UNIT VI

APPLETS: Applet Basics, Applet Life Cycle, A Simple Applet, HTML applet tag, Applet Parameters.

APPLET

Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.

Advantage of Applet

- ✓ There are many advantages of applet. They are as follows:

It works at client side so less response time.

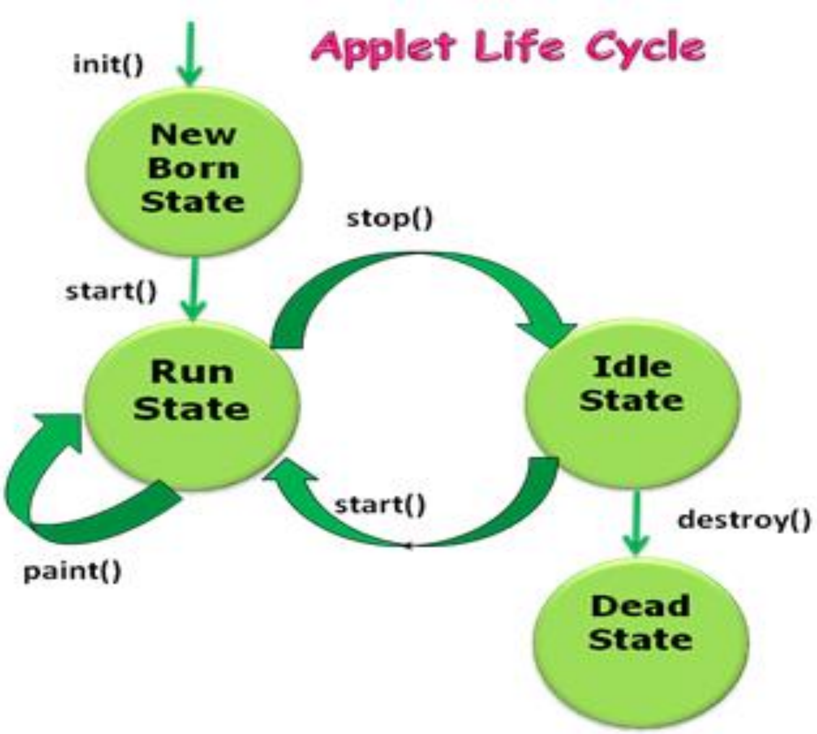
- ✓ Secured
- ✓ It can be executed by browsers running under many plateforms, including Linux, Windows, Mac Os etc.

Difference between applet and application

Applet	**Application**
Small Program	Large Program
Used to run a program on client Browser	Can be executed on stand alone computer system
Applet is portable and can be executed by any JAVA supported browser.	Need JDK, JRE, JVM installed on client machine.
Applet applications are executed in a Restricted Environment	Application can access all the resources of the computer
Applets are created by extending the java.applet.Applet	Applications are created by writing public static void main(String[] s) method.
Applet application has 5 methods which will be automatically invoked on occurance of specific event	Application has a single start point which is main method

Example:
import java.awt.*; import java.applet.*; public class Myclass extends Applet {
public void init() { } public void start() { } public void stop() {} public void destroy() {} public void paint(Graphics g) {} }
	public class MyClass { public static void main(String args[]) {} }

<u>Life cycle of Applet</u>



init: This method is intended for whatever initialization is needed for your applet. It is called after the param tags inside the applet tag have been processed.

start: This method is automatically called after the browser calls the init method. It is also called whenever the user returns to the page containing the applet after having gone off to other pages.

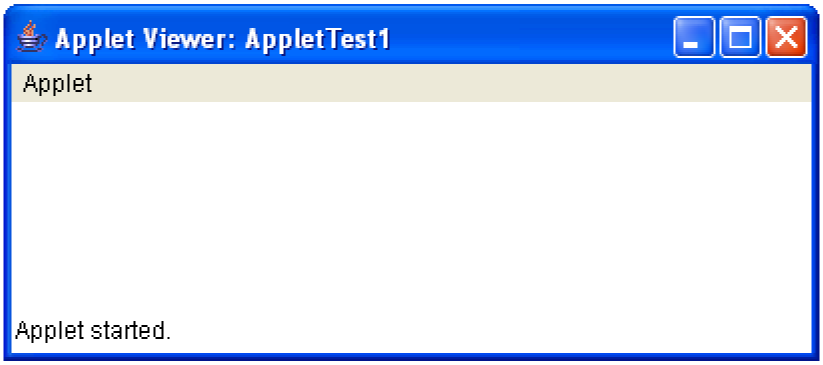
stop: This method is automatically called when the user moves off the page on which the applet sits. It can, therefore, be called repeatedly in the same applet.

destroy: This method is only called when the browser shuts down normally. Because applets are meant to live on an HTML page, you should not normally leave resources behind after a user leaves the page that contains the applet.

paint: Invoked immediately after the start() method, and also any time the applet needs to repaint itself in the browser. The paint() method is actually inherited from the java.awt.

