

STRATEGIC WEBSITE TECHNOLOGIES

LECTURE 8 (JAVA APPLETS)

Java Applets

An applet is a Java program that runs on a client.

When an applet is compiled it results in a .class file.

A .class file would normally consist of a set of bytecodes that can be interpreted by the Java interpreter for execution.

The resulting bytecode file (.class file) from an applet that usually would reside on a server can be brought into an HTML file to be executed.

ie applets are loaded when the user browses an HTML document containing the applet.

```
<html>
<applet code="hello.class" width=275 height=35>
</applet>
</html>
```

Java Syntax

```
import java.applet.Applet; //imports Applet class from java.applet package
import java.awt.Graphics; //imports Graphics class from java.awt package

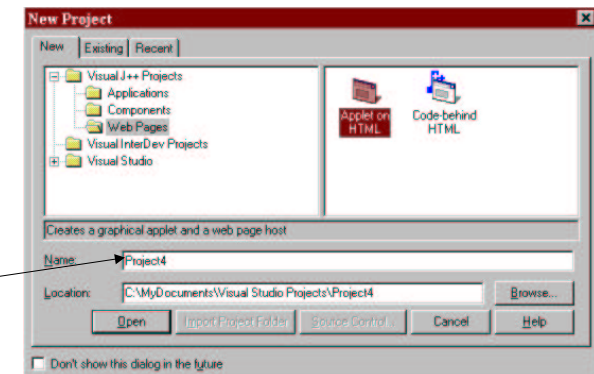
public class hello extends Applet
{
    public void paint (Graphics g)
    {
        g.drawString("Hello world!!!", 25, 25);
    }
}
```

Note that you specify exact coordinates.

Using VJ++ for writing Java Applets

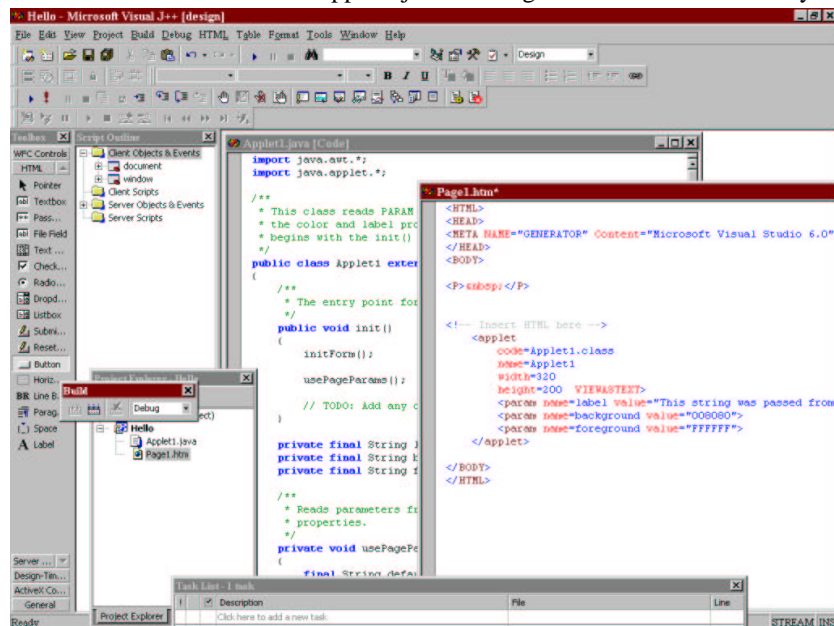
One of the platforms that can be used for creating Java applets is the Microsoft's Visual J++ environment.

When started we select Open a new project and Applet on HTML as the project type.



Choose the name properly.

VJ++ creates two files named Applet1.java and Page1.htm and fills them for you.



