthetimethattheregularexpressionis created, not executed.	

As shown in the second formof this example, you can use a regular expression created with an object initializerwithoutassigning itto avariable.Ifyoudo,however,everyoccurrence isanew regular expression. Forthis reason, if you use this formwithout assigning it to a variable, you cannot subsequentlyaccess the properties of that regular expression. For example, assume you have this script:

```
varmyRe=/d(b+)d/g;
varmyArray=myRe.exec("cdbbdbsbz");
console.log("ThevalueoflastIndexis"+myRe.lastIndex);

// "The value of lastIndex is 5"
```

However,ifyouhavethisscript:

```
var myArray =
/d(b+)d/g.exec("cdbbdbsbz");console.log("ThevalueoflastInde
xis"+/d(b+)d/g.lastIndex);

//"ThevalueoflastIndexis0"
```

The occurrences of/d(b+)d/g in the two statements are different regular expression objects and hence havedifferent valuesfortheir lastIndexproperty. Ifyouneedto accessthepropertiesofa regular expression created with an object initializer, you should first assign it to a variable.

Usingparenthesizedsubstringmatches

Including parentheses in a regular expression pattern causes the corresponding submatch to be remembered.Forexample,/a(b)c/ matchesthecharacters'abc'andremembers'b'.Torecallthese parenthesized substring matches, use the Array elements [1], ..., [n].

Thenumber ofpossibleparenthesizedsubstrings is unlimited. Thereturnedarrayholdsallthat were found. The following examples illustrate howto use parenthesized substring matches.

The following script uses the replace() method to switch the words in the string. For the replacementtext,thescript usesthe$1and$2inthereplacement todenotethefirst andsecond parenthesized substring matches.

```
varre=/(\w+)\s(\w+)/;
varstr= "JohnSmith";
varnewstr=str.replace(re,"$2,$1");
console.log(newstr);
```

Thisprints"Smith,John".

Advancedsearchingwithflags

Regularexpressionshave fouroptionalflagsthat allow forglobalandcase insensitivesearching. These flags can be used separatelyor together in anyorder, and are included as part of the regular expression.

Regularexpression flags	
Flag	**Description**
g	Globalsearch.
i	Case-insensitivesearch.
m	Multi-linesearch.
y	Performa"sticky"searchthatmatchesstartingatthecurrentpositioninthetargetstring.

Toincludeaflagwiththeregularexpression, usethissyntax:

```
varre=/pattern/flags; or

varre=newRegExp("pattern","flags");
```

Notethattheflagsareanintegralpartofaregular expression. Theycannot beaddedorremoved later.

For example, re=/\w+\s/gcreatesaregular expressionthat looksforoneor morecharacters followed by a space, and it looks for this combination throughout the string.

```
var re=/\w+\s/g;
```

```
varstr="feefifofum";
varmyArray=str.match(re);
console.log(myArray);
```

Thisdisplays["fee","fi","fo "].Inthisexample, youcould replacethe line: var re

= /\w+\s/g;

with:

varre=newRegExp("\\w+\\s","g");andgetthesameresult.

The mflag is used to specifythat a multiline input string should be treated as multiple lines. If themflag isused, ^ and$ matchatthestartorend ofanylinewithinthe input string insteadof the start or end of the entire string.

Date Object

BoththeDate(string) constructorand parse() methodworkonexactlythethesamedateformats. The difference is that the constructor creates a Dateobject, while the static Date.parse() method returns a number - more precisely, the number of milliseconds since Jan 1, 1970:?

```
var d1 = new Date("March 1, 2013");
console.log(d1);        //FriMar100:00:00EST2013
console.log(typeof d1); //object

var d2 = Date.parse("March 1, 2013");
console.log(d2);        //1332302400000
console.log(typeof d2); //number
```

Either ofthe above will also work for numeric date formats, assuming that you're dealing with a supported format, such as yyyy/MM/dd, yyyy/M/d, yyyy/MM/dd hh:mm, or yyyy/mm/dd hh:mm:ss. Aside fromthat shortlist, mostother dateformats - withthenotableexceptionoflong dateformatslikeMon, January1,2000,whichmakeexcellent candidates forstringparsing-will result inunpredictable results at best. Oddly, accordingto Wikipedia, the standard Calendar date representation allows both the YYYY-MM-DD and YYYYMMDD formats, as well as the year-month-only YYYY-MM format.

Errors&ExceptionsHandling

Therearethreetypesoferrorsinprogramming:(a) SyntaxErrors,(b) RuntimeErrors, and (c) Logical Errors.

SyntaxErrors

Syntaxerrors,also called **parsingerrors,**occuratcompiletime intraditionalprogramming languages and at interpret time in JavaScript.

Forexample,thefollowinglinecausesasyntaxerrorbecause itismissingaclosingparenthesis.

```
 <scripttype="text/javascript">
   <!--
      window.print(;
   //-->
</script>
```

When a syntax erroroccursinJavaScript,onlythe code contained within the same thread asthe syntax error is affected and the rest ofthe code in other threads getsexecuted assuming nothing in them depends on the code containing the error.

RuntimeErrors

Runtimeerrors,alsocalled**exceptions,**occurduringexecution(aftercompilation/interpretation).

Forexample,thefollowing linecausesaruntimeerrorbecause herethesyntaxiscorrect,but at runtime, it is trying to call a method that does not exist.

```
 <scripttype="text/javascript">
   <!--
      window.printme();
   //-->
</script>
```

Exceptionsalso affectthethreadinwhichtheyoccur,allowingother JavaScript threadsto continue normal execution.

LogicalErrors

Logic errors can bethe most difficult type oferrors to track down. These errors arenot the result ofasyntaxor runtimeerror. Instead,theyoccur when you makea mistake inthe logicthat drives your script and you do not get the result you expected.

Youcannot catchthoseerrors,because it dependsonyourbusinessrequirement what typeof logic you want to put in your program.

Thetry...catch...finallyStatement

Thelatest versionsofJavaScript addedexceptionhandlingcapabilities.JavaScript implements the **try...catch...finally** construct as well as the **throw** operator to handle exceptions.

Youcan**catch**programmer-generatedand**runtime**exceptions,but youcannot **catch**JavaScript syntax errors.

Hereisthe**try...catch...finally**blocksyntax−

```
<scripttype="text/javascript">
  <!--
    try{
      //Codetorun
      [break;]
    }

    catch(e){
      //Codetorunifanexceptionoccurs [break;]
    }

    [ finally{
      //Codethatisalways executedregardless of
      // anexceptionoccurring
    }]
  //-->
</script>
```

The**try**block must be followed byeither exactlyone**catch** blockorone**finally** block (oroneof both). When an exception occurs in the **try** block, the exception is placed in **e** and the **catch** block is executed. The optional **finally** block executes unconditionally after try/catch.

Examples

Hereisanexamplewherewearetryingtocallanon-existingfunctionwhichinturnisraisingan exception. Let us see how it behaves without **try...catch**−

```
<html>
  <head>

    <scripttype="text/javascript">
      <!--
        functionmyFunc()
        {
          vara= 100;
          alert("Valueofvariableais:"+a);
        }
      //-->
    </script>
  </head>

  <body>
    <p>Clickthefollowingtoseetheresult:</p>

    <form>
      <inputtype="button"value="ClickMe"onclick="myFunc();" />
    </form>

  </body>
</html>
```

EventHandler

An event handler executes a segment of a code based on certain events occurring within the application, such as onLoad, onClick. JavaScript event handers can be divided into two parts: interactiveevent handlersand non-interactiveevent handlers. An interactiveevent handler isthe one that depends on the user interactivitywith the form or the document. For example, onMouseOver is an interactive event handler because it depends on the users action with the mouse. On the other hand non-interactive event handler would be onLoad, because this event handlerwouldautomaticallyexecuteJavaScript codewithouttheuser'sinteractivity.Hereareall the event handlers available in JavaScript:

EventHandler	UsedIn
onAbort	image
onBlur	select,text,textarea
onChange	select, text, textarea
onClick	button,checkbox,radio,link,reset,submit,area image
onError	select,text,testarea
onFocus	windows, image
onLoad	link, area
onMouseOver	link,area
onMouseOut	text,textarea
onSelect	formwindow
onSubmit	
onUnload	

---

onAbort:

AnonAborteventhandler executesJavaScriptcodewhentheuser abortsloadinganimage.

```
<HTML>
<TITLE>ExampleofonAbortEventHandler</TITLE>
<HEAD>
</HEAD>
```

```
<BODY>
<H3>ExampleofonAbortEventHandler</H3>
<b>Stoptheloading ofthisimageandseewhathappens:</b><p>
<IMGSRC="reaz.gif"onAbort="alert('Youstoppedtheloadingtheimage!')">
</BODY>
</HTML>
```

Here, analert()methodiscalledusingtheonAbortevent handlerwhentheuserabortsloading the image.

onBlur:

AnonBlur event handler executesJavaScript codewheninput focus leavesthe fieldofatext, textarea, or a select option. For windows, frames and framesets the event handler executes JavaScript codewhenthewindowlosesfocus. Inwindows youneedtospecifytheevent handler in the <BODY> attribute. For example:

```
<BODYBGCOLOR='#ffffff'onBlur="document.bgcolor='#000000'">
```

Note: OnaWindowsplatform, theonBlur event doesnotworkwith<FRAMESET>. See

Example:

```
<HTML>
<TITLE>ExampleofonBlurEventHandler</TITLE>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function valid(form){ var input=0;
    input=document.myform.data.value;
               if(input<0){
                   alert("Please inputavaluethatislessthan0");
               }
}
</SCRIPT>
</HEAD>
<BODY>
<H3>ExampleofonBlurEventHandler</H3> Try
inputting a value less than zero:<br>
<formname="myform">
 <inputtype="text"name="data"value=""size=10onBlur="valid(this.form)">
</form>
</BODY>
</HTML>
```

Inthis example, 'data' is atext field. Whena user attempts to leave the field, the onBlur event handler callsthe valid() functiontoconfirmthat 'data'hasa legalvalue. Notethatthekeyword *this* is used to refer to the current object.

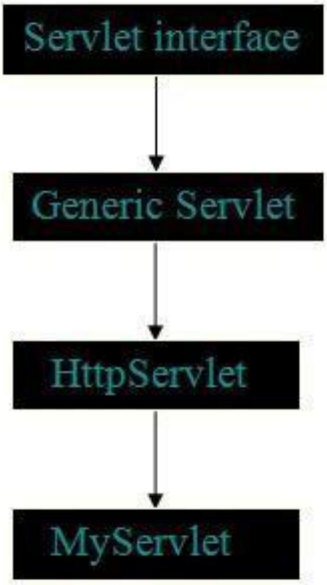
onChange:

TheonChangeevent handlerexecutesJavaScript codewheninput focusexitsthe fieldafterthe user modifies its text.

See Example:

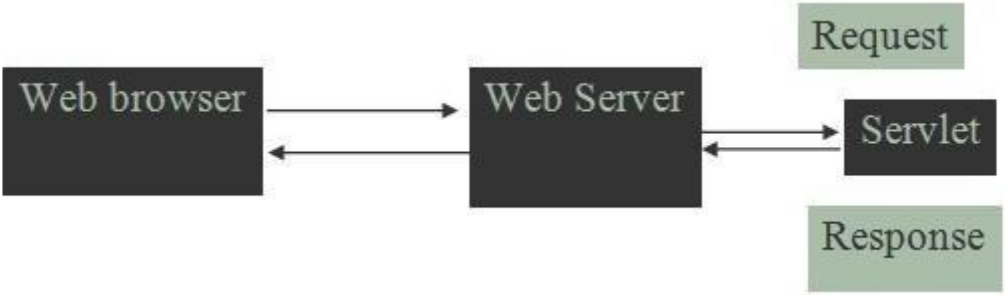
```
<HTML>
<TITLE>ExampleofonChangeEventHandler</TITLE>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function valid(form){ var input=0;
    input=document.myform.data.value;
    alert("Youhavechanged the valuefrom10to "+input);
}
</SCRIPT>
</HEAD>
<BODY>
<H3>ExampleofonChangeEventHandler</H3>
Trychanging thevaluefrom10to something else:<br>
<formname="myform">
 <inputtype="text"name="data"value="10"size=10onChange="valid(this.form)">
</form>
</BODY>
</HTML>
```

Servlet is a class, which implements the javax.servlet.Servlet interface. However instead of directlyimplementingthejavax.servlet.Servlet interfaceweextendaclassthat hasimplemented the interface like javax.servlet.GenericServlet or javax.servlet.http.HttpServlet.