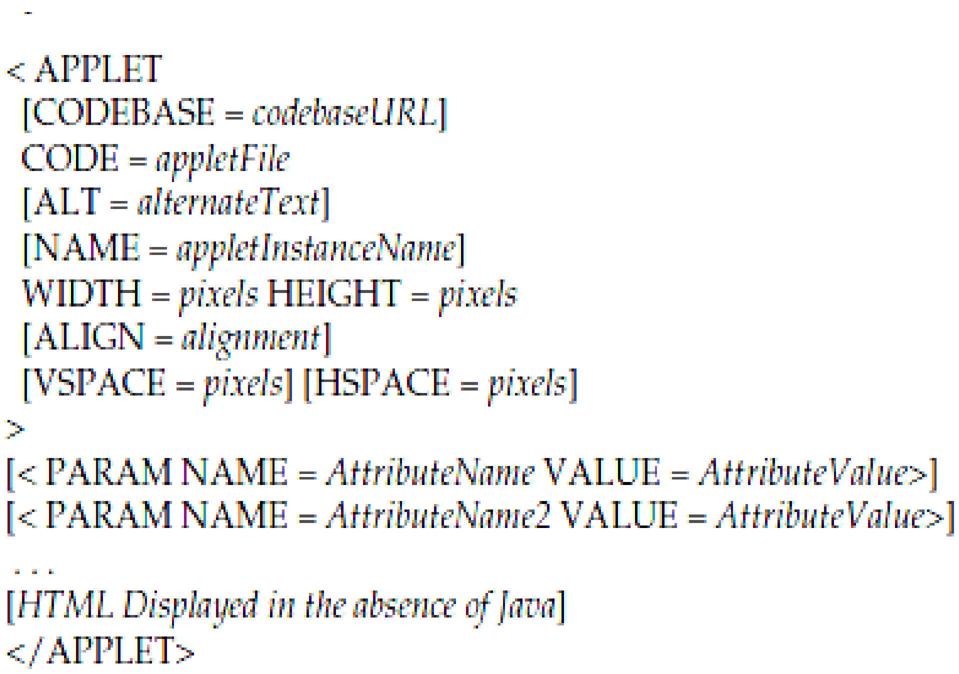
```
import java.awt.*;
import java.applet.*;

/* <applet code="AppletTest1" height=400 width=400>
</applet> */
public class AppletTest1 extends Applet
{
 public void init()
 {
  //initialization
 }
 public void start ()
 {
  //start or resume execution
 }
 public void stop()
 {
  //suspend execution
 }
 public void destroy()
 {
  //perform shutdown activity
 }
 public void paint (Graphics g)
 {
  //display the content of window
 }
}
```

Output:



```
Applet started.
```

The HTML APPLET Tag



```
< APPLET
 [CODEBASE = codebaseURL]
 CODE = appletFile
 [ALT = alternateText]
 [NAME = appletInstanceName]
 WIDTH = pixels HEIGHT = pixels
 [ALIGN = alignment]
 [VSPACE = pixels] [HSPACE = pixels]
>
[< PARAM NAME = AttributeName VALUE = AttributeValue>]
[< PARAM NAME = AttributeName2 VALUE = AttributeValue>]
 . . .
[HTML Displayed in the absence of Java]
</APPLET>
```

CODEBASE CODEBASE is an optional attribute that specifies the base URL of the applet code, which is the directory that will be searched for the applet's executable class file (specified by the CODE tag). The HTML document's URL directory is used as the CODEBASE if this attribute is not specified. The CODEBASE does not have to be on the host from which the HTML document was read.

CODE CODE is a required attribute that gives the name of the file containing your applet's compiled **.class file. This file is relative to the code base URL of the applet,** which is the directory that the HTML file was in or the directory indicated by CODEBASE if set.

