Applets

- A small program written in Java and included in a HTML page.
- It is independent of the operating system on which it runs
- An applet is a Panel that allows interaction with a Java program
- A applet is typically embedded in a Web page and can be run from a browser
- You need special HTML in the Web page to tell the browser about the applet
- For security reasons, applets run in a sandbox: they have no access to the client's file system

Applets Support

- Most modern browsers support Java 1.4 if they have the appropriate plugin
- Sun provides an application *appletviewer* to view applets without using browser.
- In general you should try to write applets that can be run with any browser

What an Applet is?

- You write an applet by extending the class Appletor JApplet
- Applet is just a class like any other; you can even use it in applications if you want
- When you write an applet, you are only writing part of a program
- The browser supplies the main method

The genealogy of Applet

The following figure shows the inheritance hierarchy of the JApplet class. This hierarchy determines much of what an applet can do and how.

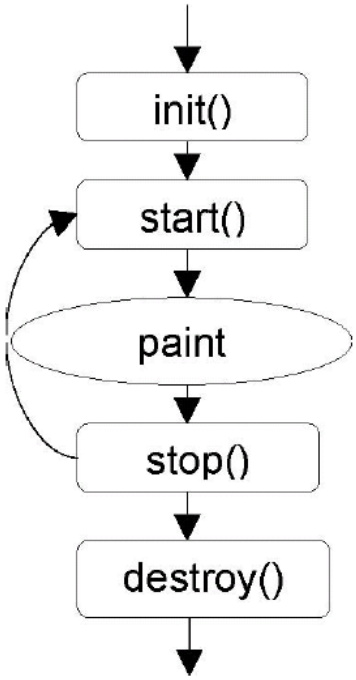
java.lang.Object
|
+----java.awt.Component
|
+----java.awt.Container|+----java.awt.Panel
|
+----java.applet.Applet|+----javax.swing.JApplet

Applet Life Cycle Methods

When an applet is loaded, an instance of the applet's controlling class (an Applet subclass) is created. After that an applet passes through some stages or methods, each of them are build for specific purpose
An applet can react to major events in the following ways:

- It can **initialize** itself.
- It can **start** running.
- It can **stop** running.
- It can perform a **final cleanup**, in preparation for being unloaded

The applet's life cycle methods are called in the specific order shown below. Not every applet needs to Override every one of these methods.



Let's take a look on each method in detail and find out what they do

init( )

- Is called only once.
- The purpose of init( ) is to initialize the applet each time it's loaded (or reloaded).
- You can think of it as a constructor

start( )

- To start the applet's execution
- For example, when the applet's loaded or when the user revisits a page that contains the applet
- start( ) is also called whenever the browser is maximized

paint( )

- paint( ) is called for the first time when the applet becomes visible
- Whenever applet needs to be repainted, paint( ) is called again
- Do all your painting in paint( ), or in a method that is called from paint( )

stop( )

- To stop the applet's execution, such as when the user leaves the applet's page or quits the browser.
- stop( ) is also called whenever the browser is minimized

destroy( )

- Is called only once.
- To perform a **final cleanup** in preparation for unloading

Example # 01 (Basic Example of creating a simple applet)

Step 1: Create a class that extends JApplet
Step 2: Create an HTML file and embed applet in HTML file to run

HelloApplet.java

```
//importing required packages
import java.awt.*; import javax.swing.*;

// extending class from JApplet so that our class also becomes an applet
public class HelloApplet extends JApplet {

// overriding paint method
public void paint(Graphics g) {
// write code here u want to display & draw by using Graphics object
g.drawString("Hello World", 30 , 30);
}
} // end class
```

```
Test.html - Notepad
File  Edit  Format  View  Help
<html>
<head><title> Simple Applet</title></head>
<body>
        <applet code="HelloApplet.class"
width=150 height=100></applet>
</body>
</html>
```

Compile & Execute

By simply double clicking on *Test.html* file, you can view the applet in your browser. However, you can also use the appletviewer java program for executing or running applets.

