APPLETS

Nihar Ranjan Roy

WHY WERE APPLETS REMOVED?

1. **Security Issues** – Applets required the Java Plugin, which had multiple security vulnerabilities.
2. **Browser Support Dropped** – Major browsers (Chrome, Firefox, Edge) stopped supporting NPAPI plugins, which were required for applets.
3. **Better Alternatives** – Technologies like **JavaScript, HTML5, and WebAssembly** replaced the need for applets.
4. **Deprecated in JDK 9** – Marked for removal.
5. **Removed in JDK 11** – Applet API and Java Plugin were officially removed.

ALTERNATIVES TO APPLETS

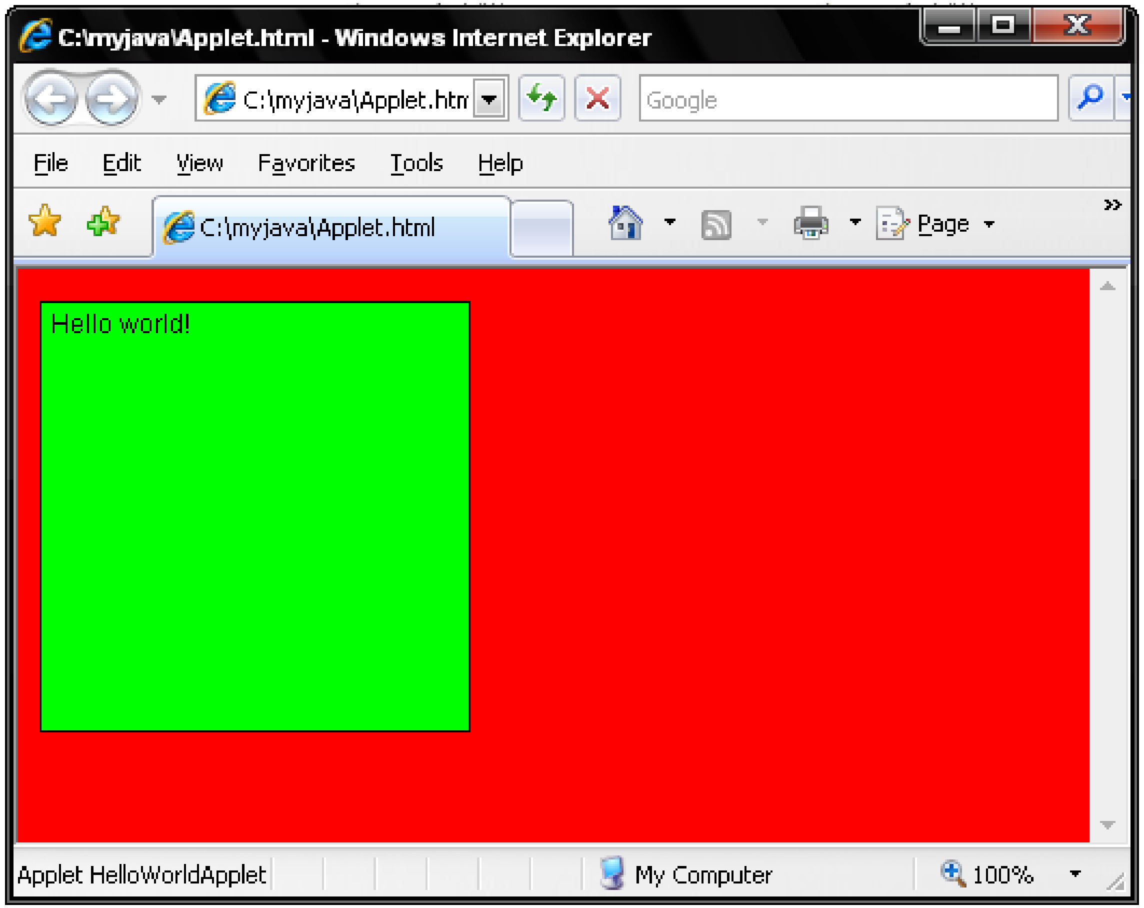
- **JavaFX** – Used for GUI applications.
- **Spring Boot/Web Apps** – For web-based applications.
- **JavaScript & HTML5** – Modern interactive web applications.

