

# 11 Applets



# Topics

- Applet definition
- Creating Applets
- Applet Methods
  - The Applet Life Cycle
  - The paint Method
  - The showStatus Method
  - Playing Audio Clips
- Applet HTML Tags



# Applets

- Special type of java program that is executed via the Internet
- Typically run on a web browser
- Not allowed access to the computer on which they are being run for security reasons
- The *Applet* class
  - Subclass of the *Panel* class defined in the AWT



# Hello World Applet

```
1 //Saved in AppletDemo.java
2 import java.awt.*;
3 import java.applet.*;
4 /* insert this part in the html code
5     <applet code="AppletDemo" width=300 height=100>
6     </applet>
7 */
8 public class AppletDemo extends Applet {
9     public void paint(Graphics g) {
10         g.drawString("Hello world!", 80, 25);
11     }
12 }
```



# Running Applets

- Compiled normally but not executed using the *java* command
  - Run on a web browser
  - Run using the applet viewer
- Opening an applet through a web browser
  - Open the HTML document in which the applet is embedded using the applet HTML tag
- Opening an applet is through the *appletviewer* command

```
appletviewer <java filename>
```



# Hello World Applet

- To run the applet example:

```
appletviewer AppletDemo.java
```

- The applet is created with a width of 300 pixels and a height of 100 pixels

```
<applet code="AppletDemo" width=300 height=100>  
</applet>
```

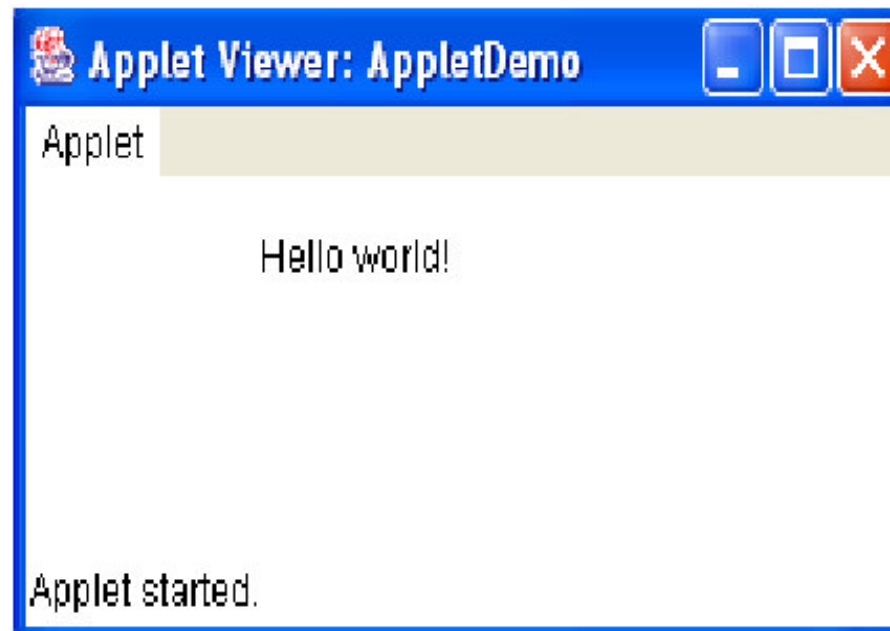
- The *drawString* method draws the "Hello world!" string at the pixel position (80,25) counting down then right

```
g.drawString("Hello world!", 80, 25);
```



# Hello World Applet

- Sample output:



# Creating Applets

- Extend the *Applet* class
- Import the *java.applet* package
  - The *Applet* class is found in the *java.applet* package
- Import the *java.awt* package
  - The *Applet* class is a subclass of the *Panel* class
  - Some methods and fields of the *Applet* class are found in the *Panel* class
  - The parent class is found in the *java.awt* package





# The Applet Life Cycle

- Has no *main* method
- The browser or the applet viewer interacts with the applet through the following methods:
  1. `init()`
  2. `start()`
  3. `stop()`
  4. `destroy()`
- Some of these methods are overridden when creating applets



# The Applet Life Cycle

- Details of applet life cycle methods
  - `init()`
    - First method called when the applet is loaded
  - `start()`
    - Next method called after *init*
    - Invoked everytime the applet's HTML document and the applet itself is displayed
  - `stop()`
    - Called when the web browser leaves the applet's HTML document
    - Inform the applet that it should stop its execution
  - `destroy()`
    - Called when the applet needs to be removed from the memory completely
    - The *stop* method is always called before this method is invoked