ServletExceution

This ishowaservlet executiontakesplacewhenclient (browser)makesarequesttothe webserver.



Servletarchitectureincludes:

a) **ServletInterface**
Towriteaservlet weneedtoimplement Servlet interface.Servlet interfacecanbeimplemented directly or indirectly by extending **GenericServlet** or **HttpServlet** class.

b) **Requesthandling methods**
Thereare3methodsdefinedinServletinterface:**init(), service() and destroy()**.

Thefirst timeaservlet isinvoked,theinit methodiscalled. It iscalledonlyonceduringthe lifetime of a servlet. So, we can put all your initialization code here.

TheService methodisusedfor handlingtheclientrequest. Astheclient requestreachestothe container it creates a thread ofthe servlet object, and request and response object are also created.Theserequest andresponseobject arethenpassedasparametertotheservice method, which then process the client request. The service method in turn calls the doGet or doPost methods (if the user has extended the class from HttpServlet).

c) **Number of instances**

BasicStructureofaServlet

```
publicclassfirstServletextendsHttpServlet{
  publicvoidinit() {
    /*Putyourinitializationcodeinthismethod,
     *asthismethodiscalled onlyonce*/
  }
  publicvoidservice() {
    //Servicerequestfor Servlet
  }
  publicvoiddestroy(){
    //Fortaking theservlet outofservice,thismethodiscalled onlyonce
  }
}
```

Aservlet lifecyclecanbedefined astheentireprocess fromitscreationtillthedestruction. The following are the paths followed by a servlet

Theservletis initializedbycallingthe**init ()**method.
Theservlet calls**service()** methodto processaclient'srequest. The
servlet is terminated by calling the **destroy()** method.
Finally,servlet isgarbagecollectedbythegarbage collectoroftheJVM. Now

let us discuss the life cycle methods in details.

Theinit() method:

The init method isdesigned to be called onlyonce. It is called whenthe servlet is first created, andnot calledagainforeachuserrequest.So,it isused forone-time initializations, just aswith the init method of applets.

Theservlet isnormallycreatedwhenauser first invokesaURLcorrespondingtotheservlet, but you can also specifythat the servlet be loaded when the server is first started.

When a user invokes a servlet, a single instance of each servlet gets created, with each user requestresulting inanewthreadthatishandedoffto doGetordoPost asappropriate. Theinit() method simplycreates or loads some data that will be used throughout the life of the servlet.

The init method definition looks like this:

```
publicvoidinit()throwsServletException{
  //Initializationcode...
}
```

Theservice() method:

The service() method is the main method to performthe actualtask. The servlet container (i.e. webserver) callstheservice() methodtohandlerequestscoming fromtheclient(browsers) and to write the formatted response back to the client.

Each time the server receives a request for a servlet, the server spawns a new thread and calls service.Theservice()methodchecks theHTTPrequesttype(GET,POST, PUT,DELETE, etc.) and calls doGet, doPost, doPut, doDelete, etc. methods as appropriate.

Here isthe signatureofthismethod:

```
publicvoidservice(ServletRequestrequest,
                  ServletResponse response)
     throwsServletException,IOException{
}
```

The service () method is called bythe container and service method invokes doGe, doPost, doPut, doDelete, etc. methods as appropriate. So youhave nothing to do withservice() method but youoverrideeitherdoGet()ordoPost()dependingonwhattypeofrequest youreceive from the client.

ThedoGet()anddoPost()aremost frequentlyused methodswithineachservicerequest.Hereis the signature of these two methods.

ThedoGet()Method

AGET request resultsfromanormalrequest for aURLor fromanHTMLformthat hasno METHOD specified and it should be handled by doGet() method.

```
publicvoiddoGet(HttpServletRequestrequest,
                HttpServletResponse response)
   throwsServletException,IOException{
```

```
   //Servletcode
}
```

ThedoPost()Method

APOST requestresults fromanHTMLformthat specificallylistsPOST astheMETHODandit should be handled by doPost() method.

```
publicvoiddoPost(HttpServletRequestrequest,
              HttpServletResponse response)
   throwsServletException,IOException{
   //Servletcode
}
```

Thedestroy()method :

The destroy() method is called onlyonce at the end ofthe life cycle ofa servlet. This method gives yourservletachancetoclosedatabaseconnections,halt backgroundthreads,writecookie lists or hit counts to disk, and perform other such cleanup activities.

Afterthedestroy()methodiscalled,theservlet objectismarkedforgarbagecollection.The destroy method definition looks like this:

```
   publicvoiddestroy(){
   //Finalizationcode...
  }
```

ArchitectureDigram:

Thefollowing figuredepictsatypicalservletlife-cyclescenario.

FirsttheHTTPrequestscomingtotheserveraredelegatedtotheservlet container. The servlet container loads the servlet before invoking the service() method. Thentheservlet containerhandlesmultiplerequestsbyspawning multiplethreads,each thread executing the service() method of a single instance of the servlet.

session

Asessionisaconversationbetweentheserver and aclient. Aconversationconsistsseriesof continuous request and response.

Whyshouldasession bemaintained?

Whenthere isaseriesofcontinuousrequest andresponse fromasameclient toaserver,the server cannot identify fromwhich client it is getting requests. Because HTTP is a stateless protocol.

When there is a need to maintain the conversational state, session tracking is needed. For example, in a shopping cart application a client keeps on adding items into his cart using multiplerequests. Wheneveryrequest is made, theserver should identifyinwhichclient'scart the item is to be added. So in this scenario, there is a certain need for session tracking.

Solutionis,whenaclient makesarequest it should introduceitselfbyprovidingunique identifier everytime. There are five different methods to achieve this.

Sessiontrackingmethods:

1. Userauthorization
2. Hiddenfields
3. URLrewriting
4. Cookies
5. SessiontrackingAPI

Thefirstfourmethodsaretraditionallyusedforsessiontrackinginalltheserver-side technologies. ThesessiontrackingAPI methodisprovidedbytheunderlyingtechnology(java servlet or PHP or likewise). Session tracking API is built on topof the first four methods.

1. UserAuthorization

Users can be authorized to use the web application in different ways. Basic concept is that the user willprovideusernameandpasswordtologinto theapplication. Basedonthattheuser can be identified and the session can be maintained.

2. HiddenFields

<INPUTTYPE=‖hidden‖NAME=‖technology‖VALUE=‖servlet‖>
Hiddenfieldsliketheabovecanbeinsertedinthewebpagesandinformationcanbesenttothe

server for sessiontracking. These fields are not visible directlyto the user, but can be viewed using view sourceoption fromthe browsers. This typedoesn't need anyspecial configuration fromthe browser ofserver and bydefault availableto use for sessiontracking. This cannot be used for session tracking when the conversation included static resources lik html pages.

3. URL Rewriting

OriginalURL: http://server:port/servlet/ServletName
RewrittenURL:http://server:port/servlet/ServletName?sessionid=7456
When a request is made, additional parameter is appended with the url. In general added additionalparameter willbesessionidorsometimestheuserid. It willsufficetotrackthesession. This type of session tracking doesn'tneed any special supportfrom the browser. Disadvantage is, implementing this type ofsession tracking is tedious. We need to keep trackofthe parameter as a chain link until the conversation completes and also should make sure that, the parameter doesn't clash with other application parameters.

4. Cookies

Cookies are the mostlyused technologyfor session tracking. Cookie is a key value pair of information, sent bytheservertothebrowser.Thisshould besaved bythe browserinitsspace inthe client computer. Whenever the browser sendsa request tothat server it sendsthe cookie along with it. Then the server can identifythe client using the cookie.
Injava,followingisthesourcecodesnippettocreateacookie:

```
Cookiecookie=newCookie(―userID‖,―7456‖);
res.addCookie(cookie);
```

Sessiontracking is easyto implement and maintainusing the cookies. Disadvantage isthat, the users can opt to disable cookies using their browser preferences. In such case, the browser will not save the cookie at client computer and session tracking fails.

5. SessiontrackingAPI

Session tracking API is built on top ofthe first four methods. This is inorder to help the developer tominimizetheoverheadofsessiontracking. Thistypeofsessiontracking isprovided bythe underlying technology. Lets take the java servlet example. Then, the servlet container manages the session tracking task and the user need not do it explicitlyusing the java servlets.
This isthebest ofallmethods,becauseallthe management anderrorsrelatedto sessiontracking will be taken care of by the container itself.

Everyclient of the server will be mapped with a javax.servlet.http.HttpSession object. Java servletscanusethesessionobjectto storeandretrieve javaobjectsacrossthesession. Session tracking is at the best when it is implemented using session tracking api.

```
package com.journaldev.servlet.session;

importjava.io.IOException;
import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
importjavax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
importjavax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
importjavax.servlet.http.HttpServletResponse;

/**
 *Servletimplementationclass LoginServlet
 */
@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
   privatestaticfinallongserialVersionUID=1L;
   private final String userID = "Pankaj";
   privatefinalStringpassword= "journaldev";

   protectedvoiddoPost(HttpServletRequestrequest,
         HttpServletResponseresponse)throwsServletException,IOException{

      //getrequestparametersforuserIDandpassword String
      user = request.getParameter("user");
      Stringpwd=request.getParameter("pwd");

      if(userID.equals(user)&&password.equals(pwd)){
         Cookie loginCookie = new Cookie("user",user);
         //setting cookie to expiry in 30 mins
         loginCookie.setMaxAge(30*60);
         response.addCookie(loginCookie);
         response.sendRedirect("LoginSuccess.jsp");
      }else{
         RequestDispatcherrd=getServletContext().getRequestDispatcher("/login.html");
         PrintWriter out= response.getWriter();
         out.println("<fontcolor=red>Eitherusernameorpasswordiswrong.</font>"); rd.include(request,
         response);
      }

   }

}
```

JSP

JSPtechnologyisusedto createwebapplicationjust likeServlet technology.It canbethought of as an extension to servlet because it provides more functionalitythan servlet such as expression language, jstl etc.

A JSP page consists ofHTML tags and JSP tags. The jsp pages are easier to maintain than servlet becausewecanseparatedesigninganddevelopment. Itprovidessomeadditionalfeatures such as Expression Language, Custom Tag etc.

Advantage ofJSPoverServlet

Thereare manyadvantagesofJSPoverservlet.Theyare asfollows:

1) ExtensiontoServlet

JSPtechnologyistheextensiontoservlet technology. Wecanuseallthe featuresofservlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP, that makes JSP development easy.

2) Easytomaintain

JSP canbeeasilymanaged becausewecaneasilyseparateour business logicwithpresentation logic. In servlet technology, we mix our business logic with the presentation logic.

3) FastDevelopment:Noneed torecompileandredeploy

IfJSPpage is modified, wedon't needtorecompileandredeploytheproject.Theservlet code needs to be updated and recompiled ifwe have to change the look and feelofthe application.