WHAT IS AN APPLET?

An applet is a special kind of Java program that a **browser enabled with Java technology** can download from the internet and run

- An applet is typically embedded inside a web-page and runs in the context of the browser.
- An applet must be a subclass of the java.applet.Applet class, which provides the standard interface between the applet and the browser environment.

EXAMPLE



Hello world!

Applet HelloWorldApplet

SIMPLE APPLET

```
import java.applet.*;
import java.awt.*;
public class HelloWorldApplet extends Applet
{
}
}
```

SIMPLE APPLET

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

public class HelloWorldApplet extends Applet {
public void paint(Graphics g)
{
g.drawRect(0, 0, getSize().width – 1, getSize().height – 1);
        g.drawString("Hello world!", 5, 15);
        setBackground(Color.green);
    }
}
```

HTML FILE FORMAT

```
<html>
<head>
</head>
<body bgcolor=red>
<applet code=HelloWorldApplet.class width=200
  height=200></applet>
</body>
</html>
```

you can run your applet using the lightweight appletviewer application that comes with the JDK. For example, if AppletWorld.html is the html file name, then you run the command as

appletviewer AppletWorld.html

QUICK RUN WITHOUT HTML FILE

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

public class HelloWorldApplet extends Applet {
    public void paint(Graphics g) {
  g.drawRect(0, 0,
        getSize().width - 1,
        getSize().height - 1);
        g.drawString("Hello world!", 5, 15);
        setBackground(Color.green);
    }
}
```

```
/*<applet code=HelloWorldApplet width=200 height=200>
  </applet>*/
```

APPLET TAG

```
<applet
code
codebase
alt
name
width
height
align
vspace
hspace>
<param  name=backcolor value=red>
……. .
</applet>
```

APPLET TAG

Attribute	Value	Description
align	Left, right top, bottom	Defines the text alignment around the applet
alt	text	An alternate text to be displayed if the browser support applets but cannot run this applet
archive	URL	A URL to the applet when it is stored in a Java Archive or ZIP file
code	URL	A URL that points to the class of the applet
codebase	URL	Indicates the base URL of the applet if the code attribute is relative
hspace	pixels	Defines the horizontal spacing around the applet
name	unique_nam e	Defines a unique name for the applet (to use in scripts)
vspace	pixels	Defines the vertical spacing around the applet

EMBED AND OBJECT TAGS

Embed and **object** tags are the other tags which can be used to embed an applet in a html page

APPLET LIFE CYCLE

Life Cycle of an Applet: Basically, there are four methods in the Applet class on which any applet is built.

- **`init`**: This method is intended for whatever initialization is needed for your applet. It is called after the param attributes of the applet tag.
- **`start:`** This method is automatically called after init method. It is also called whenever user returns to the page containing the applet after visiting other pages.
- **`stop`**: This method is automatically called whenever the user moves away from the page containing applets. You can use this method to stop an animation.
- **`destroy:`** This method is only called when the browser shuts down normally.

```
/*<applet code=appletTest widht=200
  height=200></applet>*/
public class appletTest extends Applet
{
String str="";
  public void init()
  {
  str+="init-->";
  repaint();
  showStatus(str);
  }
public void start()
{
str+="start-->";
repaint();
showStatus(str);
}
```

```
public void paint(Graphics g)
{
str+="paint-->";
g.drawString(str,20,20);
showStatus(str);
}
public void stop()
{
str+="stop-->";
repaint();
showStatus(str);
}

public void destroy()
{
str+="destroy-->";
repaint();
showStatus(str);
}
}
```

PAINT,REPAINT, UPDATE METHODS

- paint➔each time applet is redraw
- repaint➔entire window is repainted
- update➔only a portion is redrawn
    - » This method can be used to avoid flickering effect

BACKGROUND & FOREGROUND

- setBackground(Color.XXX)➔ sets applets background color
- setForeground(Color.XXX)➔sets applets foreground color

Example➔ **`Color.red,Color.green,Color.pink`** etc

SHOW SHORT MESSAGES

`showStatus(str)`

Displays information in applets status bar of the browser of applet viewer

➔useful for debugging information

APPLETS RESTRICTIONS

Current browsers impose the following restrictions on any applet that is loaded over the network:

- An applet cannot load libraries or define native methods.
- It cannot ordinarily read or write files on the host that's executing it.
- It cannot make network connections except to the host that it came from.
- It cannot start any program on the host that's executing it.
- It cannot read certain system properties.