# The Applet Life Cycle

```
1 import java.applet.*;
2 import java.awt.*;
3 /*
4     <applet code="LifeCycleDemo" width=300 height=100>
5     </applet>
6 */
7 class LifeCycleDemo extends Applet {
8     String msg = "";
9     public void init() {
10         msg += "initializing... ";
11         repaint();
12     }
13 //continued...
```



# The Applet Life Cycle

```
14 public void start () {
15     msg += "starting... ";
16     repaint ();
17 }
18 public void stop () {
19     msg += "stopping... ";
20     repaint ();
21 }
22 public void destroy () {
23     msg += "preparing for unloading...";
24     repaint ();
25 }
26 //continued...
```



# The Applet Life Cycle

```
27     public void paint(Graphics g) {  
28         g.drawString(msg, 15, 15);  
29     }  
30 }
```

- Sample HTML document with an embedded applet

```
1 <HTML>  
2 <TITLE>Life Cycle Demo</TITLE>  
3     <applet code="LifeCycleDemo" width=300  
4                                     height=100>  
5     </applet>  
6 </HTML>
```



# The *paint* Method

- Important method inherited from *Applet's* ancestor class *Component*
- Called everytime the applet's output needs to be redrawn.
  - Example: An applet is hidden by another window and then made visible again
- Usually overridden to customize the look of your applet
  - Hello World example



# The *showStatus* Method

- An applet has a status window
  - Informs you of what the applet is currently doing
- If you want to output to the status window
  - Invoke the *showStatus* method and pass a *String* argument



# The *showStatus* Method

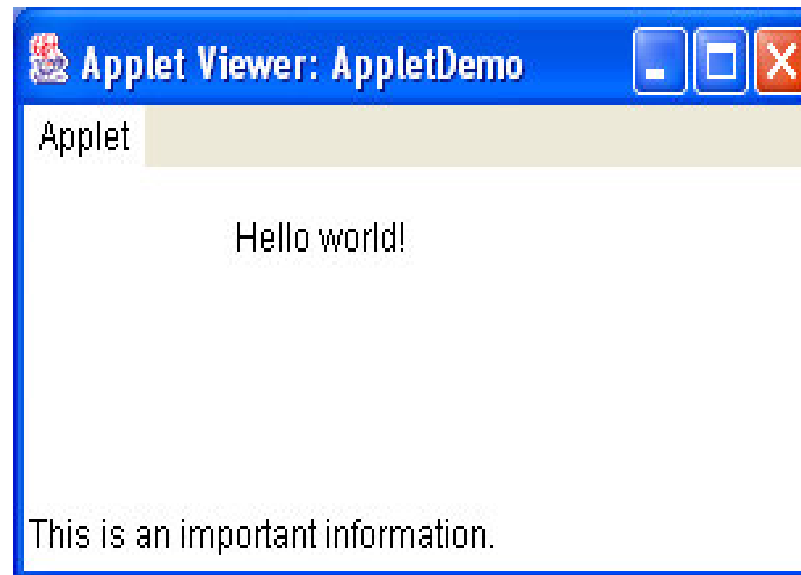
```
1 import java.awt.*;
2 import java.applet.*;
3 /*
4     <applet code="AppletDemo" width=300 height=100>
5     </applet>
6 */
7 public class AppletDemo extends Applet {
8     public void paint(Graphics g) {
9         g.drawString("Hello world!", 80, 25);
10        showStatus("This is an important information.");
11    }
12 }
```





# The *showStatus* Method

- Sample output:



# Playing Audio Clips

- Applets also allow you to play audio files
- Steps in playing an audio clip in an applet:
  1. Get the audio clip
    - Use the *getAudioClip* method
  2. Play the audio clip
    - Use the *play* method on the audio clip object
    - Or use *loop* method on the audio clip object



# Playing Audio Clips

- Difference between *play* and *loop*:
  - The *play* method
    - Plays the audio clip exactly once
  - The *loop* method
    - Loops on the audio clip until the *stop* method is called



# Playing Audio Clips

```
1 import java.awt.*;
2 import java.applet.*;
3 /*
4     <applet code="AudioApplet" width=300 height=100>
5     </applet>
6 */
7 public class AudioApplet extends Applet {
8     AudioClip ac;
9     //continued...
```



