**Java 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.

1. It works at client side so less response time.
2. Secured
3. It can be executed by browsers running under many platforms, including Linux, Windows, Mac Os etc.

**Drawback of Applet**

Plugin is required at client browser to execute applet.

**Hierarchy of Applet**

# hierarchy of applet

**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:**

The java.applet.Applet class 4 life cycle methods and java.awt.Component class provides 1 life cycle methods for an applet.

**java.applet.Applet class**

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 initialized 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**

The Component class provides 1 life cycle method of applet.

**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.

**Who is responsible to manage the life cycle of an applet?**

Java Plug-in software.

**How to run an Applet?**

There are two ways to run an applet

1. By html file.
2. By appletViewer tool (for testing purpose).

**1 . Simple example of Applet by html file:**

To execute the applet by html file, create an applet and compile it. After that create an html file and place the applet code in html file. Now click the html file.

1. //First.java
2. import java.applet.Applet;
3. import java.awt.Graphics;
4. public class First extends Applet
5. {
6. public void paint(Graphics g){
7. g.drawString("welcome",150,150);
8. }
9. }

**Note: class must be public because its object is created by Java Plugin software that resides on the browser.**

**myapplet.html**

1. <html>
2. <body>
3. <applet code="**First.class**" width="300" height="300">
4. </applet>
5. </body>
6. </html>

**2.Simple example of Applet by appletviewer tool:**

To execute the applet by appletviewer tool, create an applet that contains applet tag in comment and compile it. After that run it by: appletviewer First.java. Now Html file is not required but it is for testing purpose only.

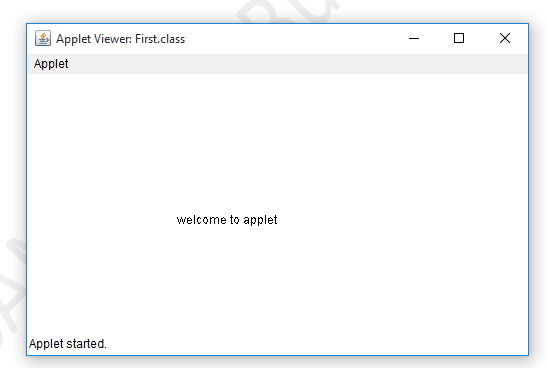
//First.java

1. import java.applet.Applet;
2. import java.awt.Graphics;
3. public class First extends Applet
4. {
5. public void paint(Graphics g)
6. {
7. g.drawString("welcome to applet",150,150);
8. }
9. }
10. /\* <applet code="First.class" width="300" height="300">
11. </applet>
12. \*/

**To execute the applet by appletviewer tool, write in command prompt:**

c:\>javac First.java

c:\>**appletviewer** First.java



**Displaying Graphics in Applet**

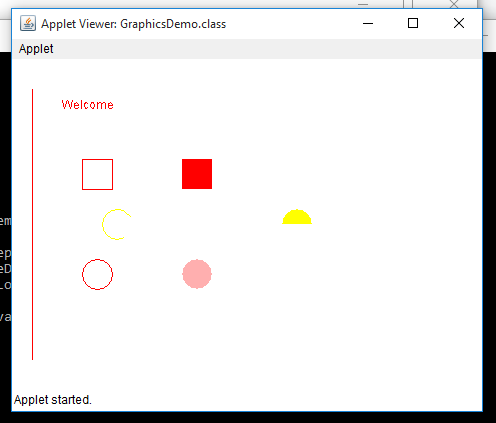
java.awt.Graphics class provides many methods for graphics programming.

**Commonly used methods of Graphics class:**

1. **public abstract void drawString(String str, int x, int y):** is used to draw the specified string.
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 drawLine(int x1, int y1, int x2, int y2):** is used to draw line between the points(x1, y1) and (x2, y2).
7. **public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):** is used draw the specified image.
8. **public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used draw a circular or elliptical arc.
9. **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.
10. **public abstract void setColor(Color c):** is used to set the graphics current color to the specified color.
11. **public abstract void setFont(Font font):** is used to set the graphics current font to the specified font.

**Example of Graphics in applet:**

1. import java.applet.Applet;
2. import java.awt.\*;
3. public class GraphicsDemo extends Applet
4. {
5. public void paint(Graphics g)
6. {
7. g.setColor(Color.red);
8. g.drawString("Welcome",50, 50);
9. g.drawLine(20,30,20,300);
10. g.drawRect(70,100,30,30);
11. g.fillRect(170,100,30,30);
12. g.drawOval(70,200,30,30);
13. g.setColor(Color.pink);
14. g.fillOval(170,200,30,30);
15. g.drawArc(90,150,30,30,30,270);
16. g.fillArc(270,150,30,30,0,180);
17. }
18. }
19. /\* <applet code="GraphicsDemo.class" width="300" height="300">   </applet>  \*/

****

**Displaying Image in Applet**

Applet is mostly used in games and animation. For this purpose image is required to be displayed. The java.awt.Graphics class provide a method drawImage() to display the image.

**Syntax of drawImage() method:**

**public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):** is used draw the specified image.

## **How to get the object of Image:**

**The java.applet.Applet class provides getImage() method that returns the object of Image.**

1. public Image getImage(URL u, String image){}
2. **Other required methods of Applet class to display image:**
3. **public URL getDocumentBase():** is used to return the URL of the document in which applet is embedded.
4. **public URL getCodeBase():** is used to return the base URL.