SecurityManager detects a violation

LOADING DATA FILES

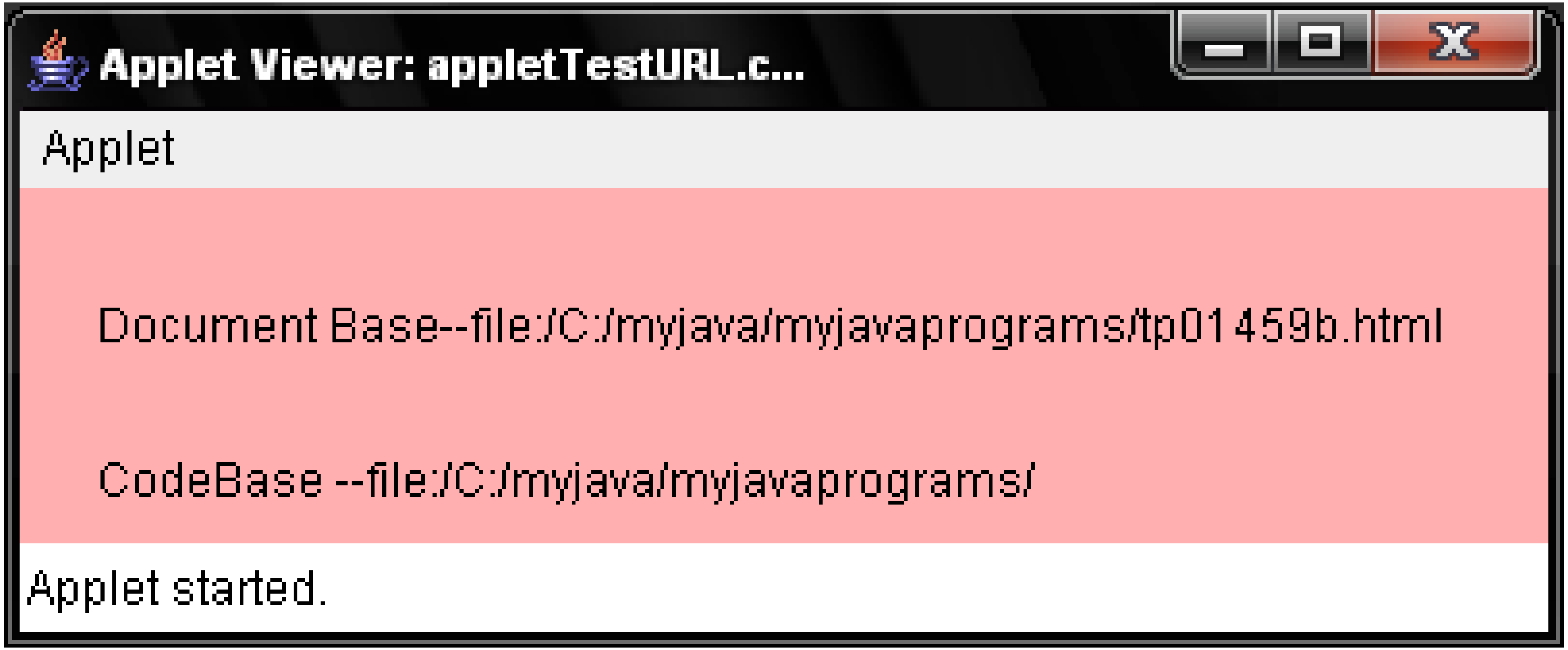
Whenever an applet needs to load some data from a file that's specified with a relative URL (a URL that doesn't completely specify the file's location), the applet usually uses either the code base or the document base to form the complete URL.

