Problem Description

How to create a basic Applet?

Solution

Following example demonstrates how to create a basic Applet by extending Applet Class. You will need to embed another HTML code to run this program.

import java.applet.\*;

import java.awt.\*;

public class Main extends Applet {

public void paint(Graphics g) {

g.drawString("Welcome in Java Applet.",40,20);

}

}

Now compile the above code and call the generated class in your HTML code as follows −

<HTML>

<HEAD>

</HEAD>

<BODY>

<div >

<APPLET CODE = "Main.class" WIDTH = "800" HEIGHT = "500"></APPLET>

</div>

</BODY>

</HTML>

Result

The above code sample will produce the following result in a java enabled web browser.

Welcome in Java Applet.

[**next →**](http://www.javatpoint.com/Graphics-in-applet)[**← prev**](http://www.javatpoint.com/LayoutManager-BorderLayout)

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. They are as follows:

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

Drawback of Applet

* Plugin is required at client browser to execute applet.

Problem Description

How to create a banner using Applet?

Solution

Following example demonstrates how to play a sound using an applet image using Thread class. It also uses drawRect(), fillRect(), drawString() methods of Graphics class.

import java.awt.\*;

import java.applet.\*;

public class SampleBanner extends Applet implements Runnable {

String str = "This is a simple Banner ";

Thread t ;

boolean b;

public void init() {

setBackground(Color.gray);

setForeground(Color.yellow);

}

public void start() {

t = new Thread(this);

b = false;

t.start();

}

public void run () {

char ch;

for( ; ; ) {

try {

repaint();

Thread.sleep(250);

ch = str.charAt(0);

str = str.substring(1, str.length());

str = str + ch;

}

catch(InterruptedException e) {}

}

}

public void paint(Graphics g) {

g.drawRect(1,1,300,150);

g.setColor(Color.yellow);

g.fillRect(1,1,300,150);

g.setColor(Color.red);

g.drawString(str, 1, 150);

}

}

Result

The above code sample will produce the following result in a java enabled web browser.

View in Browser.

The following is an another sample example to create a banner using Applet.

import java.awt.\*;

import java.applet.\*;

public class NewApplet extends Applet implements Runnable {

String msg = " It is a moving Banner. ";

char cha;

boolean stopFlag = true;

Thread t = null;

public void start() {

t = new Thread(this);

stopFlag = false;

t.start();

}

public void run() {

for(;;) {

try {

repaint();

Thread.sleep(250);

cha = msg.charAt(0);

msg = msg.substring(1,msg.length());

msg = msg + cha;

if(stopFlag) break;

}

catch(InterruptedException e) {}

}

}

public void stop(){

stopFlag = true;

t = null;

}

public void paint(Graphics g) {

g.drawString(msg,60,30);

}

}

Problem Description

How to display clock using Applet?

Solution

Following example demonstrates how to display a clock using valueOf() methods of String Class. & using Calender class to get the second, minutes & hours.

import java.awt.\*;

import java.applet.\*;

import java.applet.\*;

import java.awt.\*;

import java.util.\*;

public class ClockApplet extends Applet implements Runnable {

Thread t,t1;

public void start() {

t = new Thread(this);

t.start();

}

public void run() {

t1 = Thread.currentThread();

while(t1 == t) {

repaint();

try {

t1.sleep(1000);

}

catch(InterruptedException e){}

}

}

public void paint(Graphics g) {

Calendar cal = new GregorianCalendar();

String hour = String.valueOf(cal.get(Calendar.HOUR));

String minute = String.valueOf(cal.get(Calendar.MINUTE));

String second = String.valueOf(cal.get(Calendar.SECOND));

g.drawString(hour + ":" + minute + ":" + second, 20, 30);

}

}

Result

The above code sample will produce the following result in a java enabled web browser.

View in Browser.

The following is an another sample example to display clock using Applet.

import java.applet.\*;

import java.awt.\*;

import java.util.\*;

import java.text.\*;

public class javaApplication6 extends Applet implements Runnable {

Thread t1 = null;

int hours = 0, minutes = 0, seconds = 0;

String time = "";

public void init() {

setBackground( Color.green);

}

public void start() {

t1 = new Thread( this );

t1.start();

}

public void run() {

try {

while (true) {

Calendar cal = Calendar.getInstance();

hours = cal.get( Calendar.HOUR\_OF\_DAY );

if ( hours > 12 ) hours -= 12;

minutes = cal.get( Calendar.MINUTE );

seconds = cal.get( Calendar.SECOND );

SimpleDateFormat formatter = new SimpleDateFormat("hh:mm:ss");

Date d = cal.getTime();

time = formatter.format( d );

repaint();

t1.sleep( 1000 );

}

}

catch (Exception e) { }

}

public void paint( Graphics g ) {

g.setColor( Color.blue );

g.drawString( time, 50, 50 );

}

}

Problem Description

How to create different shapes using Applet?