4) Lesscode thanServlet

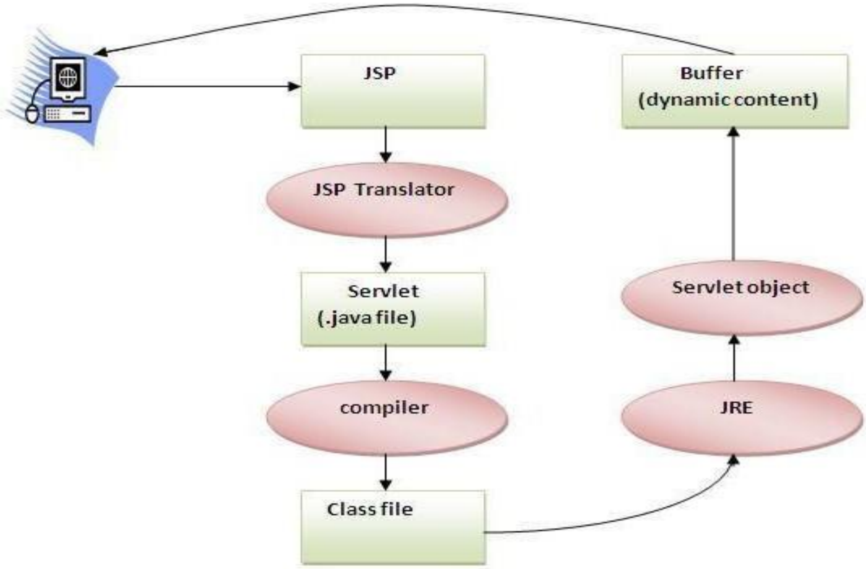
InJSP, wecanuse a lot oftagssuchasactiontags, jstl, customtagsetc.that reducesthecode. Moreover, we can use EL, implicit objects etc.

Lifecycleofa JSPPage

TheJSPpages followsthesephases:

- Translation of JSP Page
- CompilationofJSPPage
- Classloading (class file is loaded by the classloader)
- Instantiation(ObjectoftheGeneratedServlet iscreated).

Initialization(jspInit()methodisinvokedbythecontainer).
Reqeust processing(_jspService()methodisinvokedbythecontainer).
Destroy (jspDestroy() method is invoked by the container).



Asdepictedintheabovediagram,JSP pageistranslated intoservletbythehelpofJSP translator. The JSP translator is a part ofwebserver that is responsible to translate the JSP page into servlet.Afterthat Servlet pageiscompiled bythecompilerandgetsconvertedintotheclass file. Moreover, all the processes that happens in servlet is performed on JSP later like initialization, committing response to the browser and destroy.

CreatingasimpleJSPPage

Tocreatethefirst jsppage, writesomehtmlcodeasgivenbelow, andsave it by.jspextension. We have save this file as index.jsp. Put it in a folder and pastethe folder in the web-apps directory in apache tomcat to run the jsp page.

index.jsp

Let'sseethesimpleexampleofJSP, hereweareusingthescriptlet tagto putjavacodeinthe JSP page. We will learn scriptlet tag later.

```
<html>
<body>
<%out.print(2*5);%>
</body>
</html>
```

It willprint**10**onthe browser.

HowtorunasimpleJSPPage?

FollowthefollowingstepstoexecutethisJSPpage:

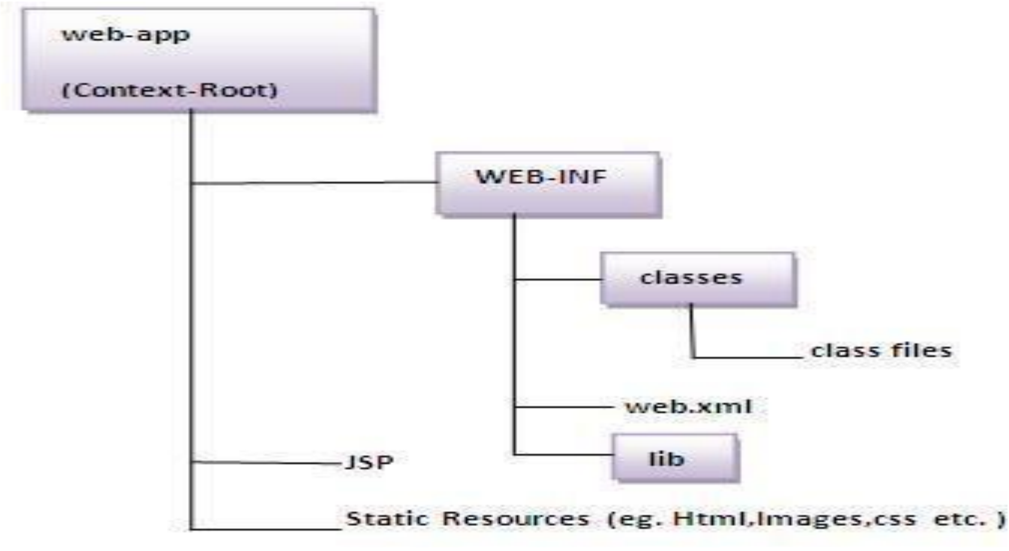
Starttheserver
putthejspfileinafolderanddeployontheserver
visit thebrowserbytheurlhttp://localhost:portno/contextRoot/jspfilee.g.
http://localhost:8888/myapplication/index.jsp

DoIneedtofollowdirectorystructuretorun asimpleJSP?

No,there is no need ofdirectorystructure if you don't have class files or tld files. For example, putjsp filesina folderdirectlyanddeploythat folder.It willberunning fine.But ifyouareusing bean class, Servlet or tld file then directorystructure is required.

Directorystructureof JSP

ThedirectorystructureofJSP pageissameasservlet. Wecontainsthe jsppageoutsidethe WEB-INF folder or in any directory.



JavaServerPagesStandard Tag

TheJavaServer PagesStandardTagLibrary(JSTL) isacollectionofusefulJSP tagswhich encapsulates core functionality common to many JSP applications.

JSTLhassupport for common, structuraltasks suchas iterationand conditionals, tags for manipulating XMLdocuments, internationalizationtags, andSQLtags. It also provides a framework for integrating existing custom tags with JSTL tags.

TheJSTLtagscanbeclassified, accordingtotheir functions, intofollowing JSTLtaglibrary groups that can be used when creating a JSP page:

Core Tags
Formattingtags
SQL tags
XML
tagsJSTLFunct
ions

InstallJSTLLibrary:

IfyouareusingApacheTomcatcontainerthenfollowthefollowingtwosimplesteps:

- DownloadthebinarydistributionfromApacheStandardTaglibandunpackthe compressed file.
- To usetheStandard Taglib fromitsJakartaTaglibsdistribution, simplycopythe JAR files in the distribution's 'lib' directoryto your application's webapps\ROOT\WEB- INF\lib directory.

Touseanyofthe libraries, you must includea<taglib>directiveatthetopofeachJSP that uses the library.

CoreTags:

Thecoregroupoftagsarethemost frequentlyused JSTLtags.Following isthesyntaxto include JSTL Core library in your JSP:

```
<%@ taglib prefix="c"
        uri="http://java.sun.com/jsp/jstl/core"%>
```

Therearefollowing CoreJSTLTags:

Tag	Description
<c:out>	Like<%=...>,butforexpressions.
<c:set >	Setstheresultofanexpressionevaluationin a 'scope'
<c:remove>	Removesascopedvariable(fromaparticularscope,ifspecified).

<c:catch>	CatchesanyThrowablethat occursinitsbodyand optionally exposes it.
<c:if>	Simpleconditionaltagwhichevalutes itsbodyifthesupplied condition is true.
<c:choose>	Simpleconditionaltagthatestablishesacontext formutually exclusive conditional operations, marked by <when> and <otherwise>
<c:when>	Subtagof<choose>that includes itsbodyifitsconditionevalutes to 'true'.
<c:otherwise>	Subtagof<choose>that follows<when>tagsand runsonlyifall of the prior conditions evaluated to 'false'.
<c:import>	RetrievesanabsoluteorrelativeURLandexposes itscontentsto either the page, a String in 'var', or a Reader in 'varReader'.
<c:forEach>	Thebasiciterationtag,acceptingmanydifferent collectiontypes and supporting subsetting and other functionality .
<c:forTokens>	Iteratesovertokens,separatedbythesupplieddelimeters.
<c:param>	Addsa parametertoa containing'import'tag'sURL.
<c:redirect>	Redirectsto anewURL.
<c:url>	Creates aURL withoptionalqueryparameters

Formattingtags:

TheJSTLformattingtagsareusedtoformat and displaytext,thedate,thetime, and numbers for internationalized Web sites. Following is the syntax to include Formatting libraryin your JSP:

```
<%@ taglib prefix="fmt"
     uri="http://java.sun.com/jsp/jstl/fmt"%>
```

FollowingisthelistofFormattingJSTLTags:

Tag	Description
<fmt:formatNumber>	Torendernumericalvaluewithspecificprecision or format.
<fmt:parseNumber>	Parsesthestringrepresentationofanumber,currency,or percentage.
<fmt:formatDate>	Formatsadateand/ortimeusingthesupplied stylesandpattern
<fmt:parseDate>	Parsesthestringrepresentationofadate and/ortime
<fmt:bundle>	Loadsaresourcebundletobeusedbyitstag body.
<fmt:setLocale>	Storesthegivenlocaleinthelocaleconfigurationvariable.
<fmt:setBundle>	Loadsaresourcebundleandstoresit inthenamed scopedvariable or the bundle configuration variable.

<fmt:timeZone>	Specifiesthetimezone foranytime formattingorparsingactions nested in its body.
<fmt:setTimeZone>	Storesthegiventime zoneinthetimezone configurationvariable
<fmt:message>	Todisplayaninternationalizedmessage.
<fmt:requestEncoding>	Setstherequestcharacter encoding

SQLtags:

TheJSTLSQLtag libraryprovidestags for interactingwithrelationaldatabases(RDBMSs) such as Oracle, mySQL, or Microsoft SQL Server.

FollowingisthesyntaxtoincludeJSTLSQLlibraryinyourJSP:

```
<%@ taglib prefix="sql"
        uri="http://java.sun.com/jsp/jstl/sql"%>
```

FollowingisthelistofSQLJSTLTags:

Tag	Description
<sql:setDataSource>	CreatesasimpleDataSourcesuitableonlyforprototyping
<sql:query>	ExecutestheSQLquerydefined in its bodyorthroughthesql attribute.
<sql:update>	ExecutestheSQLupdatedefined inits bodyorthroughthesql attribute.
<sql:param>	SetsaparameterinanSQLstatement tothespecifiedvalue.
<sql:dateParam>	SetsaparameterinanSQLstatementtothespecifiedjava.util.Date value.
<sql:transaction>	Provides nested database action elements with a shared Connection,set uptoexecuteallstatementsasonetransaction.

XML tags:

TheJSTLXMLtagsprovideaJSP-centricwayofcreating and manipulating XMLdocuments. Following is the syntax to include JSTL XML library in your JSP.