- **The code base➔**, returned by the JApplet getCodeBase method, is a URL that specifies the directory from which the applet's classes were loaded.
- **The document base➔**, returned by the JApplet getDocumentBase method, specifies the directory of the HTML page that contains the applet.
- Unless the <APPLET> tag specifies a code base, both the code base and document base refer to the same directory on the same server.

RETRIEVING CODE BASE & DOC BASE

```
import java.applet.*;
import java.awt.*;
/*<applet code=appletTestURL  widht=200 height=200></applet>*/
public class appletTestURL extends Applet
{
public void init()
{setBackground(Color.pink);
//setForeground(Color.red);
  }
public void paint(Graphics g)
{
g.drawString("Document Base--"+getDocumentBase().toString(),20,40);
g.drawString("CodeBase --"+getCodeBase().toString(),20,80);
}
}
```

OUTPUT



```
Document Base--file:/C:/myjava/myjavaprograms/tp01459b.html

CodeBase --file:/C:/myjava/myjavaprograms/
```

Applet started.

DISPLAYING DOCUMENTS IN THE BROWSER

With the **AppletContext showDocument methods**, an applet can tell the browser which URL to show and in which browser window.

```
public void showDocument(java.net.URL url)
public void showDocument(java.net.URL url, String
  targetWindow)
```

TARGET WINDOWS

"_blank" Display the document in a new, nameless window.

"windowName" Display the document in a window named *windowName*. This window is created if necessary.

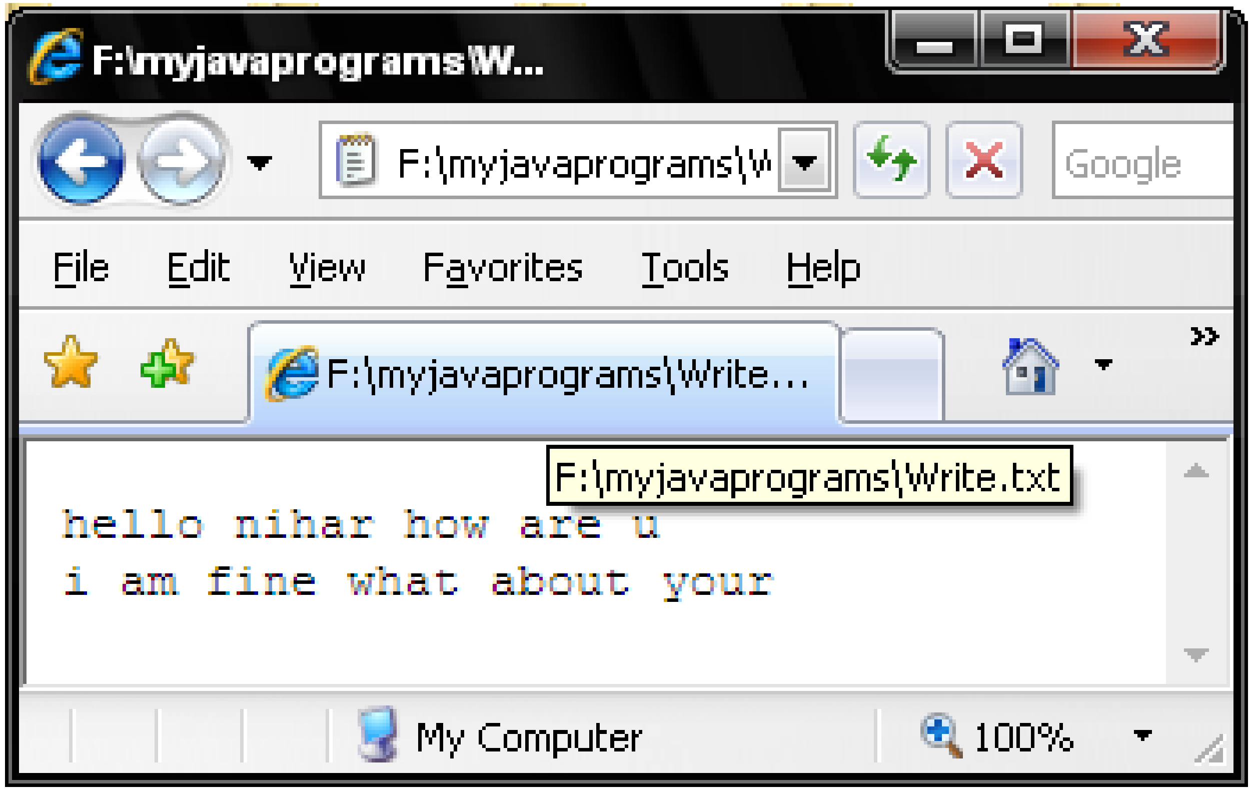
"_self" Display the document in the window and frame that contain the applet.

