

# Chapter 13

---

## Applets and HTML

- HTML
- Applets

# Overview

---

- Applets: Java programs designed to run from a document on the Internet
- HTML: Hypertext Markup Language
  - » a language used to create Internet documents
- Applets run within an HTML document
- HTML documents are run by a web browser, e.g.
  - » Netscape Navigator or
  - » Microsoft Internet Explorer

# Applets

---

- Applet: small application
- Applets are Java programs embedded in HTML documents
  - » Therefore, they run in a browser that views the document
- They are a derived class of the class `JApplet`, which is derived from `Applet`. `Applet` is in turn derived from `Panel`.
  - » Therefore, it is both a `Container` and `Component`
- `JApplet` class is in the Swing library, so import `javax.swing.*`

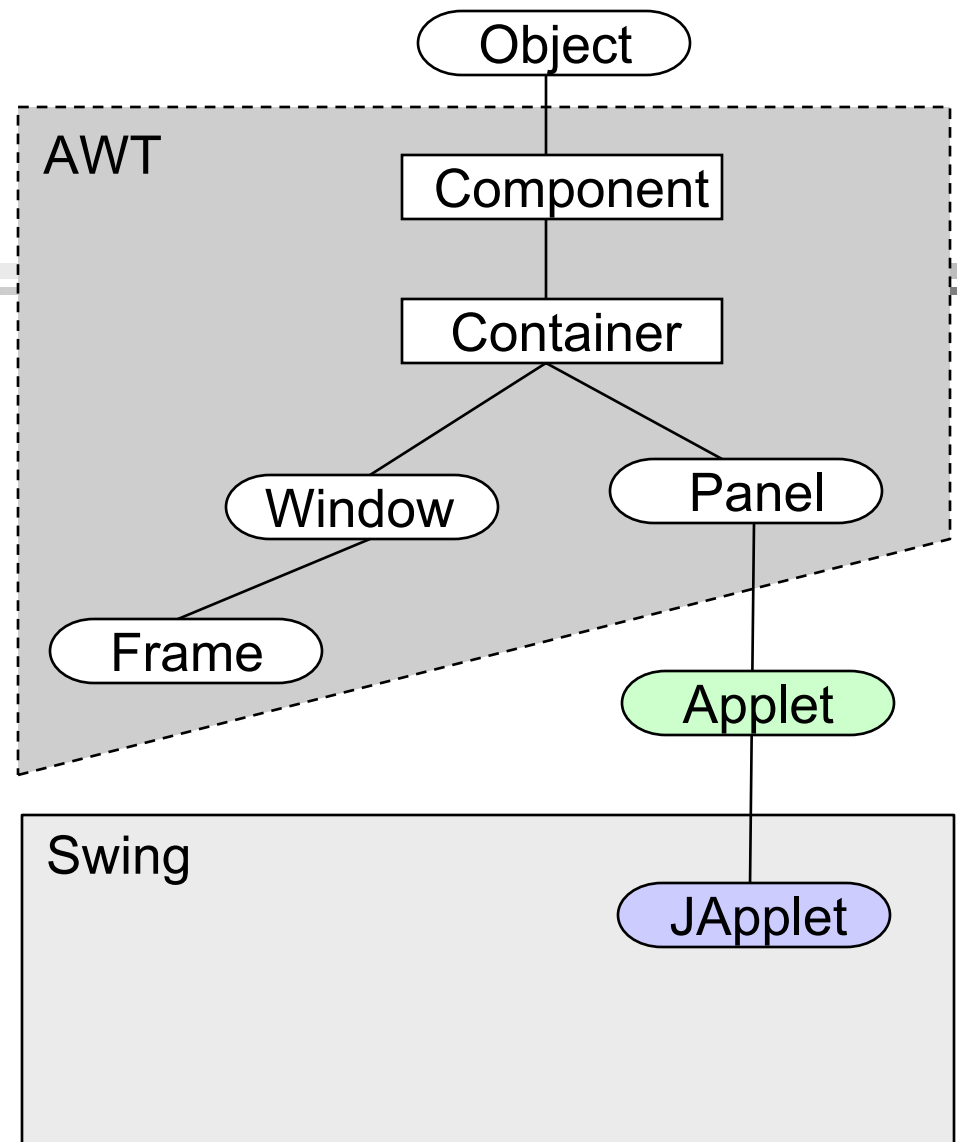
# Placing **JApplet** in the Class Hierarchy

- JApplet is part of Swing and inherits from the Applet class.
- The Applet class is older than JApplet and is not part of AWT or Swing.