TheJSTLXMLtag libraryhascustomtagsfor interacting withXMLdata.This includesparsing XML, transforming XML data, and flow control based on XPath expressions.

```
<%@ taglib prefix="x"
        uri="http://java.sun.com/jsp/jstl/xml"%>
```

Before youproceedwiththeexamples, youwouldneedto copyfollowingtwoXMLandXPath related libraries into your <Tomcat Installation Directory>\lib:

XercesImpl.jar: Downloaditfromhttp://www.apache.org/dist/xerces/j/
xalan.jar: Downloadit fromhttp://xml.apache.org/xalan-j/index.html

Following is the list of XML JSTL Tags:

Tag	Description
<x:out>	Like<%=...>,butforXPathexpressions.
<x:parse>	UsetoparseXMLdataspecifiedeither viaanattributeorinthetag body.
<x:set>	Setsa variabletothevalueofanXPathexpression.
<x:if>	Evaluatesatest XPathexpressionandifit istrue,it processesits body. If the test condition is false, the body is ignored.
<x:forEach>	ToloopovernodesinanXMLdocument.
<x:choose>	Simpleconditionaltagthatestablishesacontext formutually exclusive conditional operations, marked by <when> and <otherwise>
<x:when>	Subtagof<choose>that includes itsbodyifitsexpressionevalutes to 'true'
<x:otherwise>	Subtagof<choose>that follows<when>tagsandrunsonlyifall of the prior conditions evaluated to 'false'
<x:transform>	AppliesanXSLtransformationonaXMLdocument
<x:param>	Usealongwiththetransformtagtoset aparameterintheXSLT stylesheet

JSTL Functions:

JSTLincludesanumber ofstandard functions, mostofwhicharecommonstring manipulation functions. Following is the syntax to include JSTL Functions libraryin your JSP:

```
<%@ taglib prefix="fn"
        uri="http://java.sun.com/jsp/jstl/functions"%>
```

FollowingisthelistofJSTLFunctions:

Function	Description
fn:contains()	Testsifaninputstringcontainsthespecifiedsubstring.
fn:containsIgnoreCase()	Testsifaninput stringcontainsthespecifiedsubstringinacase insensitive way.
fn:endsWith()	Testsifaninputstringendswiththespecifiedsuffix.
fn:escapeXml()	Escapescharactersthatcouldbeinterpreted asXMLmarkup.
fn:indexOf()	Returnstheindexwithing astring ofthe first occurrenceofa

	specifiedsubstring.
fn:join()	Joinsallelements ofanarrayintoa string.
fn:length()	Returnsthenumberofitems inacollection, orthenumber of characters in a string.
fn:replace()	Returnsastringresultingfromreplacinginaninput stringall occurrences with a given string.
fn:split()	Splitsa stringintoanarrayofsubstrings.
fn:startsWith()	Testsifan inputstringstartswiththespecified prefix.
fn:substring()	Returnsa subsetofastring.
fn:substringAfter()	Returnsasubsetofastringfollowingaspecificsubstring.
fn:substringBefore()	Returnsasubsetofastringbeforeaspecific substring.
fn:toLowerCase()	Convertsallofthecharactersofa stringtolowercase.
fn:toUpperCase()	Converts allofthecharactersofastringtouppercase.
fn:trim()	Removeswhite spacesfrombothendsofastring.

CreatingHTMLformsbyembeddingJSPcode

To startofftheexplorationofHTMLforms, it's best to start withasmallformand expand from there.Also,it'sbettertostart withaJSPratherthanaservlet,because it iseasierto writeoutthe HTML. Most ofthe form handling for JSPs and servlets is identical, so after you know how to retrieve forminformation fromaJSP, youknowhowtodoit fromaservlet. Listing3.1showsan HTML file containing a simple input formthat calls a JSP to handle the form.

```
<html>
<body>

<h1>Pleasetellmeaboutyourself</h1>

<formaction="SimpleFormHandler.jsp"method="get">

Name: <input type="text" name="firstName">
  <inputtype="text"name="lastName"><br>Se
x:
  <inputtype="radio"checkedname="sex"value="male">Male
  <inputtype="radio"name="sex"value="female">Female
<p>
WhatJavaprimitivetypebestdescribesyourpersonality:
<selectname="javaType">
  <optionvalue="boolean">boolean</option>
  <optionvalue="byte">byte</option>
  <optionvalue="char"selected>char</option>
  <optionvalue="double">double</option>
  <optionvalue="float">float</option>
  <optionvalue="int">int</option>
  <optionvalue="long">long</option>
</select>
<br>
<inputtype="submit">
</form>
</body>
</html>
```

TheSimpleFormHandler JSPdoeslittle morethanretrievetheformvariablesandprintout their values. Listing 3.2 shows the contentsofSimpleFormHandler.jsp, which you cansee is pretty short.

```
<html>
<body>

<%

//Grabthevariablesfromtheform. String
  firstName =
  request.getParameter("firstName"); String
  lastName=request.getParameter("lastName");
  String sex = request.getParameter("sex");
  String javaType=request.getParameter("javaType");
%>
<%--Printoutthevariables.--%>
<h1>Hello,<%=firstName%><%=lastName%>!</h1>
Iseethatyouare<%=sex%>. Youknow, youremindmeofa
<%=javaType%>variableIonceknew.

</body>
</html>
```

Unit–IV

PHP

When working with data values in PHP, weneed some convenient way to store thesevalues so that we can easily access themand make reference to them whenever necessary. This is where PHP variables comein. Itisoftenusefultothinkof variablesascomputer memorylocations wheredata istobestored. Whendeclaringa variableinPHP it isassigneda namethatcanbeusedtoreferenceitinother locations in the PHP script. The value of the variable can be accessed, the value can be changed, and the type of variable can be altered all by referencing the name assigned at variable creation time.

NamingandCreatingaVariablein PHP

Before learning how to declare a variable in PHP it is first important to understand some rules about variablenames(alsoknownas*variablenamingconventions*).AllPHPvariablenames mustbepre-fixed witha *$*.It is this prefix whichinforms thePHP pre-processor that itis dealingwitha variable. Thefirst character of thename must be either a letter or an underscore(_). The remaining characters must compriseonlyofletters, numbersor underscores. Allother charactersaredeemedtobeinvalidfor usein a variable name. Let's look at some valid and invalid PHP variable names:

```
$_myName//valid
$myName//valid
$myvar//valid
$myVar21// valid
$_1Big// invalid-underscoremustbefollowedbyaletter
$1Big    //invalid-mustbeginwitha letter orunderscore
$_er-t//invalidcontainsnonalphanumericcharacter(-)
```

Variablenames inPHParecase-sensitive. This meansthatPHPconsiders*$_myVariable*tobea completely different variable to one that is named "*$_myvariable*.

AssigningaValuetoaPHPVariable

Values are assigned to variables using the PHP *assignment operator*. The assignment operator is represented by the = sign. Toassign a value to a variable therefore, the variable name is placed on the left of the expression, followed by the assignment operator. The value to be assigned is then placed to theright oftheassignment operator.Finallytheline, aswithallPHPcodestatements, is terminatedwith a semi-colon (;).

Let'sbeginbyassigningtheword"Circle"toa variablenamedmyShape:

```
$myShape="Circle";
```

Wehavenowdeclareda variablewiththename*myShape*andassigneda stringvaluetoit of"Circe". We can similarly declare a variable to contain an integer value:

```
$numberOfShapes=6;
```

Theaboveassignment creates a variable named *numberOfShapes* andassigns it a numeric value of 6. Once a variable has beencreated, the valueassignedto that variablecanbechangedat anytime using the same assignment operator approach:

```
<?php
$numberOfShapes=6;//Setinitial values
$myShape="Circle";
$numberOfShapes=7;//Changetheinitialvalues tonew values
$myShape="Square";

?>
```

AccessingPHPVariableValues

Nowthat we have learned how to createa variableand assignan initialvalueto it we now need tolook at howtoaccess thevaluecurrentlyassignedtoa variable. In practice, accessinga variableis as simple as referencing the name it was given when it was created.

For example, if wewanttodisplaythevaluewhichweassignedtoour *numberOfShapes*variablewecan simply reference it in our *echo* command:

```
<?php
echo"Thenumber ofshapesis $numberOfShapes.";
?>
```

Thiswillcausethefollowingoutputtoappearinthebrowser:

```
Thenumber ofshapesis6.
```

Similarlywecandisplaythevalueofthe*myShape*variable:

```
<?php
echo"$myShapeisthevalueofthecurrentshape.";
?>
```

Theexamples we haveusedfor accessingvariablevalues arestraightforwardbecause wehavealways hada spacecharacterafterthevariablename. Thequestionarises astowhatshouldbedoneif weneed to put other characters immediately after the variable name. For example:

```
<?php
```

```
echo"TheCircleisthe$numberOfShapesthshape";
?>
```

What wearelookingforinthisscenariois outputasfollows:

```
TheCircleisthe6thshape.
```

UnfortunatelyPHP willseethe*th* ontheendofthe$numberOfShapes variablenameas beingpartofthe name. It will then try to output thevalueof a variablecalled $numberOfShapesth, which does not exist. This results in nothing being displayed for this variable:

```
TheCircleistheshape.
```

Fortunatelywecanget aroundthis issuebyplacingbraces ({and})aroundthevariablenameto distinguish the name from any other trailing characters:

```
<?php
echo"TheCircleisthe${numberOfShapes}thshape";
?>
```

Togiveusthedesiredoutput:

```
                    TheCircleisthe6thshape.
```

Internal(built-in)functions

PHP comes standard with many functions and constructs. There arealso functions that require specific PHPextensionscompiledin,otherwisefatal"undefinedfunction"errorswillappear.For example, touse image functions such as imagecreatetruecolor(), PHP must be compiled with GD support. Or, to use mysql_connect(), PHPmust becompiledwithMySQLsupport.Therearemanycorefunctions thatare included in every version of PHP, suchas the string and variablefunctions. A call to phpinfo()or get_loaded_extensions() will show which extensions are loaded into PHP. Also note that many extensions are enabled by default and that the PHP manual is split up by extension. See the configuration, installation, and individual extension chapters, for information on how to set up PHP.

Reading and understanding a function's prototype is explained within the manual section titled how to reada functiondefinition. It's important torealizewhata functionreturns or if a functionworks directly on a passed in value. For example, str_replace()will return the modified string while usort()works on theactualpassedinvariableitself. Eachmanualpagealsohas specificinformationfor eachfunctionlike information on function parameters, behavior changes, return values for both success and failure, and availability information. Knowing these important (yet often subtle) differences is crucial for writing correct PHP code.

PHPUserDefinedFunctions

Besidesthebuilt-inPHPfunctions, wecancreateourownfunctions.
Afunctionisa blockofstatementsthatcanbeusedrepeatedlyina program. A
function will not execute immediately when a page loads.
Afunction willbeexecutedbyacalltothefunction.

CreateaUserDefinedFunctioninPHP

Auser definedfunctiondeclarationstartswiththeword"function":

Syntax

```
functionfunctionName(){
   code tobe executed;
}
```

In the examplebelow, wecreatea function named "writeMsg()". Theopeningcurlybrace({) indicates thebeginning of thefunctioncodeandthe closingcurlybrace (}) indicates the end of the function. The function outputs "Hello world!". To call the function, just write its name:

Example

```
<?php
function writeMsg() {
   echo"Helloworld!";
}

writeMsg();//callthefunction
?>
```

PHPFunctionArguments

Informationcanbepassedtofunctionsthrougharguments. Anargumentisjust likeavariable.

Argumentsarespecifiedafter thefunctionname, insidetheparentheses. Youcanaddasmanyarguments as you want, just seperate them with a comma.

Thefollowingexamplehasa functionwithoneargument($fname). WhenthefamilyName()functionis called, we also pass along a name (e.g. Jani), and the name is used inside the function, which outputs several different first names, but an equal last name:

Example

```
<?php
functionfamilyName($fname){
   echo"$fnameRefsnes.<br>";
}

familyName("Jani");
familyName("Hege");
familyName("Stale");
familyName("KaiJim");
familyName("Borge");
?>
```

Example

```
<?php
functionfamilyName($fname,$year){
   echo"$fnameRefsnes.Bornin$year<br>";
}

familyName("Hege","1975");
familyName("Stale","1978");
familyName("KaiJim","1983");
?>
```

PHPDefaultArgumentValue

Thefollowingexampleshowshowtousea defaultparameter. Ifwecallthefunctionsetheight() without arguments it takes the default value as argument:

Example

```
<?php
function setHeight($minheight = 50) {
   echo"Theheightis:$minheight<br>";
}

setHeight(350);
setHeight();//willusethedefaultvalueof50
setHeight(135);
setHeight(80);
?>
```

PHPFunctions-Returningvalues

Tolet afunctionreturnavalue,usethereturnstatement:

Example

```
<?php
functionsum($x,$y){
   $z=$x+$y;
   return $z;
}
echo"5+10=".sum(5,10)."<br>";
echo"7 +13= ". sum(7, 13) . "<br>"; echo
"2 + 4 = " . sum(2, 4);
?>
```

ConnectingtoaDatabase

PHP5andlatercanworkwithaMySQLdatabaseusing:

MySQLiextension(the"i"standsforimproved)
PDO(PHPDataObjects)

ShouldI UseMySQLiorPDO?

Ifyouneedashortanswer,itwouldbe"Whateveryoulike". Both

MySQLi and PDO have their advantages:

PDOwillworkon12differentdatabasesystems, whereasMySQLiwillonlyworkwithMySQL databases.

