

Applet vs. Application

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.

An **applet** is a program written in the Java programming language that can be included in an HTML page, much in the same way an image is included.

An **application** is a standalone Java program that runs as a true application, outside of a browser. Both require a JVM (Java Virtual Machine).

Lifecycle of Java Applet

1. Applet is initialized.
2. Applet is started.
3. Applet is painted.
4. Applet is stopped.
5. Applet is destroyed.

Lifecycle methods for Applet

For creating any applet **java.applet.Applet** class must be inherited. It provides 4 life cycle methods of applet.

1. public void **init()**: is used to initialize the Applet. It is invoked only once.
2. public void **start()**: is invoked after the init() method or browser is maximized. It is used to start the Applet.
3. public void **stop()**: is used to stop the Applet. It is invoked when Applet is stop or browser is minimized.
4. public void **destroy()**: is used to destroy the Applet. It is invoked only once.

java.awt.Component class

1. public void **paint(Graphics g)**: is used to paint the Applet. It provides Graphics class object that can be used for drawing oval, rectangle, arc etc.

There are two ways to **run an applet**

1. **By html file.** (this support is not available in html5)

First.html

```
<html>
  <body>
    <applet code="First.class" width="300" height="300"></applet>
  </body>
</html>
```

First is a public class that extends java.applet.Applet

2. **By appletViewer tool** (for testing purpose).

c:\>javac First.java

c:\>appletviewer First.html

java.awt. Graphics class:

1. **public abstract void drawLine(int x1, int y1, int x2, int y2):** is used to draw line between the points(x1, y1) and (x2, y2).
2. **public void drawRect(int x, int y, int width, int height):** draws a rectangle with the specified width and height.
3. **public abstract void fillRect(int x, int y, int width, int height):** is used to fill rectangle with the default color and specified width and height.
4. **public abstract void drawOval(int x, int y, int width, int height):** is used to draw oval with the specified width and height.
5. **public abstract void fillOval(int x, int y, int width, int height):** is used to fill oval with the default color and specified width and height.
6. **public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used draw a circular or elliptical arc.
7. **public abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used to fill a circular or elliptical arc.
8. **public abstract void drawString(String str, int x, int y):** is used to draw the specified string.
9. **public abstract void setColor(Color c):** is used to set the graphics current color to the specified color.
10. **public abstract void setFont(Font font):** is used to set the graphics current font to the specified font.

Note:

java.awt.Font

[Font\(String name, int style, int size\)](#)

String constants:

Font.DIALOG

Font.DIALOG_INPUT

Font.MONOSPACED

Font.SANS_SERIF

Font.SERIF

int constants:

Font.BOLD

Font.ITALIC

Font.PLAIN

E.g.

```
g.setColor(Color.red);
```

```
g.setFont(new Font("Gujrati Saral-1", Font.PLAIN, 20));
```

```
g.drawString("gujratI", 50, 50);
```

11. **public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):**

is used draw the specified image.

```
Image img = this.getImage(this.getCodeBase(),"../images/image1.jpg");
g.drawImage(img, 50, 100, this);
```

```
public Image getImage(URL u, String image)
```

public URL getDocumentBase():the URL of the document in which this applet is embedded.

public URL getCodeBase():the URL of the directory which contains this applet

The **Component** class implements **ImageObserver** interface. So, current class object would also be treated as ImageObserver because Applet class indirectly extends the Component class.

java.lang.**Object**

java.awt.**Component** (A component is an object having a graphical representation that can be displayed on the screen and that can interact with the user. Examples of components are the buttons, checkboxes, and scrollbars of a typical graphical user interface.)

java.awt.**Container** (A generic Abstract Window Toolkit (AWT) container object is a component that can contain other AWT components.)

java.awt.**Panel** (Panel is the simplest container class. A panel provides space in which an application can attach any other component, including other panels. The default layout manager for a panel is the FlowLayout layout manager.)

java.applet.**Applet**

Applet Parameter

We can get any information from the HTML file as a parameter. For this purpose, Applet class provides a method named **getParameter()**.

```
import java.applet.Applet;
import java.awt.Graphics;
public class UseParam extends Applet{
    public void paint(Graphics g){
        String str=getParameter("msg");
        g.drawString(str,50, 50);
    }
}
<html>
    <body>
        <applet code="UseParam.class" width="300" height="300">
            <param name="msg" value="Welcome to applet">
        </applet>
    </body>
</html>
```