Key:

*Class*

*Abstract Class*



If there is a line between two classes, then the lower class is a derived class of the higher class.

# Applet Basics

---

- When writing applets your list of import statements will usually look like this:

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;
```

- Applets do not normally have a `main` method

# Applet Basics

---

- Applets do not normally use constructors but they do use a method named `init()` for similar purposes:
  - » setting colors
  - » adding buttons
  - » adding text fields, etc.
  - » the `init()` method has no parameters
- The HTML document takes care of sizing the applet
- Use a layout manager when defining an applet
  - » use the `setLayout` method
  - » `BorderLayout` is the default

# Example: a Trivial Applet

HelloApplet.java

```
import javax.swing.*;  
import java.awt.*; // for Container class
```

```
public class HelloApplet extends JApplet
```

```
{
```

```
    public void init()
```

```
    {
```

```
        Container contentPane = getContentPane();
```

```
        contentPane.setLayout(new FlowLayout());
```

```
        JLabel myFirstLabel = new JLabel("Hello out there!");
```

```
        contentPane.add(myFirstLabel);
```

```
    }
```

```
}
```



Defining an applet

# Running an Applet

---

One way to run an applet:

- » put the applet in an HTML document
  - good practice is to name the HTML document the same as the applet, but with ".html" instead of ".java"
- » then run it using `appletviewer`
- » like `javadoc`, `javac`, and `java`, `appletviewer` comes with the JDK (Java Development Kit)

Syntax:

```
appletviewer Applet_File_Name.html
```

Example:

```
appletviewer FirstApplet.html
```



# Converting a Swing Application to an Applet

---

Steps to convert an application using `JFrame` to an applet:

1. Derive the class from `JApplet` instead of deriving it from `JFrame`
2. Remove the main method.
3. Replace the constructor with a method named `init()`. Some statements from the constructor will be removed as in the following steps.
4. Delete any invocation of `addWindowListener`.
5. Delete any invocation of `setTitle`.
6. Delete any invocation of `setSize`.
7. Make an HTML page with an `APPLET` tag that refers to the byte-code (`.class`) version of the applet.

# Icons and the Class **ImageIcon**

- An *icon* is a small picture
  - » not really required to be small in Java
- Class **ImageIcon** is used to convert a picture file to a Swing icon

Syntax:

```
ImageIcon Name_Of_ImageIcon =  
    new ImageIcon(Picture_File_Name) ;
```

- » *Picture\_File\_Name* is a string giving a path name to the picture file

Example:

```
ImageIcon smileyFaceIcon =  
    new ImageIcon("smiley.gif") ;
```

```
import javax.swing.*;  
import java.awt.*;
```

# Adding Icons to an Applet

```
public class DukeApplet extends JApplet
```

```
{
```

```
    public void init()
```

```
    {
```

Uses content pane as  
JFrame does

```
        Container contentPane = getContentPane();
```

```
        contentPane.setLayout(new BorderLayout());
```

```
        JLabel spacer = new JLabel("                ");
```

```
        contentPane.add(spacer, "West");
```

```
        JLabel niceLabel = new JLabel("Java is fun!");
```

```
        ImageIcon dukeIcon =
```

```
            new ImageIcon("duke_waving.gif");
```

```
        niceLabel.setIcon(dukeIcon);
```

Add icon to label

```
        contentPane.add(niceLabel, BorderLayout.CENTER);
```

```
    }
```

Add label to content pane of applet

```
}
```

# HTML:

## HyperText Markup Language

---

- *Hypertext*: text with items you can click with a mouse to send you to another document
- *Link*: an item you can click with the mouse to go to another document
  - » also called *hyperlink*
- *Page*: an HTML document
  - » it is *not* equal to a page of print or a single screen of display
  - » *HTML page* and *HTML document* mean the same thing
- HTML is a "suggestive formatting" language
  - » it is not a full-blown programming language like Java
  - » it *suggests* to the browser how to display text or other objects

# HTML Basics

---

- Most HTML commands are of the form

