Java Applet Basics

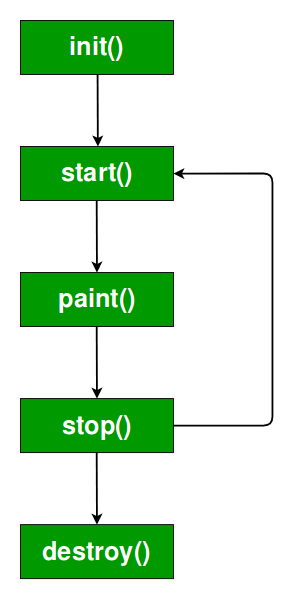
Applet is a Java program that can be embedded into a web page. It runs inside the web browser and works at client side. Applet is embedded in a HTML page using the APPLET or OBJECT tag and hosted on a web server.

Applets are used to make the web site more dynamic and entertaining.

**Some important points :**

1. All applets are sub-classes (either directly or indirectly) of *[java.applet.Applet](https://docs.oracle.com/javase/7/docs/api/java/applet/Applet.html" \t "_blank)* class.
2. Applets are not stand-alone programs. Instead, they run within either a web browser or an applet viewer. JDK provides a standard applet viewer tool called applet viewer.
3. In general, execution of an applet does not begin at main() method.
4. Output of an applet window is not performed by *System.out.println()*. Rather it is handled with various AWT methods, such as *drawString()*.

**Life cycle of an applet :**



It is important to understand the order in which the various methods shown in the above image are called. When an applet begins, the following methods are called, in this sequence:

1.init()  
2.start()  
3. paint( )

When an applet is terminated, the following sequence of method calls takes place:  
1.stop()  
2.destroy()  
Let’s look more closely at these methods.

1. **init( ) :**The **init( )** method is the first method to be called. This is where you should initialize variables. This method is called **only once** during the run time of your applet.
2. **start( ) :**The **start( )** method is called after **init( )**. It is also called to restart an applet after it has been stopped. Note that **init( )**is called once i.e. when the first time an applet is loaded whereas **start( )** is called each time an applet’s HTML document is displayed onscreen. So, if a user leaves a web page and comes back, the applet resumes execution at **start( )**.
3. **paint( ) :**The **paint( )** method is called each time an AWT-based applet’s output must be redrawn. This situation can occur for several reasons. For example, the window in which the applet is running may be overwritten by another window and then uncovered. Or the applet window may be minimized and then restored.

**paint( )** is also called when the applet begins execution. Whatever the cause, whenever the applet must redraw its output, **paint( )**is called.

The **paint( )** method has one parameter of type [Graphics](https://docs.oracle.com/javase/7/docs/api/java/awt/Graphics.html). This parameter will contain the graphics context, which describes the graphics environment in which the applet is running. This context is used whenever output to the applet is required.

1. **stop( ) :**The **stop( )** method is called when a web browser leaves the HTML document containing the applet—when it goes to another page, for example. When **stop( )** is called, the applet is probably running. You should use **stop( )** to suspend threads that don’t need to run when the applet is not visible. You can restart them when **start( )** is called if the user returns to the page.
2. **destroy( ) :** The **destroy( )** method is called when the environment determines that your applet needs to be removed completely from memory. At this point, you should free up any resources the applet may be using. The **stop( )** method is always called before **destroy( )**.

**Creating Hello World applet :**

Let’s begin with the HelloWorld applet :

|  |
| --- |
| // A Hello World Applet  // Save file as HelloWorld.java    import java.applet.Applet;  import java.awt.Graphics;    // HelloWorld class extends Applet  public class HelloWorld extends Applet  {      // Overriding paint() method      @Override      public void paint(Graphics g)      {          g.drawString("Hello World", 20, 20);      }    } |