So,ifyouhavetoswitchyour projecttouseanother database,PDOmakestheprocess easy. Youonly have to change the connection string and a few queries. With MySQLi, you will need to rewrite the entire code - queries included.

Bothareobject-oriented,butMySQLialsooffersaproceduralAPI.

BothsupportPreparedStatements. PreparedStatementsprotectfromSQLinjection,andarevery important for web application security.

MySQLExamplesinBothMySQLiandPDOSyntax

Inthis,andinthefollowingchapterswedemonstratethreewaysofworkingwithPHP andMySQL:

- MySQLi(object-oriented)
- MySQLi (procedural)
- PDO

MySQLiInstallation

ForLinuxandWindows:TheMySQLiextensionisautomaticallyinstalledinmostcases,whenphp5 mysqlpackageis installed.

Forinstallationdetails,goto:http://php.net/manual/en/mysqli.installation.php

PDOInstallation

Forinstallationdetails,goto:http://php.net/manual/en/pdo.installation.php

OpenaConnectiontoMySQL

BeforewecanaccessdataintheMySQLdatabase,weneedtobeabletoconnecttotheserver:

Example(MySQLiObject-Oriented)

```
<?php
$servername="localhost";
$username="username";
$password="password";

// Createconnection
$conn=newmysqli($servername,$username,$password);

//Checkconnection
if($conn->connect_error){
   die("Connectionfailed:".$conn->connect_error);
}
echo"Connectedsuccessfully";
?>
```

Example(MySQLiProcedural)

```
<?php
$servername="localhost";
$username="username";
$password="password";

// Createconnection
$conn=mysqli_connect($servername,$username,$password);

//Checkconnection
```

```
if(!$conn){
   die("Connectionfailed:". mysqli_connect_error());
}
echo"Connectedsuccessfully";
?>
```

Example(PDO)

```
<?php
$servername="localhost";
$username="username";
$password="password";

try{
   $conn=newPDO("mysql:host=$servername;dbname=myDB",$username,$password);
   //setthePDOerrormodetoexception
   $conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
   echo"Connectedsuccessfully";
   }
catch(PDOException$e)
   {
   echo"Connectionfailed:". $e->getMessage();
   }
?>
```

ClosetheConnection

Theconnectionwillbeclosedautomaticallywhenthescript ends. Toclosetheconnectionbefore, use the following:

Example(MySQLiObject-Oriented)

```
$conn->close();
```

Example(MySQLiProcedural)

```
mysqli_close($conn);
```

Example(PDO)

```
$conn=null;
```

UsingCookies

A cookie is often used to identify a user. A cookieis a small file that theserver embeds on the user's computer. Eachtimethesamecomputer requestsapagewitha browser, itwillsendthecookietoo. With PHP, you can both create and retrieve cookie values.

CreateCookiesWithPHP

Acookieiscreatedwiththesetcookie()function.

Syntax

setcookie(*name*,*value*,*expire*,*path*,*domain*,*secure*,*httponly*);

Onlythe*name*parameterisrequired.Allotherparametersareoptional.

PHPCreate/RetrieveaCookie

Thefollowingexamplecreates acookienamed"user"withthevalue"JohnDoe". Thecookiewill expire after 30days (86400 *30).The"/" means thatthecookieis availablein entirewebsite(otherwise, select the directory you prefer).

Wethenretrievethevalueof thecookie"user"(usingtheglobalvariable$_COOKIE). Wealsousethe isset() function to find out if the cookie is set:

Example

```
<?php
$cookie_name="user";
$cookie_value="JohnDoe";
setcookie($cookie_name,$cookie_value,time()+(86400*30),"/");//86400=1day
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
   echo"Cookienamed'".$cookie_name. "'isnot set!";
}else{
   echo"Cookie'".$cookie_name."'isset!<br>"; echo
   "Value is: " . $_COOKIE[$cookie_name];
}
?>
```

```
</body>
</html>
```

ModifyaCookieValue

Tomodifya cookie, justset(again)thecookieusingthesetcookie()function:

Example

```
<?php
$cookie_name="user";
$cookie_value="AlexPorter";
setcookie($cookie_name,$cookie_value,time()+(86400*30),"/");
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
   echo"Cookienamed'".$cookie_name. "'isnotset!";
}else{
   echo"Cookie'".$cookie_name."'isset!<br>"; echo
   "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```

DeleteaCookie

Todeleteacookie,usethesetcookie()functionwithanexpirationdateinthepast:

Example

```
<?php
//settheexpirationdatetoonehourago
setcookie("user", "", time() - 3600);
?>
<html>
<body>

<?php
echo"Cookie'user'is deleted.";
?>
```

```
</body>
</html>
```

CheckifCookiesareEnabled

Thefollowingexamplecreatesasmallscriptthatcheckswhether cookiesareenabled. First,trytocreate a test cookie with the setcookie() function, then count the $_COOKIE array variable:

Example

```
<?php
setcookie("test_cookie","test",time()+3600,'/');
?>
<html>
<body>

<?phpif(count($_COOKIE)
>0){
   echo"Cookiesareenabled.";
}else{
   echo"Cookiesaredisabled.";
}
?>

</body>
</html>
```

RegularExpressions

Regular expressions are nothing more than a sequence or pattern of characters itself. They provide the foundation for pattern-matching functionality.

Using regular expression you can search a particular string inside a another string, you can replace one string by another string and you can split a string into many chunks.

PHPoffersfunctionsspecifictotwosetsofregularexpressionfunctions,eachcorrespondingtoa certain type of regular expression. You can use any of them based on your comfort.

- POSIXRegular Expressions

- PERLStyleRegularExpressions

POSIXRegularExpressions

The structure of a POSIX regular expression is not dissimilar to that of a typical arithmetic expression: various elements (operators) are combined to form more complex expressions.

The simplest regular expression is one that matches a single character, such as g, inside strings such as g, haggle, or bag.

Letsgiveexplainationforfewconceptsbeingusedin POSIXregularexpression.Afterthatwewill introduce you wih regular expression related functions.

Brackets

Brackets ([]) have a special meaning when used in the context of regular expressions. They are usedto find a range of characters.

Expression	Description
[0-9]	Itmatchesanydecimaldigitfrom0through9.
[a-z]	Itmatchesanycharacterfromlowercaseathroughlowercasez.
[A-Z]	ItmatchesanycharacterfromuppercaseAthroughuppercaseZ.
[a-Z]	It matchesanycharacterfromlowercaseathroughuppercaseZ.

The ranges shown above are general; you could also use the range [0-3] to match any decimal digit ranging from0 through3, or therange[b-v] to matchany lowercasecharacter ranging fromb through v.

Quantifiers

The frequency or position of bracketed character sequences and single characters can be denoted by a special character. Each pecial character having a specific connotation. The +, *, ?, {int. range}, and $ flags all follow a character sequence.

Expression	Description
p+	It matchesanystringcontainingatleastonep.
p*	Itmatchesanystringcontainingzeroormorep's.
p?	Itmatchesanystringcontainingzeroormorep's. Thisisjustanalternativewaytouse

p{**N**}	Itmatchesanystringcontaininga sequenceof**N**p's
p{2,3}	Itmatchesanystringcontainingasequenceoftwoorthreep's.
p{2,}	Itmatchesanystringcontainingasequenceofatleasttwop's.
p$	Itmatches anystringwithpattheendofit.
^p	Itmatches anystringwithpatthebeginningofit.

Examples

Followingexampleswillclearyourconceptsaboutmatchingchracters.

Expression	**Description**
[^a-zA-Z]	It matchesanystringnot containinganyofthecharactersrangingfroma throughzand through Z.
p.p	Itmatchesanystringcontainingp,followedbyanycharacter,inturnfollowedbyanot
^.{2}$	Itmatchesanystringcontainingexactlytwocharacters.
<b>(.*)</b>	Itmatchesanystringenclosedwithin<b>and</b>.
p(hp)*	It matchesanystringcontaininga pfollowedbyzeroormoreinstancesofthesequenc

PredefinedCharacterRanges

For your programming convenience several predefined character ranges, also known as character classes, are available. Character classes specify an entire range of characters, for example, the alphabet or an integer set:

Expression	**Description**
[[:alpha:]]	It matchesanystringcontainingalphabeticcharactersaAthroughzZ.
[[:digit:]]	Itmatchesanystringcontainingnumericaldigits0through9.
[[:alnum:]]	ItmatchesanystringcontainingalphanumericcharactersaAthroughzZand0 through
[[:space:]]	Itmatchesanystringcontaininga space.

PHP'sRegexpPOSIXFunctions

PHPcurrentlyofferssevenfunctionsforsearchingstringsusingPOSIX-styleregular expressions:

Function	Description
ereg()	Theereg()functionsearchesa stringspecifiedbystringfor astringspecifiedbypa returning true if the pattern is found, and false otherwise.
ereg_replace()	Theereg_replace()functionsearchesfor stringspecifiedbypatternandreplacespa with replacement if found.
eregi()	Theeregi()functionsearchesthroughoutastringspecifiedbypatternfor astringsp by string. The search is not case sensitive.
eregi_replace()	Theeregi_replace()functionoperatesexactlylikeereg_replace(), exceptthatthese for pattern in string is not case sensitive.
split()	Thesplit()functionwilldivideastringintovarious elements,theboundariesof eac element based on the occurrence of pattern in string.
spliti()	Thespliti()functionoperates exactlyinthesamemanner asitssiblingsplit(),exce it is not case sensitive.
sql_regcase()	Thesql_regcase() functioncanbethought of asa utilityfunction, convertingeach character inthe input parameter string into a bracketed expression containingtwo characters.

PERLStyleRegularExpressions

Perl-style regular expressions are similar to their POSIX counterparts. The POSIX syntax can be used almost interchangeably with the Perl-style regular expression functions. In fact, you can use any of the quantifiers introduced in the previous POSIX section.

Lets give explaination for few concepts being used in PERL regular expressions. After that we will introduce you wih regular expression related functions.

Metacharacters

Ametacharacterissimplyanalphabeticalcharacterprecededbyabackslashthatactstogivethe combination a special meaning.

For instance, youcansearchfor large moneysums usingthe '\d' metacharacter:**/([\d]+)000/**, Here**\d** will search for any string of numerical character.

Followingisthelistof metacharacterswhichcanbeusedinPERLStyleRegularExpressions.

Character	Description
.	a singlecharacter

\s	awhitespacecharacter(space,tab,newline)
\S	non-whitespacecharacter
\d	adigit(0-9)
\D	anon-digit
\w	awordcharacter(a-z,A-Z,0-9,_)
\W	anon-wordcharacter
[aeiou]	matches a single character in the given set
[^aeiou]	matchesasinglecharacteroutsidethegivenset
(foo|bar|baz)	matches any of the alternatives specified

Modifiers

Several modifiers areavailablethatcanmake your workwithregexps much easier, likecasesensitivity, searching in multiple lines etc.

Modifier	**Description**
i	Makesthematchcaseinsensitive
m	Specifiesthatifthestringhasnewlineorcarriage return characters, the ^ and $ operators will now match against a newline boundary, instead of a string boundary
o	Evaluatestheexpressiononlyonce
s	Allowsuseof. tomatchanewlinecharacter
x	Allowsyoutousewhitespaceintheexpressionforclarity g
	Globally finds all matches
cg	Allowsasearchtocontinueevenafteraglobalmatchfails

PHP'sRegexpPERLCompatibleFunctions

PHPoffersfollowingfunctionsforsearchingstringsusingPerl-compatibleregularexpressions:

Function	**Description**
preg_match()	Thepreg_match()functionsearchesstringforpattern,returningtrueifpatternex and false otherwise.
preg_match_all()	Thepreg_match_all()functionmatchesalloccurrencesofpatterninstring.
preg_replace()	Thepreg_replace()functionoperatesjustlikeereg_replace(),exceptthatregular expressions can be used in the pattern and replacement input parameters.

preg_split()	Thepreg_split()functionoperates exactlylikesplit(),exceptthatregularexpress are accepted as input parameters for pattern.
preg_grep()	Thepreg_grep()functionsearchesallelementsofinput_array,returningallelem matching the regexp pattern.
preg_quote()	Quoteregularexpressioncharacters

XML

XML is a markup language that looks a lot like HTML. An XML document is plain text and contains tags delimited by < and >.There are two big differences between XML and HTML:

XMLdoesn'tdefineaspecificsetoftagsyoumustuse. XML

is extremely picky about document structure.

XMLgivesyoualotmorefreedomthanHTML.HTMLhasacertainsetoftags:the<a></a>tags surrounda link, the<p> startsa paragraphand so on. An XML document, however, canuseanytags you want. Put <rating></rating> tags around a movie rating, >height></height> tags around someone's height. Thus XML gives you option to device your own tags.

XML is verystrict when it comes to document structure. HTML lets you playfast and loose withsome opening and closing tags. BUt this is not the case with XML.

HTMLlistthat'snotvalidXML

