

Applet Basics

A Java application is a stand-alone program that is executed using the Java interpreter. Conversely, a Java *applet* is meant to be transported over a network and executed within a web browser that contains a Java interpreter. An application must have a `main` method. An applet does not have a main method, but when you write an applet, you must overwrite at least one of several methods inherited from its super class `Applet`.

Java applications may be either console programs, such as those that we have written in the previous labs, or they may be graphical. Applets are always graphical. Writing graphical programs requires an understanding of the basics of the Java language, much of which we have now covered. In this lab we will write two types of applets. The first will draw on a `Graphics` object and the second will introduce components such as labels and text fields.

To execute an applet, the applet must be embedded in a HTML (Hyper-Text Markup Language) file. HTML code is interpreted by a browser. The result is the a display of a web page with text in different formats, images, and links to other web pages. The browser determines how to display the elements in a web page based on a set of HTML *tags* that generally come in pairs. For example, listed are some tags and what they define.

beginning of	end of	
<code><html></code>	<code></html></code>	an HTML file
<code><body></code>	<code></body></code>	the body of the page
<code><applet></code>	<code></applet></code>	applet
<code><p></code>	<code></p></code>	paragraph
<code></code>	<code></code>	bold font

The HTML statement

```
<applet code="Man.class" width=300 height=200> </applet>
```

embeds an applet with compiled bytecode stored in the file `Man.class`. The `width` and `height` define the size of the applet measured in pixels (picture elements) as it appears on a web page.

The Java System Development Kit includes an *appletviewer* that allows us to develop and experiment with applets outside of a browser. The *appletviewer* executes an applet that is embedded in an HTML document, while ignoring the remaining document. Your teacher will tell you how to access the *appletviewer* on your particular system.

For our first applet we will use the file `J12E01.html`

```
<html>
  <body>
    <p>My <b>very first</b> applet</p>
    <applet CODE="J12E01.class" WIDTH=400 HEIGHT=300>
  </applet>
  </body>
</html>
```

An applet that has a width of 400 and a height of 300 has 120,000 (= 400 x 300) pixels, each of which is displayed in a specified color. The combination of colored pixels produces an image on the applet. Each pixel is associated with an ordered pair of positive coordinates, (x,y). The upper left hand corner has coordinate (0,0). The positive x-axis runs horizontally to the right of (0,0) and the positive y-axis runs vertically down from(0,0). The pixel with coordinates (50,100) is 50 pixels to the right and 100 pixels below (0,0).

All applets must import the `java.applet.Applet` class, which defines attributes and some basic methods that all applets inherit. Because applets are meant to be transported over the internet, the class must be modified by `public`. All applets must take this form:

```
import java.applet.Applet;

public class ClassName extends Applet
{
}
```

A First Applet

Because our first applet will draw on a `Graphics` object, the package `java.awt.*` must also be imported. The AWT (Abstract Windowing Toolkit) is a package that defines classes that are used in graphical programs. To draw on a `Graphics` object the applet overwrites the method

```
public void paint(Graphics g)
```

that is inherited from the `Applet` class. This method is automatically called by the browser, which creates and passes the `Graphics` object to the parameter `g`.

The `Graphics` class provides the following methods that are invoked on the `Graphics` object:

```
public void drawRect(int x, int y, int width, int height)
```

draws a rectangle with upper left-hand corner at the pixel (x, y) with `width` pixels in the direction of the x-axis and `height` pixels in the direction of the y-axis.

```
public void drawOval(int x, int y, int width, int height)
```

draws an oval inscribed in a non-visible rectangle defined by the parameters.

```
public void drawLine(int x1, int y1, int x2, int y2)
```

draws a line segment with endpoints (x_1, y_1) and (x_2, y_2) .

```
public void drawString(String s, int x, int y)
```

draws a string on an invisible line with left-hand endpoint at (x, y) .

Experiment 12.1

Step 1. The applet code is stored in `J12E01.java`. Compile the code. Then execute the applet using `appletviewer` and the HTML file `J12E01.html`.

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

public class J12E01 extends Applet
{
    public void paint(Graphics g)
    {
        g.drawRect(0,0,50,100);
        g.drawOval(100,20,50,100);
    }
}
```


Step 3. Modify the applet by adding

```
g.drawLine(0,299, 399,0);  
g.drawString("Applets are fun",150, 290);
```

to the end of the `paint` method. Compile and execute the applet. Describe the results and add the additional figures to the diagram in Step 1.

Step 4. Add the following code to the end of the `paint` method.

```
g.fillRect(300,100,40,40);  
g.fillOval(300,150,40,40);
```

Compile and execute the applet. Describe the results and add to the diagram in Step 1.

Step 5. The `Graphics` class also defines the methods

```
void drawArc(int x, int y, int w, int h, int begin, int sweep)  
void fillArc(int x, int y, int w, int h, int begin, int sweep)
```

which draw either an arc or a filled arc inside the bounding rectangle with upper left-hand corner at (x,y) , width w and height h . The arc begins at angle `begin` and sweeps through an angle of `sweep`. When determining the angles, visualize a coordinate system superimposed on the bounding rectangle with the origin at the center of the rectangle. The positive x -axis is

Refresh or reload the browser page. What is the effect of the HTML tag <u> ?

The Color Class

The `java.awt` package also contains a `Color` class that can be used to add color to our applets. We have seen that the default color for drawing on an applet is black. Java uses RGB (red-green-blue) colors where the amount of any one of these three colors is stored as an integer value between 0 and 255. A value of 0 indicates that a color is not present and a value of 255 indicates that the maximum amount of the color is present. Color objects can be constructed using the constructor `public Color(int red, int green, int blue)`. The statement

```
Color purple = new Color(200, 0, 175);
```

creates a new `Color` object with the specified amounts of red (200), green (0) and blue (175). The `Color` class also defines some constant colors such as `black`, `blue`, `cyan`, `gray`, `green`, `magenta`, `orange`, `pink`, `red`, and `white`. To access these constants, which are not capitalized, use the name of the class and the dot operator. For example: `Color.red` and `Color.blue`.

To change the drawing color on a graphics object, use the `Graphics` class method

```
public void setColor(Color c)
```

Experiment 12.2

Step 1. The code for this applet is found in the file `J12E02.java`. The related HTML file, `J12E02.html`, sets the width and height of the applet to 250.

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

public class J12E02 extends Applet
{
    public void paint(Graphics g)
    {
        g.setColor(Color.red);
        g.fillRect(0,0,30,30);
        g.setColor(Color.cyan);
        g.fillRect(30,0,30,30);
    }
}
```