```
<COMMAND>  
some text  
</COMMAND>
```
- Some commands do not require the `</COMMAND>` part
- HTML is *not* case sensitive
  - » i.e. the *commands* are not
  - » text displays in whatever case it is in in the document
- HTML files are plain text files created with any text editor
- HTML file names should follow the operating system's rules and end in `.html` or `.htm`

# A Few HTML Commands

<code>&lt;H1&gt; Page Title &lt;/H1&gt;</code>	Heading 1- largest, most prominent heading
<code>&lt;H2&gt; text &lt;/H2&gt;</code>	Heading 2 - a sub-heading
<code>&lt;H3&gt;</code> , etc.	Additional sub-headings
<code>&lt;BR&gt;</code>	Break - begin a new line ( <code>&lt;/BR&gt;</code> not required)
<code>&lt;P&gt;</code>	Paragraph - begin a new paragraph ( <code>&lt;/P&gt;</code> not required)
<code>&lt;!-- comments --&gt;</code>	Comment field - not displayed by browser
<code>&lt;CENTER&gt;text&lt;/CENTER&gt;</code>	Center text horizontally on page
<code>&lt;TITLE&gt;title&lt;/TITLE&gt;</code>	Used for browser "book marks" (displays in browser title bar)

# Outline of a Basic Page and a Few More HTML Commands

<b>&lt;HTML&gt;</b>	begin an HTML document
<b>&lt;HEAD&gt;</b>	begin the document's header (title usually goes here)
<b>&lt;/HEAD&gt;</b>	end of document's header
<b>&lt;BODY&gt;</b>	begin stuff that will appear in the main window of the browser blank lines are ignored - use them to make code more readable (body of html document: html commands, text, applets, etc.)
<b>&lt;/BODY&gt;</b>	end stuff that displays in main window of browser
<b>&lt;ADDRESS&gt;</b>	date and link to the email address that follows (email address of person maintaining the page)
<b>&lt;/ADDRESS&gt;</b>	end email address
<b>&lt;/HTML&gt;</b>	end the HTML document

# HTML: Suggestive Formatting

---

- Most HTML commands *suggest* rather than dictate how text will be displayed
- For example, Heading 1 in one browser may be in Arial font but another browser may display it in Times New Roman
  - » the font size and other attributes may differ as well
- Different browsers and systems also result in different line breaks, colors, etc., so WYSIWYG (What You See Is What You Get) does not apply to either the text HTML file or what displays in your browser
  - » getting HTML documents to display nicely in all browsers and systems is important but non-trivial
  - » the examples used here are simple and should not be a problem, so we will not go into any detail about differences in browsers



# Inserting a Hyperlink

## Hyperlinks - The real power of HTML

- Syntax:

```
<A HREF="Path_To_Document"> text to display</A>
```

- » the path can be local - any absolute or relative path to a local file
- » or . . .

**It can be a URL (internet address)  
plus path name to a file on a remote system!**

- Example:

```
<A HREF="http://www-cse.ucsd.edu/users/Savitch">  
Walter Savitch</A>
```

Displays as: [Walter Savitch](http://www-cse.ucsd.edu/users/Savitch)

# URL: Uniform Resource Locator

---

- An internet address that uniquely identifies a location
- A kind of path name for the Internet
- But not just an ordinary address
  - » the first part tells what *protocol* to use to make the link
  - » a protocol is a set of rules for transferring and interpreting data
  - » both ends of a link must implement the same protocol to have intelligible communications
  - » http and ftp are two examples of Internet protocols
  - » http is the name of the protocol for HTML documents

# *Gotcha*: Not Using Your Reload (Refresh) Button

---

- Browsers keep copies of most recently used HTML pages
  - » runs faster because it can recover pages much faster
  - » might not show changes that you make to a page
- Click Reload, Refresh (or similar) button
  - » browser will reload page and show you the latest version
- Reload, Refresh do not always work with Java applets

# Automatic Documentation with Java

---

- Standard Java includes a utility called `javadoc` that automatically generates documentation files from source code for class definitions
- Usage: `javadoc yourFile.java`
- `javadoc` produces an HTML document that can be read in a browser
  - » it includes selected lines from the source file, e.g. comments and headers
- `javadoc` is described in Appendix 9

# Displaying a Picture

---

- Here is the HTML command for inserting a picture in a document:

```
<IMG SRC="Picture_File">
```

- For example, if the picture file you want is named *mypicture.gif* and it is in the *images* subdirectory of the directory that contains your HTML document:

```
<IMG SRC="images/mypicture.gif">
```

- Various picture formats may be used and are generally indicated by the ".xxx" extension in the file name
  - » .gif and .jpg are two common ones

# HTML Is a Low-Level Language

---

- HTML for web browser is analogous to assembly language for a computer
  - » many details that designer must take care of
- Most web page designers now use a high-level design language that translates into HTML
- Three examples
  - » Dreamweaver (Macromedia, Inc.)
  - » FrontPage (Microsoft Corporation)
  - » GoLive (Adobe Systems Inc.)

# Running an Applet from an HTML Page

---

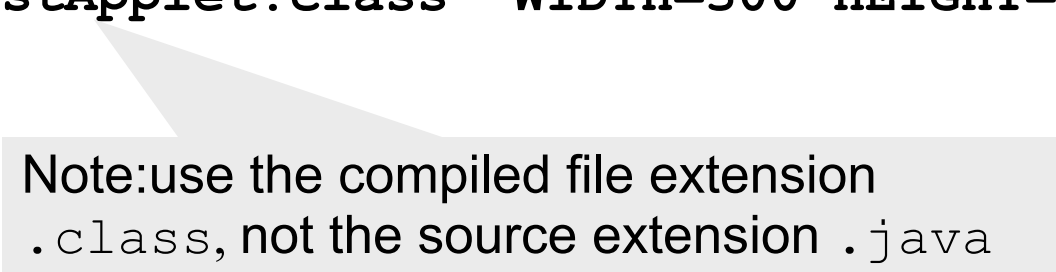
The normal way to run an applet is from an HTML document.

Syntax:

```
<APPLET CODE="Name_Of_.class_File" WIDTH=Integer
        HEIGHT=Integer>
</APPLET>
```

Example:

```
<APPLET CODE="FirstApplet.class" WIDTH=300 HEIGHT=200>
</APPLET>
```



Note: use the compiled file extension  
.class, not the source extension .java

# Another Way to Test an Applet

---

- Use the appletviewer in an integrated Java development environment
- There are many commercial development environments that provide handy tools to write, compile, run and debug Java programs
- Typically they have a menu selection something like "Run" or "Execute" to run an application
  - » usually they will automatically create an applet viewer if you run an applet



# *Gotcha:* Using an Old Web Browser

---

- Updates to browsers are usually much later than updates to core Java language.
  - » Web browsers don't use the same Java interpreter that runs Java applications.
- Test your applets in different browsers and in different versions of browsers.
  - » Can't expect everyone to have most recent browser.
  - » Might need to use older `Applet` class if you want your applet to run in older browsers.

# The Older **Applet** Class

---

- Older versions of web browsers can't run applets that use `JApplet` and other Swing classes.
- Using the old `Applet` class instead of `JApplet` will allow your applet to run in more browsers.
- To convert a `JApplet` program to an `Applet` program:

- » Replace:

```
import javax.swing.*;
```

with

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

- » Replace Swing classes that start with **J** (`JButton`, `JPanel`, etc.) with AWT classes that don't start with **J** (`Button`, `Panel`, etc.)
- » Remove references to the content pane of the applet.

# Applets and Security

---

- Applets are someone else's program running on your computer.
- Security issues:
  - » viruses
  - » reading confidential information
  - » corrupting operating system
- Security features of applets to help avoid these problems:
  - » Applets cannot run any programs on your computer.
  - » Applets cannot read or write to files on your computer (unless the applet originated on your computer).

# Summary

---

- HTML is a language for writing documents to be read across the Internet or with a web browser.
- HTML stands for Hypertext Markup Language.
- Applets are Java programs designed to be placed in and run from an HTML document.
- Applets are derived from the class `Applet` and are similar to AWT GUIs derived from the class `Frame`.