**Explanation :**

1. The above java program begins with two import statements. The first import statement imports the Applet class from applet package. Every AWT-based(Abstract Window Toolkit) applet that you create must be a subclass (either directly or indirectly) of Applet class. The second statement import the [Graphics](https://docs.oracle.com/javase/7/docs/api/java/awt/Graphics.html) class from awt package.
2. The next line in the program declares the class HelloWorld. This class must be declared as public, because it will be accessed by code that is outside the program. Inside HelloWorld, **paint( )** is declared. This method is defined by the AWT and must be overridden by the applet.
3. Inside **paint( )** is a call to *drawString( )*, which is a member of the [Graphics](https://docs.oracle.com/javase/7/docs/api/java/awt/Graphics.html) class. This method outputs a string beginning at the specified X,Y location. It has the following general form:
4. void drawString(String message, int x, int y)

Here, message is the string to be output beginning at x,y. In a Java window, the upper-left corner is location 0,0. The call to *drawString( )*in the applet causes the message “Hello World” to be displayed beginning at location 20,20.

Notice that the applet does not have a **main( )** method. Unlike Java programs, applets do not begin execution at **main( )**. In fact, most applets don’t even have a **main( )** method. Instead, an applet begins execution when the name of its class is passed to an applet viewer or to a network browser.

**Running**

**Using java enabled web browser :**To execute an applet in a web browser we have to write a short HTML text file that contains a tag that loads the applet. We can use APPLET or OBJECT tag for this purpose. Using APPLET, here is the HTML file that executes HelloWorld :

1. <applet code="HelloWorld" width=200 height=60>
2. </applet>

The width and height statements specify the dimensions of the display area used by the applet. The APPLET tag contains several other options. After you create this html file, you can use it to execute the applet.



1. **appletviewer with java source file :**If you include a comment at the head of your Java source code file that contains the APPLET tag then your code is documented with a prototype of the necessary HTML statements, and you can run your compiled applet merely by starting the applet viewer with your Java source code file. If you use this method, the HelloWorld source file looks like this :

|  |
| --- |
|  |
|  |
|  |
| **Drawing Right-angled Rectangles**  Java comes with two methods to draw right-angled rectangles.  Supporting methods in Graphics class   * **void drawRect(int x, int y, int width, int height):** Draws an outline rectangle with the left-top coordinates of **x** and **y** and with the **width** and **height** specified. * **void fillRect(int x, int y, int width, int height):** Draws a solid rectangle with the left-top coordinates of **x** and **y** and with the **width**and **height** specified.   Following program draws two right-angled rectangles – **outline** and **solid** in applet window.  **Applet file name: RectanglesDrawing.java**     |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12 | | import java.awt.\*;  import java.applet.\*;  public class RectanglesDrawing extends Applet  {    public void paint(Graphics g)    {      g.setColor(Color.blue);      g.drawRect(50, 80, 150, 100);      g.setColor(Color.magenta);      g.fillRect(50, 80, 150, 100);    }  } | | 1  2 | <applet code="RectanglesDrawing" width="400" height="300">  </applet> | |   [Java Draw Rectangle Applet graphics](http://way2java.com/wp-content/uploads/2011/06/ss.png)  *g.setColor(Color.blue); g.drawRect(50, 80, 150, 100);*  We know, the **setColor()** method of Graphics class sets the drawing color. In the above statement, the color is set to **blue** (one of the 13 predefined colors) and this color will be effective until changed again.  **drawRect()** method takes 4 parameters. 50 and 80 are the **x** and **y** positions that represents the top-left coordinates of the rectangle. 150 and 100 are the **width** and **height**of the rectangle. The method draws an outline rectangle in blue color.  *g.setColor(Color.magenta); g.fillRect(230, 80, 150, 100);*  **fillRect()**method draws a solid rectangle filled with **magenta** (one of the [13 predefined colors](http://way2java.com/awt-graphics/java-color/)) color. 230, 80 are**x**, **y** coordinates and 150, 100 are **width** and **height** of the rectangle  **Drawing Square**  *g.drawRect(50, 80, 150, 150);*  The above statement draws a square. Observe, the width and height are same of 150 pixels. |

**Features of Applets over HTML**

* + Displaying dynamic web pages of a web application.
  + Playing sound files.
  + Displaying documents
  + Playing animations