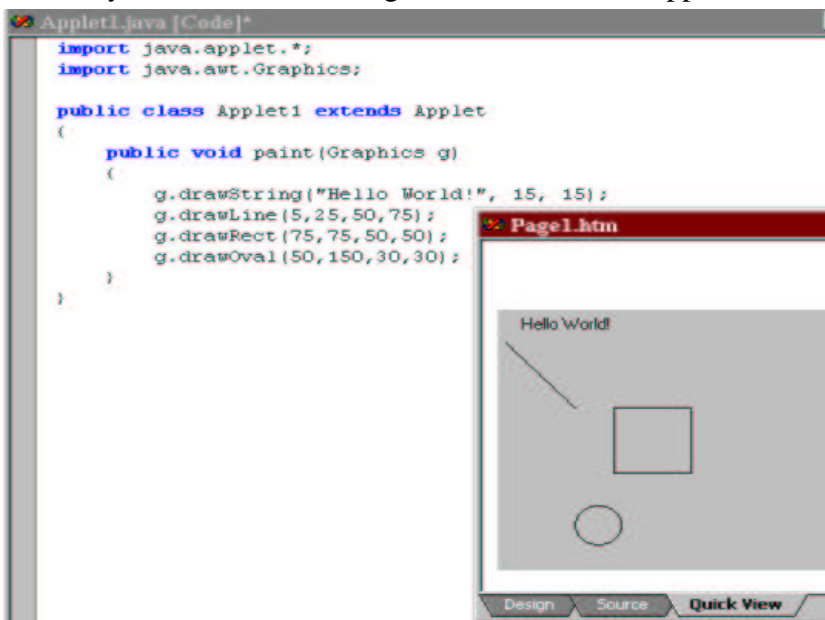
Retype Applet1.java as follows and build it. In page1.htm check quick view tab to see the applet you have created.

You can change the size of the applet from the source tab of the page1.htm.

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

public class Applet1 extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Hello World!", 15, 15);
    }
}
```

It is easy to draw lines, rectangles, ovals etc on the applet area.



Communicating with the HTML code

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

public class Applet1 extends Applet
{
    String name;

    public void init(){
        name=getParameter("name");
    }

    public void paint(Graphics g)
    {
        g.drawString("Hello "+name+"!", 15, 15);
        g.drawLine(5,25,50,75);
        g.drawRect(75,75,50,50);
        g.drawOval(50,150,30,30);
    }
}
```

Mixing Java and JavaScript

```
<HTML>
<BODY>
<Script>
nm=prompt("Enter your name?","Turan");
document.write("<applet "+
               "code=Applet1.class "+
               "name="+ nm +
               " width=320 "+
               "height=200 VIEWASTEXT> "+
               "</applet>");
</Script>
</BODY>
</HTML>
```

Scribble

In the following example we illustrate how Java can be used to create complex interactions with the user.



In this example, the user is able to draw any graphics on a designated area in the browser. It could be possible to save these pictures in hard disk.

```
import java.awt.*;
import java.applet.*;
public class Scribble extends Applet {
    private int last_x, last_y;
    private Color current_color = Color.black;
    private Button clear_button;
    private Choice color_choices;
    public void init()
    {
        this.setBackground(Color.white);
        clear_button = new Button("Clear");
        clear_button.setForeground(Color.black);
        clear_button.setBackground(Color.lightGray);
        this.add(clear_button);
        color_choices = new Choice();
        color_choices.addItem("black");
        color_choices.addItem("red");
        color_choices.addItem("yellow");
        color_choices.addItem("green");
        color_choices.setForeground(Color.black);
        color_choices.setBackground(Color.lightGray);
        this.add(new Label("Color :"));
        this.add(color_choices);
    }
}
```

Scribble.java

```
public boolean mouseDown(Event e, int x, int y)
{
    last_x=x; last_y=y;
    return true;
}
public boolean mouseDrag(Event e, int x, int y)
{
    Graphics g = this.getGraphics();
    g.setColor(current_color);
    g.drawLine(last_x, last_y, x, y);
    last_x=x; last_y=y;
    return true;
}
```

```

public boolean action(Event event, Object arg)
{
    if(event.target == clear_button)
    {
        Graphics g=this.getGraphics();
        Rectangle r=this.getBounds();
        g.setColor(this.getBackground());
        g.fillRect(r.x,r.y,r.width,r.height);
        return true;
    }
    else if(event.target == color_choices) {
        if(arg.equals("black")) current_color = Color.black;
        else if(arg.equals("red")) current_color = Color.red;
    else if(arg.equals("yellow")) current_color = Color.yellow;
    if(arg.equals("green")) current_color = Color.green;
        return true;
    }
    else return super.action(event, arg);
}
}

```

Security Issues

Java includes security features to reinforce its use on the Internet.

Java uses code verification and limited file system access to make sure that the code won't damage anything on your local machine.

When an applet is downloaded to your local machine, the code is checked to make sure that the applet only uses valid system calls and methods. If the applet passes verification, it is allowed to run, but applets are not given access to the local machine's file system.

Limiting access to the file system imposes some limits on applets, but it also prevents applets from intentionally removing files, planting viruses, or otherwise reeking havoc with your system

Why applets?

- You can write an application in Java.
- An application is like a **program** written in any language and **will run on the user's machine**.
- This is useful when you need to execute **graphics or audio intensive** programs.

Servlets

A servlet is like an applet but runs on the server unlike an applet that runs on a client.