ALT The ALT tag is an optional attribute used to specify a short text message that should be displayed if the browser understands the APPLET tag but can’t currently run Java applets. This is distinct from the alternate HTML you provide for browsers that don’t support applets.

NAME NAME is an optional attribute used to specify a name for the applet instance. Applets must be named in order for other applets on the same page to find them by name and communicate with them. To obtain an applet by name, use **getApplet( ),** which is defined by the **AppletContext interface.**

WIDTH AND HEIGHT WIDTH and HEIGHT are required attributes that give the size (in pixels) of the applet display area.

ALIGN ALIGN is an optional attribute that specifies the alignment of the applet.
This attribute is treated the same as the HTML IMG tag with these possible values: LEFT, RIGHT, TOP, BOTTOM, MIDDLE, BASELINE, TEXTTOP, ABSMIDDLE, and ABSBOTTOM.

VSPACE AND HSPACE These attributes are optional. *VSPACE specifies the space,*in pixels, above and below the applet. HSPACE specifies the space, in pixels, on each side of the applet. They’re treated the same as the IMG tag’s VSPACE and HSPACE attributes.

PARAM NAME AND VALUE The PARAM tag allows you to specify appletspecific arguments in an HTML page. Applets access their attributes with the **getParameter( ) method.**

Example:

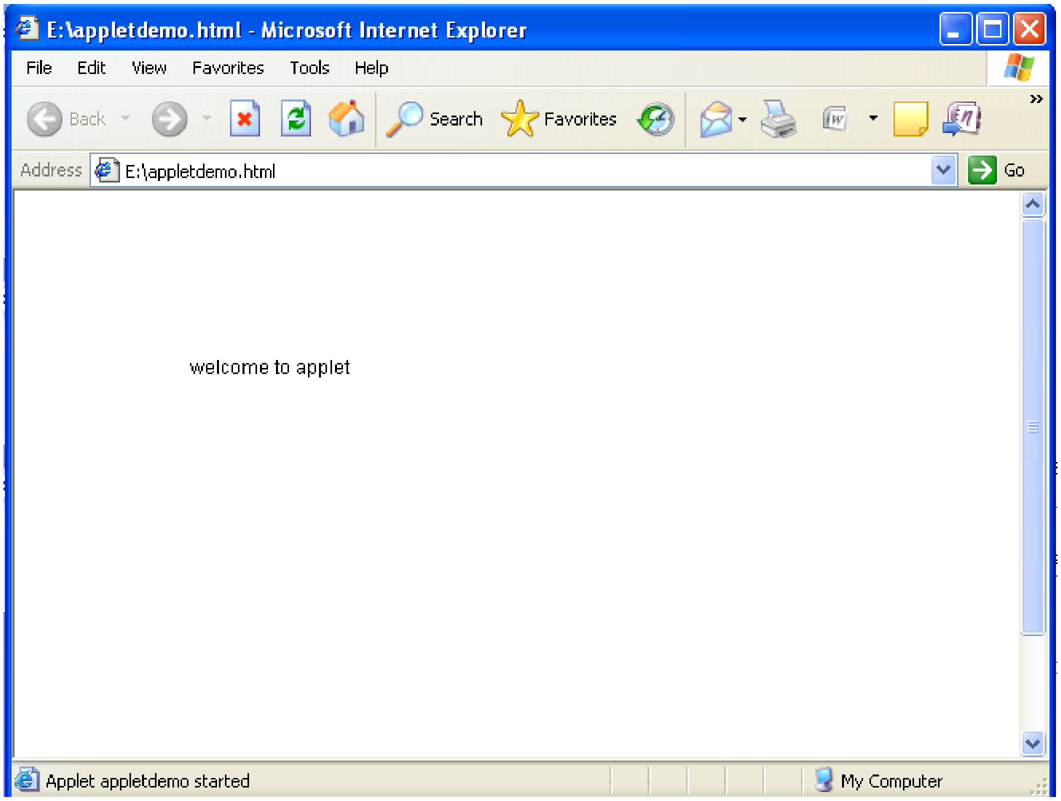
appletdemo.java

```
import java.applet.Applet;
import java.awt.Graphics;


public class appletdemo extends Applet{

public void paint(Graphics g){
g.drawString("welcome to applet",150,150);
}

}
```

Demo.html

```
<html>
<applet code="appletdemo" width=400 height=400>
</applet>
</html>
```

Output:



```
welcome to applet
```

Applet appletdemo started