

## Simple applet program:

```
import java.applet.*;
import java.awt.*;
/*<applet code="SimpleApplet.class" width=200 height=60></applet>*/
public class SimpleApplet extends Applet
{
    public void paint(Graphics g) {
        g.drawString("A simple Applet",20,20);
    }
}
```

**Save as SimpleApplet.java**

**Compile the file and run as**

**appletviewer SimpleApplet.java**

## HTML file:

```
<html>
<applet code="SimpleApplet" width=200 height=60></applet>
</html>
```

Save as 123.html

## An Applet skeleton.

```
import java.awt.*;
import java.applet.*;
/*
<applet code="AppletSkel" width=300 height=100>
</applet>
*/
public class AppletSkel extends Applet {
    // Called first.
    public void init() {
        // initialization
    }
    /* Called second, after init(). Also called whenever
```

```
the applet is restarted. */
public void start() {
// start or resume execution
}
// Called when the applet is stopped.
public void stop() {
// suspends execution
}
/* Called when applet is terminated. This is the last
method executed. */
public void destroy() {
// perform shutdown activities
}
// Called when an applet's window must be restored.
public void paint(Graphics g) {
// redisplay contents of window
}
}
```

### **Sample program:**

```
import java.awt.*;
import java.applet.*;
/* <applet code="Sample" width=200 height=100>
</applet>
*/
public class Sample extends Applet
{
String msg;
public void init()
{
setBackground(Color.cyan);
setForeground(Color.red);
msg = "Inside init()-";
}
```

```
}  
public void start()  
{  
    msg += "Inside start()-";  
}  
public void paint(Graphics g)  
{  
    msg += "Inside paint()-";  
    g.drawString(msg,10,30);  
    showStatus("This is shown at status");  
}  
}
```

### **Familiarization of CodeBase and DocumentBase methods:**

```
import java.awt.*;  
import java.applet.*;  
import java.net.*;  
/* <applet code="Bases" width=300 height=50>  
</applet>  
*/  
public class Bases extends Applet  
{  
    public void paint(Graphics g)  
    {  
        String msg;  
        URL url= getCodeBase();  
        msg= "Code Base:" +url.toString();  
        g.drawString(msg,10,20);  
        url= getDocumentBase();  
        msg= "Document Base:" +url.toString();  
        g.drawString(msg,10,40);  
    }  
}
```

## Passing parameters to applets:

```
import java.awt.*;
import java.applet.*;
/* <applet code="ParamDemo" width=300 height= 80>
<param name=fontName value=Courier>
<param name=fontSize value=14>
<param name=leading value = 2>
<param name=accountEnabled value= true>
</applet>
*/
public class ParamDemo extends Applet
{
String fontName;
int fontSize;
float leading;
boolean active;
public void start()
{
String param;
fontName=getParameter("fontName");
if(fontName==null)
fontName= "Not Found";
param=getParameter("fontSize");
try
{
if(param!=null)
fontSize=Integer.parseInt(param);
else
fontSize=0;
}
catch(NumberFormatException e)
{
```

```

fontSize=-1;
}
param=getParameter("leading");
try
{
if(param!=null)
leading=Float.parseFloat(param);
else
leading=0;
}
catch(NumberFormatException e)
{
leading=0;
}
param=getParameter("accountEnabled");
if (param!=null)
active=Boolean.parseBoolean(param);
}
public void paint(Graphics g)
{
g.drawString("Font Name." + fontName,0,10);
g.drawString("Font Size." + fontSize,0,26);
g.drawString("Leading." + leading,0,42);
g.drawString("Account Active." + active,0,58);
}
}

```

## Rotating a string in an applet:

```

import java.awt.*;
import java.applet.*;
/*
<applet code="SimpleBanner" width=300 height=50>
</applet>

```

```
*/  
  
public class SimpleBanner extends Applet implements Runnable {  
    String msg = " Hello";  
    Thread t;  
    int state;  
    boolean stopFlag;  
    // Set colors and initialize thread.  
    public void init() {  
        t=null;  
        setBackground(Color.cyan);  
        setForeground(Color.red);  
    }  
    // Start thread  
    public void start() {  
        t = new Thread(this);  
        stopFlag = false;  
        t.start();  
    }  
    // Entry point for the thread that runs the banner.  
    public void run() {  
        char ch;  
        // Display banner  
        for( ; ; ) {  
            try {  
                repaint();  
                Thread.sleep(250);  
                ch = msg.charAt(0);  
                msg = msg.substring(1, msg.length());  
                msg += ch;  
                if(stopFlag)  
                    break;  
            } catch (InterruptedException e) { }  
        }  
    }  
}
```

```
}  
// Pause the banner.  
public void stop() {  
    stopFlag = true;  
    t = null;  
}  
// Display the banner.  
public void paint(Graphics g) {  
    g.drawString(msg, 50, 30);  
}  
}
```