**APPLET PROGRAM TO DISPLAY IMAGE**

**import java.awt.\*;**

**import java.applet.\*;**

**/\*<applet code="DisplayImage.class" width="300" height="300">**

**</applet> \*/**

**public class DisplayImage extends Applet**

**{**

**Image picture;**

**public void init()**

**{**

**picture = getImage(getDocumentBase(),"image.jpg");**

**}**

**public void paint(Graphics g)**

**{**

**g.drawImage(picture, 30,30, this);**

**}**

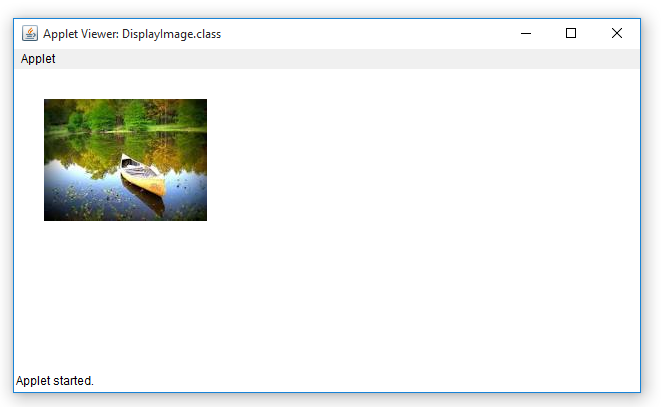
**}**

In the above example, drawImage() method of Graphics class is used to display the image. The 4th argument of drawImage() method of is ImageObserver object. The Component class implements ImageObserver interface. So current class object would also be treated as ImageObserver because Applet class indirectly extends the Component class.

**OUTPUT:**

**E:\java\AWT\applet>javac DisplayImage.java**

**E:\java\AWT\applet>appletviewer DisplayImage.java**

****

**Animation in Applet**

Applet is mostly used in games and animation. For this purpose image is required to be moved.

import java.awt.\*;

import java.applet.\*;

/\*<applet code="AnimationExample.class" width="300" height="300">

</applet> \*/

public class AnimationExample extends Applet

{

Image picture;

public void init()

{

picture =getImage(getDocumentBase(),"gif1.gif");

}

public void paint(Graphics g)

{

for(int i=0;i<500;i++)

{

g.drawImage(picture, i,30, this);

try{ Thread.sleep(100); }

catch(Exception e){}

}

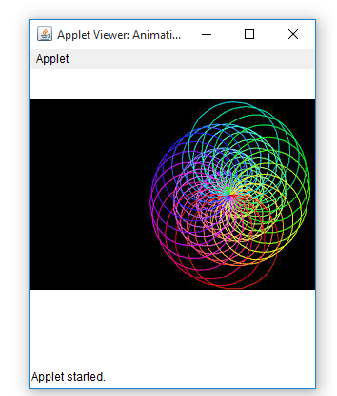
}

}

**OUTPUT:**

**E:\java\AWT\applet>javac AnimationExample.java**

**E:\java\AWT\applet>appletviewer AnimationExample.java**

****

**EventHandling in Applet**

As we perform event handling in AWT or Swing, we can perform it in applet also. Let's see the simple example of event handling in applet that prints a message by click on the button.

//EventApplet.java

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

/\*<applet code="EventApplet.class" width="300" height="300">

</applet> \*/

public class EventApplet extends Applet implements ActionListener{

Button b;

TextField tf;

public void init()

{

tf=new TextField();

tf.setBounds(30,40,150,20);

b=new Button("Click");

b.setBounds(80,150,60,50);

add(b);add(tf);

b.addActionListener(this);

setLayout(null);

}

public void actionPerformed(ActionEvent e)

{

tf.setText("Welcome");

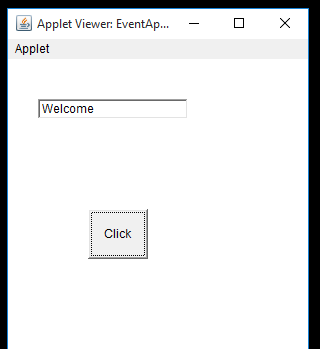
}

}

**OUTPUT**

E:\java\AWT\applet>javac EventApplet.java

E:\java\AWT\applet>appletviewer EventApplet.java

****

## **Playing Audio**

An applet can play an audio file represented by the AudioClip interface in the java.applet package. The AudioClip interface has three methods, including −

* **public void play()** − Plays the audio clip one time, from the beginning.
* **public void loop()** − Causes the audio clip to replay continually.
* **public void stop()** − Stops playing the audio clip.

To obtain an AudioClip object, you must invoke the getAudioClip() method of the Applet 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 play the audio clip.

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

/\*<applet code="AudioEX.class" height="400" width="300">

</applet>\*/

public class AudioEX extends Applet implements ActionListener

{

Button b1,b2;

AudioClip ac;

public void init() {

b1=new Button("Play");

add(b1);

b1.addActionListener(this);

b2=new Button(" Stop ");

add(b2);

b2.addActionListener(this);

ac= getAudioClip(getCodeBase(), "audio.mp3");

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==b1)

{

ac.play();

System.out.println("play");

}

if(ae.getSource()==b2)

{

ac.stop();

System.out.println("stop");

}

}

}

**OUTPUT:**