"_parent" Display the document in parent frame of the applet's frame. If the applet frame has no parent frame, this acts the same as "_self".

"_top" Display the document in the top-level frame. If the applet's frame is the top-level frame, this acts the same as "_self".

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

/*<applet code=appletContext   width=200 height=200></applet>*/
public class appletContext extends Applet
{

public void start()
{ setBackground(Color.pink);
   AppletContext ac=getAppletContext();
   URL url=getCodeBase();

try{
   ac.showDocument(new URL(url+"Write.txt"));
   }catch(MalformedURLException me){}
   }
}
```

OUTPUT

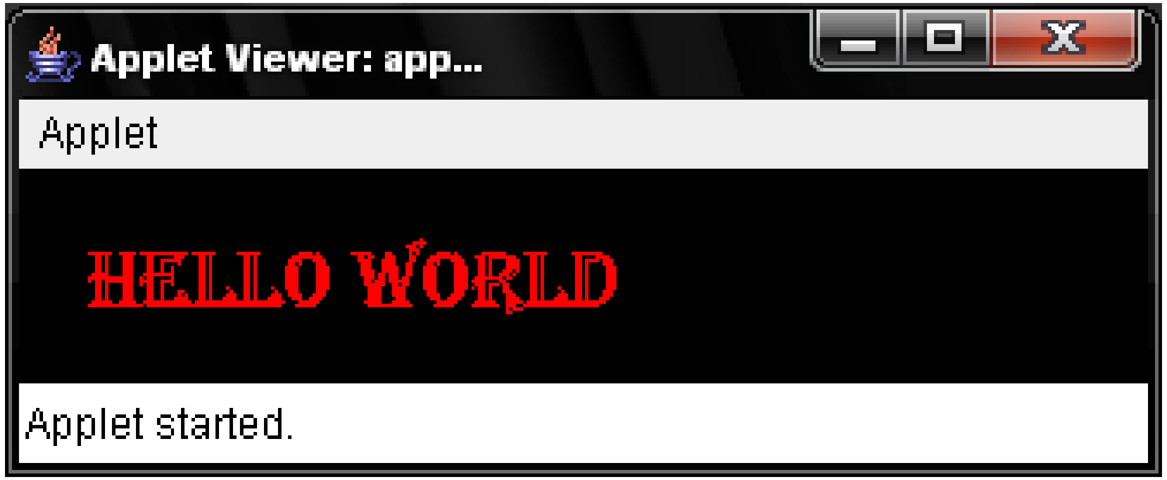


F:\myjavaprograms\W...
F:\myjavaprograms\W
Google
File Edit View Favorites Tools Help
F:\myjavaprograms\Write...
F:\myjavaprograms\Write.txt
My Computer
100%

```
hello nihar how are u
i am fine what about your
```

CHANGING THE FONT OF AN APPLET

```
Font f=new Font("algerian",Font.PLAIN,24);
        g.setFont(f);
```



HELLO WORLD

Applet started.

Font to be changed in which method?