The applet viewer is invoked from the command line by the command

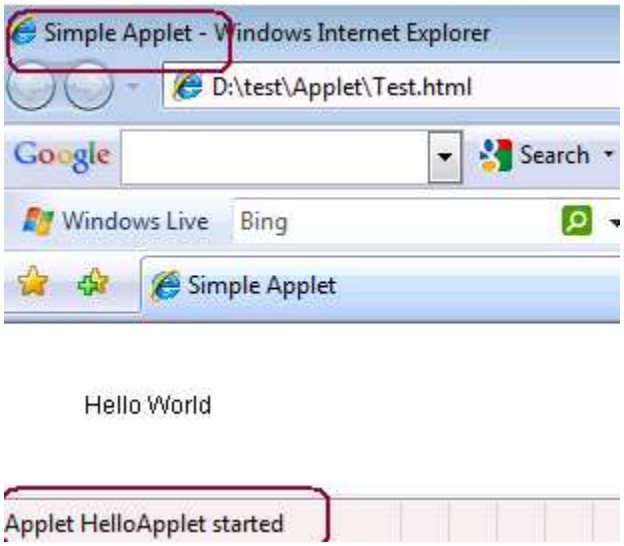
appletviewer *htmlfile*

where htmlfile is the name of the file that contains the html document. For our example, the command looks like this:

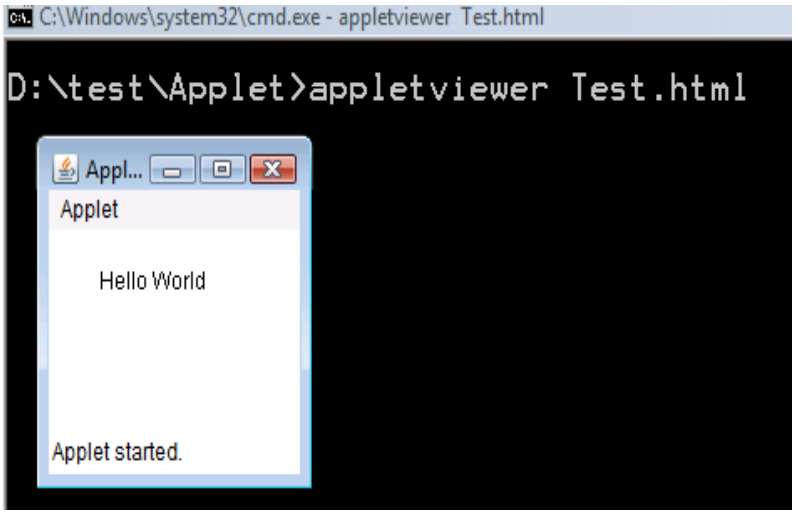
appletviewer *Test.html*

As a result, you will see the following output

Run in HTML



Compile on command prompt:



Example # 02 (Creating applet with Timer class)

Step 1: Create a class that extends JApplet
Step 2: Create an HTML file and embed applet in HTML file to run

JavaAnim.java

```
import java.awt.*; import java.awt.event.*; import javax.swing.*;

public class JavaAnim extends JApplet implements ActionListener {
// used to count how many times paint is called
int clearCounter;
// declaring Timer reference
Timer t;

// overriding init method, used to initialize variables
public void init() {
     clearCounter = 0;
     Timer t = new Timer(1000, this);
     t.start();
}

// overriding paint method discussed above
public void paint(Graphics g) {
     clearCounter++;
     Graphics2D g2 = (Graphics2D) g;
     if (clearCounter == 10) {
          g2.clearRect(0, 0, 1000, 700);
          clearCounter = 0;
     }
     for (int i = 1; i <= 40; i++) {
          Color c = chooseColor();
          g2.setColor(c);
          Font f = chooseFont();
          g2.setFont(f);
          drawTxt(g2);
     }
}
// overriding actionPerformed()of ActionListener interface called by Timer object
   public void actionPerformed(ActionEvent ae) {
        repaint();
   }

   // chooseColor method discussed above
   public Color chooseColor() {
        int r = (int) (Math.random() * 255);
        int g = (int) (Math.random() * 255);
        int b = (int) (Math.random() * 255);
        Color c = new Color(r, g, b);
        return c;
   }

   // chooseFont method discussed above
   public Font chooseFont() {
        int fontChoice = (int) (Math.random() * 4) + 1;
        Font f = null;

        switch (fontChoice) {
        case 1: f = new Font("Serif", Font.BOLD + Font.ITALIC, 20);
                break;
        case 2: f = new Font("SansSerif", Font.PLAIN, 17);
                break;
        case 3: f = new Font("Monospaced", Font.ITALIC, 23);
                break;
        case 4: f = new Font("Dialog", Font.ITALIC, 30);
                break;
        }
        return f;
   }

   // drawTxt() method discussed above
   public void drawTxt(Graphics2D g2) {
        int x = (int) (Math.random() * 1000);
        int y = (int) (Math.random() * 700);
        g2.drawString("BSEF07", x, y);
   }

   } // end class
```

AppView.html - Notepad

File Edit Format View Help

```
<html>
<head><title> Animated Java Word </title></head>
<body>

<applet code="JavaAnim.class" width=1000
height=700> </applet>

</body></html>
```

Compile on command prompt:

Run in HTML