```
<ul>
<li>BraisedSeaCucumber
<li>BakedGibletswithSalt
<li>AbalonewithMarrowandDuck Feet
</ul>
```

Thisisnotavalid XMLdocumentbecause there are no closing </li>tagsto match upwith the three opening <li> tags. Every opened tag in an XML document must be closed.

HTMLlistthatisvalidXML

```
<ul>
<li>BraisedSeaCucumber</li>
<li>BakedGibletswithSalt</li>
<li>AbalonewithMarrowandDuckFeet</li>
```

```
</ul>
```

ParsinganXMLDocument

PHP5'snew**SimpleXML** modulemakesparsinganXMLdocument,well,simple.ItturnsanXML document into an object that provides structured access to the XML.

TocreateaSimpleXMLobjectfromanXMLdocument to storedinastring,passthestring **simplexml_load_string( )**. It returns a SimpleXML object.

Example

Tryout followingexample:

```
<?php

$channel=<<<_XML_
<channel>
<title>What'sForDinner<title>
<link>http://menu.example.com/<link>
<description>Choosewhattoeattonight.</description>
</channel>
_XML_;

$xml=simplexml_load_string($channel);
print"The$xml->titlechannelisavailableat$xml->link.";
print "The description is \"$xml->description\"";
?>
```

Itwillproducefollowingresult:

```
TheWhat'sForDinnerchannelisavailableathttp://menu.example.com/.Thedescriptionis"Choose what to
eat tonight."
```

NOTE: Youcanusefunction**simplexml_load_file(filename)**ifyouhaveXMLcontentina file.

Fora completedetailofXMLparsingfunctioncheck**PHPFunctionReference**.

GeneratinganXMLDocument

scratch.

The easiestway to generate an XML document is to build a PHP array whose structure mirrors that of the XML document and then to iterate through the array, printing each element with appropriate formatting.

Example

Tryout followingexample:

```
<?php


$channel=array('title'=>"What'sForDinner",
               'link'=>'http://menu.example.com/',
               'description'=>'Choosewhattoeattonight.');
print "<channel>\n";
foreach($channelas$element=>$content){
  print "<$element>";
  print htmlentities($content);
  print "</$element>\n";
}
print"</channel>";
?>
```

Itwillproducefollowingresult:

```
<channel>
<title>What'sForDinner</title>
<link>http://menu.example.com/</link>
<description>Choosewhattoeattonight.</description>
</channel></html>
```

DOM

ADOM(DocumentObjectModel)definesastandardwayfor accessingandmanipulatingdocuments. The

XML DOM defines a standard way for accessing and manipulatingXML documents.

TheXMLDOMviewsanXMLdocumentasatree-structure.

AllelementscanbeaccessedthroughtheDOMtree. Their content(textandattributes)canbemodifiedor deleted, and new elements can be created. The elements, their text, and their attributes areall known as nodes.

YoucanlearnmoreabouttheXMLDOMinourXMLDOMtutorial.

TheHTMLDOM

TheHTMLDOMdefinesastandardwayforaccessingandmanipulatingHTMLdocuments. All

HTML elements can be accessed through the HTML DOM.

YoucanlearnmoreabouttheHTMLDOMinourJavaScripttutorial.

LoadanXMLFile-Cross-browserExample

Thefollowingexampleparsesan XMLdocument("note.xml")intoanXMLDOMobjectandthen extracts some info from it with a JavaScript:

Example

```
<html>
<body>
<h1>W3SchoolsInternalNote</h1>
<div>
<b>To:</b><spanid="to"></span><br/>
<b>From:</b><spanid="from"></span><br/>
<b>Message:</b><spanid="message"></span>
</div>

<script>
if(window.XMLHttpRequest)
  {//codeforIE7+,Firefox,Chrome,Opera,Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {//codeforIE6, IE5
  xmlhttp=newActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","note.xml",false);
xmlhttp.send();
```

```
xmlDoc=xmlhttp.responseXML;

document.getElementById("to").innerHTML=
xmlDoc.getElementsByTagName("to")[0].childNodes[0].nodeValue;
document.getElementById("from").innerHTML=
xmlDoc.getElementsByTagName("from")[0].childNodes[0].nodeValue;
document.getElementById("message").innerHTML=
xmlDoc.getElementsByTagName("body")[0].childNodes[0].nodeValue;
</script>

</body>
</html>
```

LoadanXMLString-Cross-browserExample

Thefollowingexampleparses anXMLstringintoanXMLDOMobject andthenextractssomeinfofrom it with a JavaScript:

Example

```
<html>
<body>
<h1>W3SchoolsInternalNote</h1>
<div>
<b>To:</b><spanid="to"></span><br/>
<b>From:</b><spanid="from"></span><br/>
<b>Message:</b><spanid="message"></span>
</div>

<script>txt="<note>";
txt=txt+"<to>Tove</to>";
txt=txt+"<from>Jani</from>";
txt=txt+"<heading>Reminder</heading>";
txt=txt+"<body>Don'tforgetmethisweekend!</body>";
txt=txt+"</note>";

if(window.DOMParser)
  {
  parser=new DOMParser();
  xmlDoc=parser.parseFromString(txt,"text/xml");
  }
else//InternetExplorer
  {
  xmlDoc=newActiveXObject("Microsoft.XMLDOM");
  xmlDoc.async=false;
  xmlDoc.loadXML(txt);
  }

document.getElementById("to").innerHTML=
xmlDoc.getElementsByTagName("to")[0].childNodes[0].nodeValue;
document.getElementById("from").innerHTML=
xmlDoc.getElementsByTagName("from")[0].childNodes[0].nodeValue;
document.getElementById("message").innerHTML=
xmlDoc.getElementsByTagName("body")[0].childNodes[0].nodeValue;
</script>
</body>
</html>
```

DocumentTypeDefinition

AnXMLdocumentwithcorrectsyntaxiscalled"WellFormed".

AnXMLdocumentvalidatedagainstaDTDis"WellFormed"and"Valid".

ValidXML Documents

A"Valid"XMLdocument isa "WellFormed"XMLdocument, whichalsoconformstotherulesofa DTD:

```
<?xmlversion="1.0"encoding="UTF-8"?>
<!DOCTYPEnoteSYSTEM"Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don'tforgetmethisweekend!</body>
</note>
```

TheDOCTYPEdeclaration, intheexampleabove, isareferencetoanexternalDTDfile. Thecontent of the file is shown in the paragraph below.

XMLDTD

Thepurposeofa DTDis todefinethestructureofanXMLdocument. It defines thestructurewitha list of legal elements:

```
<!DOCTYPEnote
[
<!ELEMENTnote(to,from,heading,body)>
<!ELEMENTto(#PCDATA)>
<!ELEMENTfrom(#PCDATA)>
<!ELEMENTheading(#PCDATA)>
<!ELEMENTbody(#PCDATA)>
]>
```

TheDTDaboveisinterpretedlikethis:

!DOCTYPEnotedefinesthattherootelementofthedocumentisnote
!ELEMENT notedefinesthatthenoteelement mustcontainfourelements:"to,from,heading, body"
!ELEMENTtodefines thetoelementtobeoftype"#PCDATA"
!ELEMENTfromdefinesthefromelement tobeoftype"#PCDATA"
!ELEMENTheadingdefinestheheadingelementtobeoftype"#PCDATA"
!ELEMENT bodydefinesthebodyelementtobeoftype"#PCDATA"

UsingDTDfor EntityDeclaration

Adoctypedeclarationcanalsobeusedtodefinespecial charactersandcharacter strings, usedinthe document:

Example

```
<?xmlversion="1.0"encoding="UTF-8"?>

<!DOCTYPEnote[
<!ENTITYnbsp"&#xA0;">
<!ENTITYwriter"Writer:DonaldDuck.">
<!ENTITYcopyright"Copyright:W3Schools.">
]>

<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don'tforget methisweekend!</body>
<footer>&writer;&nbsp;&copyright;</footer>
</note>
```

WhyUseaDTD?

Witha DTD, independentgroupsofpeoplecanagreeona standardfor interchangingdata. With a DTD, you can verify that the data you receive fromthe outside world is valid.

DisplayingXMLwithXSLT

XSLT(eXtensibleStylesheetLanguageTransformations)istherecommendedstylesheet languagefor XML.

XSLT isfar moresophisticatedthanCSS.WithXSLTyoucanadd/removeelementsandattributestoor fromthe output file. You can also rearrangeand sort elements, performtests and makedecisions about which elements to hide and display, and a lot more.

XSLTusesXPathtofindinformationinanXML document.

XSLT Example

WewillusethefollowingXMLdocument:

```
<?xmlversion="1.0"encoding="UTF-8"?>
<breakfast_menu>

<food>
<name>BelgianWaffles</name>
<price>$5.95</price>
<description>TwoofourfamousBelgianWaffleswithplentyofrealmaplesyrup</description>
<calories>650</calories>
</food>

<food>
<name>StrawberryBelgianWaffles</name>
<price>$7.95</price>
<description>LightBelgianwafflescoveredwithstrawberriesandwhippedcream</description>
<calories>900</calories>
</food>

<food>
<name>Berry-BerryBelgianWaffles</name>
<price>$8.95</price>
<description>LightBelgianwafflescoveredwithanassortmentoffreshberriesandwhipped cream</description>
<calories>900</calories>
</food>

<food>
<name>FrenchToast</name>
<price>$4.50</price>
<description>Thickslicesmadefromourhomemadesourdoughbread</description>
<calories>600</calories>
</food>

<food>
```

```
<name>HomestyleBreakfast</name>
<price>$6.95</price>
<description>Twoeggs,baconorsausage,toast,andourever-popularhashbrowns</description>
<calories>950</calories>
</food>

</breakfast_menu>
```

UseXSLT totransformXMLintoHTML, beforeitisdisplayedina browser: Example

XSLT Stylesheet:

```
<?xmlversion="1.0"encoding="UTF-8"?>
<htmlxsl:version="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<bodystyle="font-family:Arial;font-size:12pt;background-color:#EEEEEE">
<xsl:for-eachselect="breakfast_menu/food">
  <divstyle="background-color:teal;color:white;padding:4px">
    <spanstyle="font-weight:bold"><xsl:value-ofselect="name"/>-</span>
    <xsl:value-ofselect="price"/>
    </div>
  <divstyle="margin-left:20px;margin-bottom:1em;font-size:10pt">
    <p>
    <xsl:value-ofselect="description"/>
    <spanstyle="font-style:italic">(<xsl:value-ofselect="calories"/>caloriesperserving)</span>
    </p>
 <
 /
 d
 i
 v
 >
</xsl:for-each>
</body>
</html>
```

Unit5
AJAX-AsynchronousJavaScriptandXML

1 Introduction

The main topic in this paper is the evaluation and discussion of AJAX - asynchronous Java Script and XML.Technical aspectsare described and an overview aboutthe main concepts isgiven. This paper alsodiscusses Web 2.0and the his- tory ofWeb Services, shows examples and givesan idea aboutthe principlesof the asynchronous JavaScript and XML technologyby implementing examples and evaluating the concepts. The first chapter will give a short common intro- duction to Web programming. After this introduction details about AJAX are describedin chapter 2. Further detailsandmore technicalaspectsareevaluated in chapter 3. Chapter 4 deals with a practical part, in which also source code examples are implemented. In chapter 5a review isgiven and an outlook for the futureis discussed.

HistoryofWebServices

Similarto almostevery IT- sectoralso in thepartofWeb services andprogram- ming thedevelopment undinnovation of technologiesin thelastfewdecadeswas riseninaverysignificalway.Asin1989 TimBernersLeeinventedtheHyper- textMarkupLanguage,nobodyknewwhatkindof rapiddevelopmentitwould lead to. In the first years HTML was only used for staticwebsites and for layout purposes.But HTMLis still,likenearlytwenty yearsago,alsonowadays(today theXHTML2.0standardiscommon)hierarchicallystructuredandassembled bysocalledtags.ThisisaveryimportantaspectfortheDOM-Document ObjectModel[15], whichwillbeevaluatedlaterindetail.

Themoresiteswebdesignersandwebprogrammersimplemented,themore the demand for dynamic web sites increased. In 1998 first implementations of DynamicHTML were publishedtechnicallyfeasible withJavaScript.These were the first foreriders ofthe newAJAX framework, which generally only uses existing technologies.But not everyone was affected with that hype, one ofthe problems wasthe NetscapeMicrosoft browser war. WhileNetscape invented the JavaScript object based language,Microsoftcounteredwith its Jscript which had similar functionalities but for web programmers there were too many problems withcompatibility. Also nowadaysit is noteasy to createJavaScript applicationscompatiblefor every browser.WhileMicrosoftuses theActive X support in its InternetExplorer,the Gecko browsers (Mozilla, Firefox, etc.) are not completelycompatibletothem.

Laterwebsitesgainedinteractivityanddynamicactionsthrough Javaapplets andFlashapplicationswhichalluse thecommonbrowser-serverrequest:auser opens a browser, sendsarequesttoaserver,theserverhandlestherequestand ofAJAXa lotoftheseconnectionscanbe realizedsimultaneouslywhile theuser is working.

In2000theestablishmentofXMLallowedthedescribingofdata.XML,which isametalanguage,formsthebasisofmanyWebservicesandallowstoexchange datain a standardized way.Italso workswiththeuse of tags,whichcan,in con- trast to HTML, be selfinvented.For the last fewyears more end devices (mobiles, PDAs, etc.) have createda new challenge. Through these developments and evolutionsthenextstep-buildinginteractiveWebapplicationswasnotveryfar away.Suchinteractiveapplicationsarethebasisofthenewgenerationofthe Web- Web2.0.

Web2.0

The latestgeneration ofthe world wide Web is the so called Web 2.0. Through the development andthesuccessof WebServices,informationandseveralproviders like Wikipedia.org or Google thekindof information exchangein theWebis changingandevolving.Interactive Web applications in whichusers canbe impor- tant interactorsor can play partswithin lead to eliminate the border to desktop applications. An example for these developments is the online photo shop and information service ’flickr’. Users can upload theirown photos and give them an XML-tagkeywordtheylike.So othershavethepossibility tosearchforphotos withthesekeywords.Table1showstheevolutionoftheWeb.Anotheraspect of thenew Web2.0generationis thetrend awayfrompersonalWebsitesto blogginginformation.AreasonforthehypeofWeb2.0arethatthebroadband hasbecome mainstream andubiquitous, resultingin anincreasedusage of the Internetforevensmalltasksondifferentdevicesandsomorepeoplegoonline foravarietyoftasksandshopping-relatedactivities.Thetrendsgoto[14]

- A social phenomenonreferring to an approach to creatingand distributing Webcontentitself,characterizedbyopencommunication,decentralization ofauthority, freedom to shareandre-use, and ‖themarket as a conversation‖
- ThetransitionofWebsitesfromisolatedinformationsilostosourcesofcontent and functionality, thus becoming a computing platform serving Web applications toendusers
- Amoreorganizedandcategorizedcontent,withafarmoredevelopeddeep linkingWebarchitecture
- Web2.0isamarketingtermtodifferentiatenewWebbusinessesfromthose of thedotcomboom,whichduetothebustnowseemdiscredited
- The resurgence of excitement around the possibilities of innovativeWeb applicationsandservicesthatgainedalotofmomentumaroundmid2005.

[14,9]

Web1.0	Web2.0
BritannicaOnline	Wikipedia
personalWebsites	blogging
domainnamespeculation	searchengineoptimization
contentmanagementsystems	wikis
publishing	participation
directories(taxonomy)	tagging
screenscraping	Webservices
stickiness	syndication

Table 1: Comparision Web 1.0 - Web 2.0

2 AJAX

Thischapterdescribesthemainconceptsof AJAX-asynchronousJavaScript and XML -framework.The benefits and disadvantages are discussed and some examplesaregiven.

ConceptsofAjax

AJAX - asynchronous JavaScript and XML - is no programming or scriptlanguage, no new invention and no seperate Web service, module or plug-in. In common it is a marketing term for 'Remote Scriptingwith JavaScript, CSS and DOM'. It isan algorithm with 'old'technologies similar to the Dynamic HTML. Ajax allows to createserver connectionsin the background while a user isinteracting with a Web front-end. These connections can be createdasynchronously, which means that theuser need not waituntil theserver replies. They are usually createdasaconsequence ofevents, realized in JavaScriptwhich offerseasy event handling.XML is usedto exchangedatabetweenserverandclient (browser). For the user nocomplete reloading ofthe Website isnecessary. E.g. when a user types an email addressinto an input form, it is possible via AJAXto createa server connectioninthe background,check iftheaddress isvalid ornot and give theinformation backtotheuservia anoutput [8].

Benefits

There are a lot ofadvantages ofthe AJAX technology.No pushing on a submit button and reloadingofa completeWebsiteare needed. Sothe interactivity and the speed for the users are more efficient. A service can be adopted on a persons needandgainmoreinformationif theusersdecidesforacertainstep.AJAX is notageneralsolutionforallWebdevelopment problems,but itcanbe used in a very needful und actualway creating a user friendly application. On the developer's side it ispossible to create database connectionsor script executions duringinteracting withusers.

On the other hand there are also some disadvantages. As AJAX is a very new combinationof oldtechnologies,noonecanbesureif itis onlyamarketing hype now or if it will really be established in some years. So thereare no best practices. There are a lot of applications which use it, but perhaps a better technologycanchallengeit.One big problemis thecompatibility. Thereare some problemswith Microsoft Internet Explorer,which can be avoided with a littlebitofprogramming but some IT-technicians bewareofthat. Anotherpoint is that JavaScriptcan be switchedoffin browsers, because of security reasons. Without thesupport of JavaScript no event handlingandserverconnections on the client side are possible. The nextdisadvantage as regardssurfing comfort is thatnobackbuttoninAJAXapplicationsisavailable(asinGmail).Because ofthe asynchronous generated code the browser has no former page in the cache and cannot reload it exactly. To establish AJAX connections a little bit client-serverWeb knowledge likeparameter passingisneeded(get,post,put,etc.). Some of theseproblemscanbesolvedwithAJAXframeworks,whichhelpdevelopers to createAJAXrequestsandservices more easily [1]. These frameworksare discussedin chapter4.1.

Server/BrowserModel

As alreadymentioned priorAJAXallows multipleserverbrowserconnectionsin the background. It’s possible to create two kinds ofconnections:synchronousand asynchronous.Forinteractivityandactualityitisnecessarytousethesecond, so thebrowserdoesnotwait forareplyoftheserverscript.So theWebsiteneed nottobecompletelyreloaded,thefocusisonthe-fortheuser-informative part.Fig.1andFig.2show thedifferencesbetweentheusualserver-browserand theAJAXserver-browsermodelwherea AJAX-engineis neededtosendand receiverequestsasynchronously.

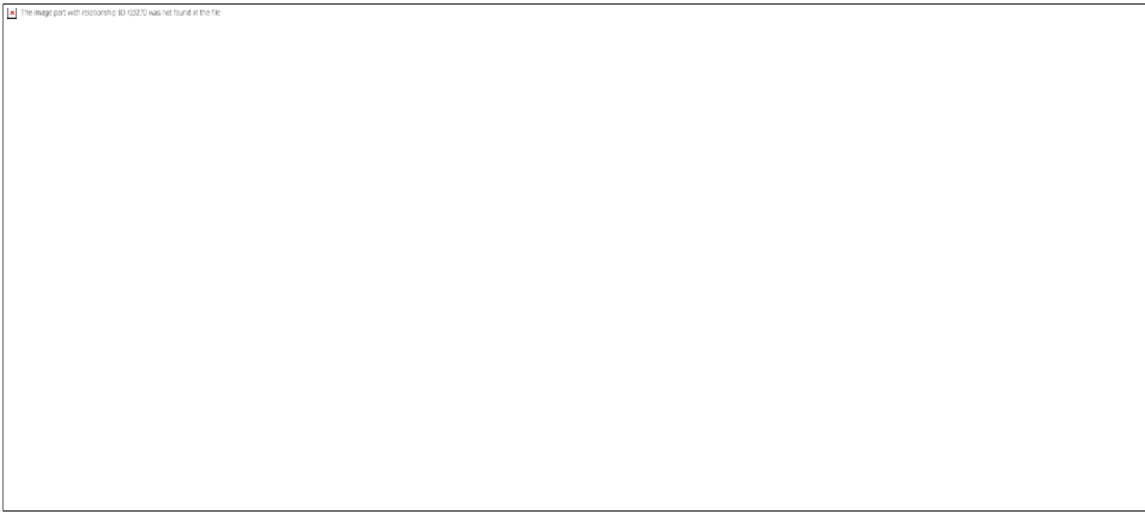
The image part with relationship ID rId269 was not found in the file

Fig.1.CommonServerBrowserRequest

ExistingAjaxTools

There are already several existing AJAX applications. One of the biggest supporters of this technology is Google. In the following part of this paper, some interesting interactive andAJAX-based applications aredescribedandevalu- ated.

Google Suggest[5]-is anextensionof thecommonGoogle Searchengine in Betastate.As soon as lettersare typed into the form, a pull down menu with search resultsand combined keywords pops up. The user isimmediatelyable to seepossible resultsand can choosehisdesired combinationor even combinations he has not known before.



The image part with relationship ID rId270 was not found in the file

Fig.2.AjaxServerBrowserRequest

GoogleMapsandGoogleEarth[6]-isaworldwidemapbuiltupwith satellitepictures.Usershaveeventhepossibilitytoseethreedimensionalviews of landscapesand famous cities. There is also additional information for tourisms places and geographicalvalues. Not the complete information is loaded when the programis started,through AJAXonlytheneededandwantedinformation-chosenbyamouseclickbytheuser-is reloadedandzoomed.Itis evenpossible tocreateroutemapswithGoogleEarth.

Writely [13]isanonlinebrowser-dependent wordprocessingsoftwarewhichis by some people seen as an opponent to Microsoft Word. It is not necessary to installor buy an expensive Officedistribution, you only need a standard browser like the Microsoft Internet Explorer,type in theURLandyouarereadytouse the program.The complete styleisvery similar to Microsoft Word. AJAX han- dles thewritten characters andsavesthemin the background.

Gmail- [4]The Google Mail System is also based on AJAX Services. The complete menu to administrate your received and sent emails is not reloaded if you deleteordisarrangealetter.Arequesttothedatabaseinthebackgroundis sent.

3 TechnicalAspects

Thischapter describesthefunctionsandconceptsofAJAXinmoredetailand in technical aspects. It describes the basics, thedata exchange and manipulation and also the creation of requests.

TechnicalOverview

AJAX is implemented with the client sided JavaScript programming language. JavaScriptprovideseasyeventhandlingandisalmostintegratedinHTML.It canbeusedasanobjectbasedlanguageandallowseasymanipulationofdata withtheDocumentObjectModel(DOM)andCascadingStyleSheets(CSS). Thesetwo technologiessupport theuseof theobjectbasedfeatureof JS.The syntaxis similarto Java buttheuse is completelydifferent. Becauseof compat- ibility reasons and the possibility in browsers to switch it off, JavaScript lacks in popularityintheIT-environment.Butalsoinspiteofitsbadfailurerecogni- tion it is available on every browser and offersvery simple and various techniques whicharenecessarytoknowforallWebprogrammersanddesigners[3].

XmlhttpRequest

TheXmlHTTPRequestis theheart of allAJAXapplications. Itis aJavaScript objectwhichcanusuallybesimplyinstantiated.