```
public class appletTestFont extends Applet
{

public void init()
{
   setBackground(Color.black);
   }

public void paint(Graphics g)
{
   Font f=new Font("algerian",Font.PLAIN,24);
   g.setFont(f);
g.setColor(Color.red);
g.drawString("Hello World",20,40);

}
}
```

SOUND-RELATED METHODS

getAudioClip(URL),

getAudioClip(URL, String)

- Return an object that implements the AudioClip interface.

play(URL),

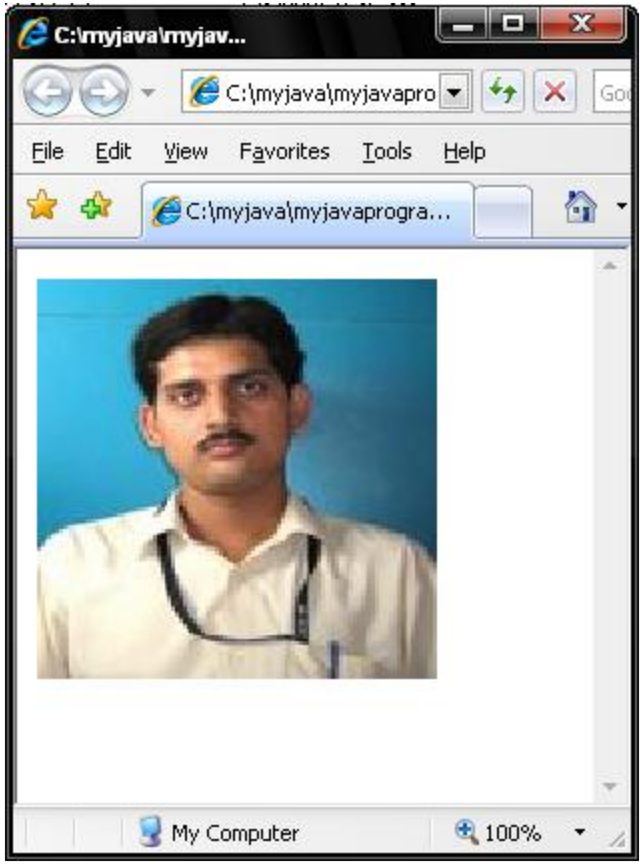
play(URL, String)

- Play the AudioClip corresponding to the specified URL. The AudioClip interface defines the following methods:
- loop ➔Starts playing the clip repeatedly.
- Play➔ Plays the clip once.
- Stop➔ Stops the clip. Works with both looping and one-time sounds.

```
public class appletAudio extends Applet
{
            AudioClip au=null;

public void init()
{
            AppletContext ac=getAppletContext();
            URL url=getCodeBase();
            try{
             au=ac.getAudioClip(new URL(url+"spacemusic.au"));
            }catch(MalformedURLException me){}
}

public void start()
{
//au.start();
au.loop();
}
            public void stop()
            {
            au.stop();
            }
}
```

LOADING IMAGE

```
public class appletImage extends Applet
{    Image img;
     URL u;
     int w,h;

public void init()
{    u=getCodeBase();
     img=getImage(u,"nihar.jpg");
     w=getWidth();
     h=getHeight();
     }
               public void paint(Graphics g)
               {
               g.drawImage(img,0,0,w,h,this);
               }
}
```



PASSING PARAMETER

```
<applet.....>
<param name=par1 value=val1>
<param name=par2 value=val2>
<param name=par3 value=val3>
</applet>
```

```
String str=getParmeter(par_name);
```

Problem

Make an applet in which the background color of the applet is passed as a parameter to the applet

APPLET INTERFACES

- AppletContext
- AppletStub
- AudioClip

GRAPHICS

- Drawing lines➔

```
g.drawLine(int x1,int y1,int x2,int y2);
```

- Drawing rectangles

```
g.drawRect(int x,int y,int w, int h);
g.fillRect(int x,int y,int w, int h);
```

- Change brush color

```
setColor(Color.XXX())
```

➔drawing Ovals

- **g.drawOval(int x, int y, int w, int h);**
- **g.fillOval(int x, int y, int w, int h);**

➔drawing Arcs

- **drawArc(int x,int y, int w, int h, int start_angle,int end_angle);**

➔drawing Polygons

- **drawPoly(int x[],int y[],int length)**
- **fillPoly(int x[],int y[],int length)**

➔drawing Polyline

- **drawPolyLine(int x[],int y[],int length)**

APPLET VS APPLICATION

Applets	Applications
1. Init method	
2. Java enable web browser	
3.GUI based	
4.Restrictions	
5.Embeded into HTML files	