# Playing Audio Clips

```
10 public void init() {
11     try {
12         /* audio clip is saved in same directory
13            as java code */
14         /* spaceMusic- downloaded from java.sun.com */
15         ac = getAudioClip(getCodeBase(),
16                            "spaceMusic.au");
17         ac.loop();
18     } catch (Exception e) {
19         System.out.println(e);
20     }
21 }
```

```
22 //continued... Introduction to Programming 2
```



# Playing Audio Clips

```
23 public void stop() {
24     ac.stop();
25 }
26 public void paint(Graphics g) {
27     g.drawString("Playing space music!", 80, 25);
28 }
29 }
```



# Applet HTML Tags

```
<APPLET
  [CODEBASE = codebaseURL]
  CODE = appletFile
  [ATL = alternateText]
  [NAME = appletInstanceName]
  WIDTH = widthInPixels HEIGHT = heightInPixels
  [ALIGN = alignment]
  [VSPACE = vspaceInPixels] [HSPACE = hspaceInPixels]
>
<!-- continued.. -->
```



# Applet HTML Tags

```
[<PARAM NAME = parameterName1 VALUE = parameterValue1>]
[<PARAM NAME = parameterName2 VALUE = parameterValue2>]
...
[<PARAM NAME = parameterNamen VALUE = parameterValuen>]
[HTML that will be displayed in the absence of Java]
</APPLET>
```





# Applet HTML Tags: Keywords

<i>Applet HTML Tag Keywords</i>	
CODEBASE	Directory where the applet class is located. Refers to the HTML document's URL directory by default.
CODE	Name of the file containing the applet code. With or without the <i>.java</i> or <i>.class</i> extension name.
ALT	Text displayed if the browser understands applet tags but cannot currently execute the applet. May happen if Java is disabled.
NAME	Name of the applet. Used to allow other applets to communicate with this applet by referring to it by its name.
WIDTH, HEIGHT	Width and height of the applet window. Specified in pixels.



# Applet HTML Tags: Keywords

## *Applet HTML Tag Keywords*

### ALIGN

Alignment or positioning of the applet. Either "left", "right", "top", "bottom", "middle", "baseline", "texttop", "absmiddle", or "absbottom". Default placement depends on the environment.

"top" - top of applet aligned with tallest item in the current line.

"bottom", baseline - bottom of applet aligned with bottom of other content in the current line.

"middle" - middle of applet aligned with bottom of other content in the current line.

"texttop" - top of applet aligned with top of tallest text in the current line.

"absmiddle" - middle of applet aligned with vertical middle of other content in the current line.

"absbottom" - bottom of applet aligned with bottom of other content in the current line

### VSPACE, HSPACE

Space above and below (VSPACE) and on the sides (HSPACE) of the applet.

### PARAM NAME, VALUE

To specify parameters that can be passed to applets; applets can invoke the `getParameter(String paramName)` method.



# Applet HTML Tags: Example

```
1  /* Accessing a parameter specified at the HTML tag */
2  import java.awt.*;
3  import java.applet.*;
4  /*
5     <applet code="ParamDemo" width=300 height=100>
6         <param name="myParam" value="Hello world!">
7     </applet>
8  */
9  public class ParamDemo extends Applet {
10     public void paint(Graphics g) {
11         g.drawString(getParameter("myParam"), 80, 25);
12     }
13 }
```



# Summary

- Applet definition
- Creating Applets
  - Extend the *Applet* class
  - Import these packages:
    - *java.applet*
    - *java.awt*
  - Use the applet html tags



# Summary

- Applet Methods
  - The Applet Life Cycle
    1. *init*
    2. *start*
    3. *stop*
    4. *destroy*
  - The *paint* Method
  - The *showStatus* Method
  - Playing Audio Clips
    1. *getAudioClip*
    2. *play* or *loop*
    3. *stop*



# Summary

- Applet HTML Tags

```
<APPLET
  [CODEBASE = codebaseURL]
  CODE = appletFile
  [ATL = alternateText]
  [NAME = appletInstanceName]
  WIDTH = widthInPixels HEIGHT = heightInPixels
  [ALIGN = alignment]
  [VSPACE = vspaceInPixels] [HSPACE = hspaceInPixels]
>
[<PARAM NAME = parameterName1 VALUE = parameterValue1>]
[<PARAM NAME = parameterName2 VALUE = parameterValue2>]
...
[<PARAM NAME = parameterNameN VALUE = parameterValueN>]
[HTML that will be displayed in the absence of Java]
</APPLET>
```