```
functioncreateXMLHttpRequest(){
var req =null;
try{req=newActiveXObject(MSXML2.XMLHTTP);}
catch (err_MSXML2){ req = new ActiveXObject(Microsoft.XMLHTTP) ;}
catch (err_Microsoft){
if(typeof XMLHTTPRequest !=undefined)
req =new XMLHTTPRequest;
}
}
returnreq;
}
req.onreadystatechange = handleStateChange;
req.open(GET,http://w3.org/,true);
req.send(anything);
```

After instantiating an XmlHttpRequest the response ofthe server can easily be derivedwithhelp ofthehandlestateChange functionand the readystate variable.

```
functionhandleStateChange(){
switch (req.readystate){
case0://UNINITIALISED
case1://LOADING
case2://LOADED
case3://INTERACTIVE
break;
case4://COMPLETED
handleResponse(req.status,req.responseText);
break;
default:;//failurestate}};
```

With its open method the XmlhttpRequest object offersconnections with http-requests (get,post,put,etc.) and the possibility to choose between synchronous (false-parameter) and asynchronous(true) connections.Some frameworks and librariesoffertheencapsulationoftheXmlHTTPRequestwhicheasetheuse and handling for programmers. Asynchrony allows user to work during code generation.

DatatransformationandExchange

Serversandbrowsershavetocommunicateafteranopenedrequest.Thiscan be done with the readystate variable,which gives information about the current connectionstatus.Butifmoreinformationisneeded-whichisthecommon case-likedatabaseinformationorotherdata,thegeneraldatatransformation andexchangeformatisXML.Logicallyseenyoudonotneedtouseit,e.g.if youonly waitfor a one wordreplyor a parameter (ainserted primarykey value whichwasgeneratedautomaticallyetc.)butwiththeuseof abigamountofdata itmakessense.WiththehelpofXMLitispossibletostructureanddescribe datalogically andscriptlanguagesalso support or even integrate XML - parsers whichtransforms XMLinusabledata.

RepresentationandManipulationofData

The representation and manipulation ofdataare managedin a way ofhandling objects. The Document Object Model (DOM) ofJavascript allows to create objects of a HTML document. These objects are instantiatedhierarchicallystructuredlike the tags in the code. That is why it important to create a well formed HTML document. If not DOM cannot recognize the tag soup. The Document Object Models then creates child nodes and parent nodes which can easily be alteredor appended. Thisis done by supported functions in JavaScript. Exam- ple:

```
<divclass="testclass"id="testid"onclick="change()";>HelloWorld!</div>
<script
language="javascript">function
change{
var testclass = document.getElementbyId("testid");
testclass.innerHTML="New World";
}
</script>
//otherfunctions:
//getelementbytagname(),getelementbyid,haschildnode(),appendchildnode();
```

DOM can be used with Cascading Style Sheets (CSS) to represent data.That allows splitting code and adding events (onmouseover(),onmouseclick(),etc.) to several classes. The Document Object Model is supported by all new browsers where JavaScript is turned on [9,15].

4 PracticalPart

This chapter tries to represent the application levels of AJAX and gives some examples ofhowto useit. For thatreason someframeworks and alsoserver sided AJAXsolutionsare evaluated. Forthe TECproject a draganddropfront end withanAJAXimplementation inthebackground was developed.

FrameworksforAJAX

Frameworkseasetheworkandtheprogrammingof Webapplications.They areaverymighty toolandcanbe veryusefulwhenusingAJAXtechnology. Thefollowing sectionsdealswithsomeimportant AJAXframeworks. SAJAX [12]isa frameworkfor serversidedimplementation ofthe AJAXalgorithmand a possibility to execute server sided functions through the use of browser sided Javascript.It isan interfaceto integrateclientcode on servers and offersmodules for PHP,ASP,Perl,Python andalso Coldfusion.Thereare5 stepswhichhave tobe implemented [8]:

1. integratelibrary–require(sajax.php);
2. definefunctionsonserverscript
3. initialise–sajax-init();
4. exportfunctionstomakethemavailabletoclients–sajax-export(functionname);
5. handleclientrequests–functionsareavailableforclientswithx-functionname

Sarissa[10] isahugeJavaScriptlibrarywhichhelpsprogrammersanddevelopersto easetheuseof AJAX.Althoughthereisnohelpforconnectionsthere areneedfulfunctionswhichextendDOMandXMLHttpRequest.Sarissaisof- ten used with the Prototype librarywhich is very useful in implementing design featuresandvisualeffects.

ATLAS[7]isaMicrosoftsupportedAJAXsolutionwhichiscompatibletoall .NET applications and frameworks.It is usually used with Microsoft's ASP.NET scriptlanguage.

wiki.script.aculo.us [11] is an online JavaScript libraryimplemented by two Austriansand which provides a lot ofuseful informationand encapsulated AJAX functions.Itprovidesalso a lot ofdocumentation andhelpful demos. Thebasics of thesefunctionswherealso usedintheTIGS[2]project.

SourcecodeExample

The TIGS [2] project is an online syndication portalsimilar to a content management system for tourism providers (attractions and institutions) and tourism disposer (hotels or tourism institutions) implementedby the ForschungUrstein together with the Fachhochschule Salzburg. For the selection ofthe providerinformationthetourismdisposershave to use aninteractive draganddropfront end with AJAX technology in the background. It is implemented with a server sided PHP-PEAR engine and the template system SMARTYwhere the HTML andJavaScript code is generated using themodel-view-controller architecture. Theimportantpartsofthesourcecodecanbefoundintheappendix.

Alternatives

Remotescriptingcan alsobeimplementedwithothertechnologies.Forexample a simple Inline Frame could also be able to reload server - sided dynamic generatedWebsites.But the disadvantage ofiframes are security and compatibility reasons. Iframes areneither dynamically producible. Other alternativesare Live-Connectfunctionality ofFlashapplications or JAVA implementations, but these technologies are manufacturer specificand it’s difficult to implement extensions compared to AJAX. Another very new alternative is the DOM 3.0load save specification which alsoallows several browser -server connectionsin the background. But the disadvantageisthatit’s not yet compatiblein all browsers, only Opera supports it. AJAX combines the advantages ofthese alternatives without havingtheirdisadvantages.

5 ReviewandSummary

WebdevelopersandprogrammershavetodecideifAJAXwillreallybea commonusedstandardinthenextdecade.Criticssaythereisnothingnew with AJAX, it’s only a marketing term for old dynamic HTML combined with JavaScript.Throughthecommonuseofbroadbandinternetconnectionsthe moreserverconnectionsinthebackgroundwon’tbethatslow,othercriticism issecurityandcompatibilityaspectsbecauseofMicrosoft’sActiveXstandards. ItisnotpossibletouseAJAXineveryproblemorwitheveryWebsitebutit is a useful way togaininteractivity anditis easy to learnandextend.NoteverythingcanbedonewithAJAXbutitisanewchallengeandastepfurther fordisappearingtheborderbetweendesktopapplicationsandWebapplications andservices.Withthehelpof frameworksAJAXisaverypowerfultechnique, nomatterifit’sonlyatermforwellestablishedtechnologies.

References

1. ChristianWenz.Ajax.Entwicklerpress(2006),2006.
2. ForschungUrstein. TourimusInfoGate (TEC/TIGS). http://www.tourimus-info-gate.at,2006.
3. GerhildMaier. AjaxvonAbisX.http://krottmaier.cgv.tugraz.at/docs/seminar/sem2005_ajax.pdf, year= 2006.
4. Google.GoogleGmail.http://gmail.google.com,year=2005.
5. Google.GoogleSuggest.http://www.google.com/webhp?hl=de,year=2006.
6. Google.GoogleMaps.http://maps.google.com,2006.
7. Microsoft. ATLAS. http://www.microsoft.com/germany/msdn/library/web/AJAXUndASPNET.mspx?mfr=true, year=2006.
8. Olaf Bergmann,Carsten Bormann. Ajax-Web2.0.SPCTeijaVerlag(16.11.2005), 2005.
9. PaulMiller.Web2.0:BuildingtheNewLibrary.http://www.ariadne.ac.uk/issue45/miller/, year= 2006.
10. Sarissa.JavascriptLibrary.http://swik.net/Sarissa/,year=2006.
11. Scriptacolous.script.aculo.us.
12. SimpleAjaxToolkit.SAJAX.http://www.modernmethod.com/sajax/,year= 2006.
13. The WebWord Processor. Writely. http://www2.writely.com/info/WritelyOverflowWelcome.htm, year= 2006.
14. TimO’Reilly.WhatisWeb2.0?http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html (23.05.2006),2006.
15. W3C Community. DOM 3.0. http://www.w3.org/TR/2004/NOTE-DOM-Level-3-XPath-20040226/,year=2006.

A Appendix

Fileeditabo.php(controller):

```
<?include"intro.php";
   check_auth();
$db->setFetchMode(DB_FETCHMODE_OBJECT,"paket");
  $sql="SELECTdistinct*FROMpaket
  LEFTJOINuserUSING(USERID)ORDERBYKURZTITEL";
    $result=&$db->query($sql);
    $pakete =array();
while($result->fetchInto($paket)){
$pakete[]=$paket;
}
$smarty->assign(’pakete’,$pakete);
$smarty->assign(’titel’,"NeuesAboerstellen");
$smarty->assign(’view’,"editabo");
$smarty->display(’index.tpl’);
?>
```

Fileeditabo.tpl(view):

```
<scriptsrc="scriptaculous/lib/prototype.js"type="text/javascript"></script>
<scriptsrc="scriptaculous/src/scriptaculous.js"type="text/javascript"></script>
<scriptsrc="scriptaculous/src/dragdrop.js"type="text/javascript"></script>

<divalign="center"><strong>Warenkorb:</strong></div><br/>
<divid="korb"class="korb">
<palign="center">ZiehenSiehiermittelsDrag andDrop Ihregew&uuml;nschten Pakete
 hinein</p></div>
<divid="anzahl"align="center">0</div><divalign="center">Paketegesamt</div>
<divid="loeschen"class="loeschen">
<palign="center">Ziehen Sie Pakete aus dem Warenkorb hier rein, umsie zu
l&ouml;schen<p></div>

<divid="alle-pakete">
{foreachfrom=$paketeitem=paket}
<divid="{$paket->PAKETID}"class="paket">
<spanclass="zeile">{$paket->NAME}</span>
<spanclass="zeile"><b>{$paket->KURZTITEL}</b></span>

<script type="text/javascript"
 language="javascript">new Draggable(’{$paket->PAKETID}’
 ,{revert:true});
    </script>
  {/foreach}</script>
</div>
```

```
<scripttype="text/javascript"
    14language="javascript">
      Droppables.add('loeschen', {
    accept: 'waren',
     onDrop:function(element){
      //Layoutchange

  var handlerFunc =function(t) {
      aboid.innerHTML = t.responseText;
}

  varerrFunc=function(t){
     alert('Error'+t.status+'--'+t.statusText);
}

newAjax.Request('pakettoabo.php',
    {parameters:'delete=true&paketid='+element.id+'&aboid='+abo_id,
    onSuccess:handlerFunc,
 }})

Droppables.add('korb', {
   accept: 'paket',
  onDrop:function(element){

//Layoutchange

varerrFunc=function(t){
     alert('Error'+t.status+'--'+t.statusText);
}

newAjax.Request('pakettoabo.php',
{parameters:'paketid='+element.id+'&aboid='+abo
  _id, onSuccess:handlerFunc,
  onFailure:errFunc});
    }})</script>
```