Solution

Following example demonstrates how to create an applet which will have a line, an Oval & a Rectangle using drawLine(), drawOval(, drawRect() methods of Graphics clas.

import java.applet.\*;

import java.awt.\*;

public class Shapes extends Applet {

int x = 300, y = 100, r = 50;

public void paint(Graphics g) {

g.drawLine(30,300,200,10);

g.drawOval(x-r,y-r,100,100);

g.drawRect(400,50,200,100);

}

}

Result

The above code sample will produce the following result in a java enabled web browser.

A line , Oval & a Rectangle will be drawn in the browser.

Problem Description

How to fill colors in shapes using Applet?

Solution

Following example demonstrates how to create an applet which will have fill color in a rectangle using setColor(), fillRect() methods of Graphics class to fill color in a Rectangle.

import java.applet.\*;

import java.awt.\*;

public class fillColor extends Applet {

public void paint(Graphics g) {

g.drawRect(300,150,200,100);

g.setColor(Color.yellow);

g.fillRect( 300,150, 200, 100 );

g.setColor(Color.magenta);

g.drawString("Rectangle",500,150);

}

}

Result

The above code sample will produce the following result in a java enabled web browser.

A Rectangle with yellow color filled in it will be drawn in the browser.

Problem Description

How to goto a link using Applet?

Solution

Following example demonstrates how to go to a particular webpage from an applet using showDocument() method of AppletContext class.

import java.applet.\*;

import java.awt.\*;

import java.net.\*;

import java.awt.event.\*;

public class tesURL extends Applet implements ActionListener {

public void init() {

String link = "yahoo";

Button b = new Button(link);

b.addActionListener(this);

add(b);

}

public void actionPerformed(ActionEvent ae) {

Button src = (Button)ae.getSource();

String link = "http://www."+src.getLabel()+".com";

try {

AppletContext a = getAppletContext();

URL u = new URL(link);

a.showDocument(u,"\_self");

} catch (MalformedURLException e){

System.out.println(e.getMessage());

}

}

}

Result

The above code sample will produce the following result in a java enabled web browser.

View in Browser.

Problem Description

How to create an event listener in Applet?

Solution

Following example demonstrates how to create a basic Applet having buttons to add & subtract two nos. Methods used here are addActionListener() to listen to an event(click on a button) & Button() construxtor to create a button.

import java.applet.\*;

import java.awt.event.\*;

import java.awt.\*;

public class EventListeners extends Applet implements ActionListener {

TextArea txtArea;

String Add, Subtract;

int i = 10, j = 20, sum = 0, Sub = 0;

public void init() {

txtArea = new TextArea(10,20);

txtArea.setEditable(false);

add(txtArea,"center");

Button b = new Button("Add");

Button c = new Button("Subtract");

b.addActionListener(this);

c.addActionListener(this);

add(b);

add(c);

}

public void actionPerformed(ActionEvent e) {

sum = i + j;

txtArea.setText("");

txtArea.append("i = "+ i + "\t" + "j = " + j + "\n");

Button source = (Button)e.getSource();

if(source.getLabel() == "Add") {

txtArea.append("Sum : " + sum + "\n");

}

if(i >j) {

Sub = i - j;

} else {

Sub = j - i;

}

if(source.getLabel() == "Subtract") {

txtArea.append("Sub : " + Sub + "\n");

}

}

}

Result

The above code sample will produce the following result in a java enabled web browser.

View in Browser.

Problem Description

How to display image using Applet?

Solution

Following example demonstrates how to display image using getImage() method. It also uses addImage() method of MediaTracker class.

import java.applet.\*;

import java.awt.\*;

public class appletImage extends Applet {

Image img;

MediaTracker tr;

public void paint(Graphics g) {

tr = new MediaTracker(this);

img = getImage(getCodeBase(), "demoimg.gif");

tr.addImage(img,0);

g.drawImage(img, 0, 0, this);

}

}

Result

The above code sample will produce the following result in a java enabled web browser.

View in Browser.

Problem Description

How to open a link in a new window using Applet?

Solution

Following example demonstrates how to go open a particular webpage from an applet in a new window using showDocument() with second argument as "\_blank".

import java.applet.\*;

import java.awt.\*;

import java.net.\*;

import java.awt.event.\*;

public class testURL\_NewWindow extends Applet implements ActionListener {

public void init() {

String link\_Text = "google";

Button b = new Button(link\_Text);

b.addActionListener(this);

add(b);

}

public void actionPerformed(ActionEvent ae) {

Button source = (Button)ae.getSource();

String link = "http://www."+source.getLabel()+".com";

try {

AppletContext a = getAppletContext();

URL url = new URL(link);

a.showDocument(url,"\_blank");

} catch (MalformedURLException e) {

System.out.println(e.getMessage());

}

}

}

Result

The above code sample will produce the following result in a java enabled web browser.

View in Browser.