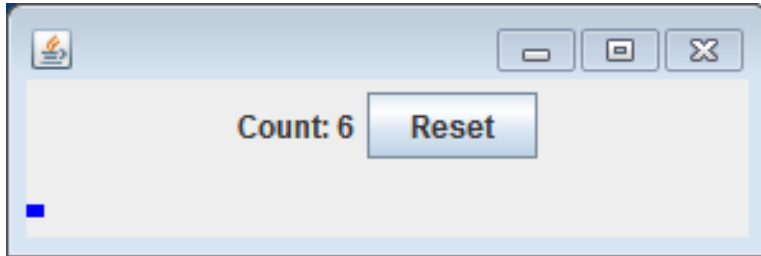


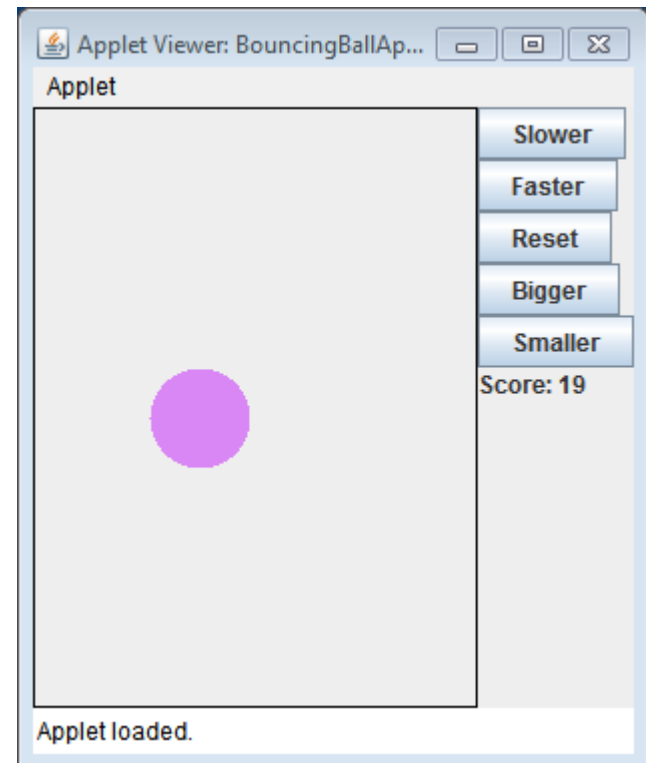
# Java Applets



```
<html>
  <body>
    <h2>Don't panic about frogs!</h2>
    <applet code="PanicAppletParam.class" width="500" height="500">
      <param name="image" value="frog.jpg">
      <param name="sound" value="frog.wav">
      <param name="delay" value="500">
      Java plugin not installed
    </applet>
  </body>
</html>
```

# Overview

- Complete slideshow GUI
- Applets
  - A Java GUI inside your browser!
    - Quite similar to building a desktop Java GUI
  - Important methods
  - Drawing images
  - Playing audio
  - Getting input parameters
  - Double buffering



# Last time: programming activity

- Create an image slideshow GUI

- Loads images specified on command line
- Outputs height/width, ignore bogus files
- Cycles between images every second

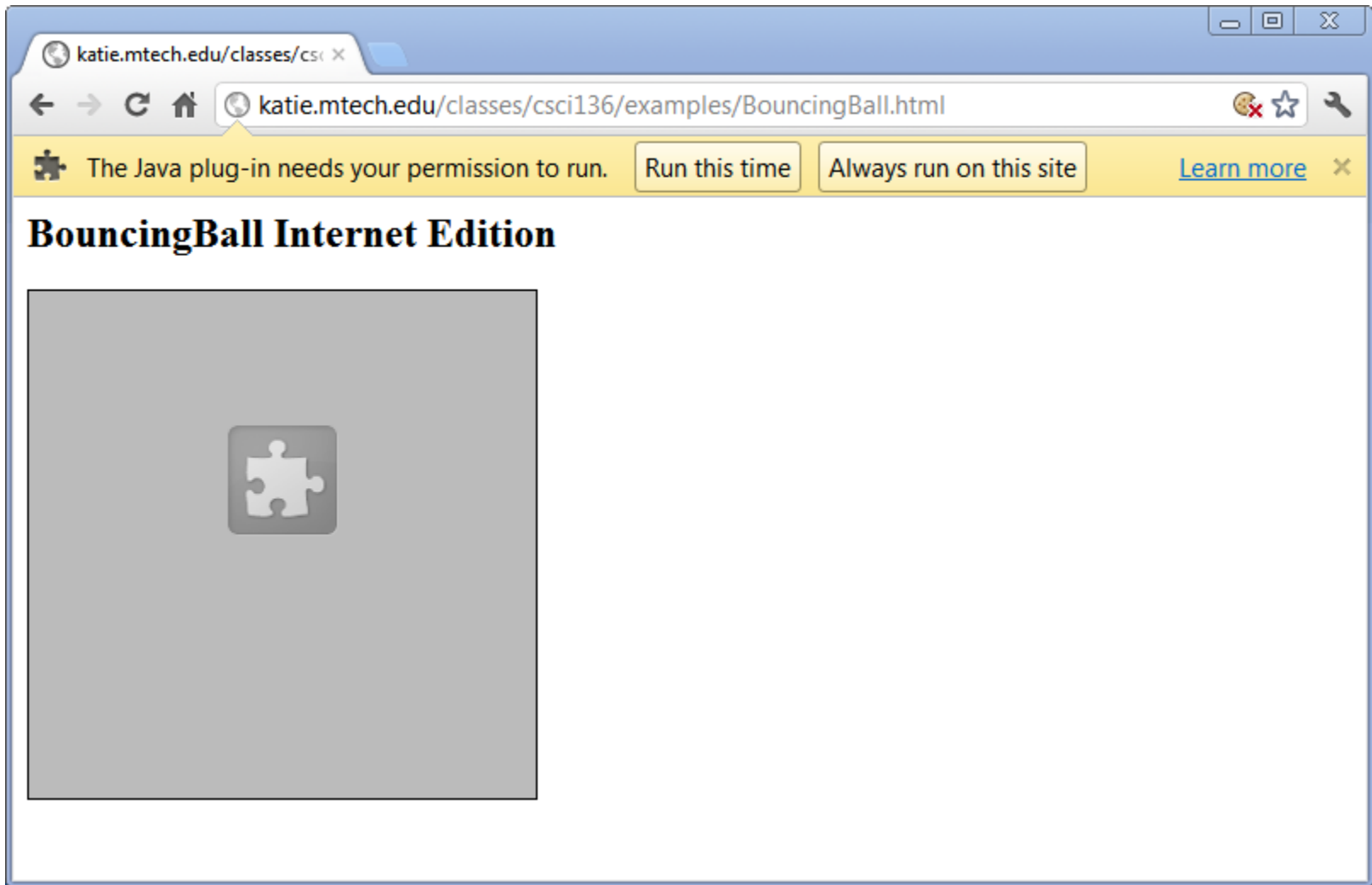
```
% java ShowImages cat.jpg frog.jpg dog.jpg bogus.jpg  
cat.jpg [240 x 160]  
frog.jpg [240 x 118]  
dog.jpg [240 x 228]
```



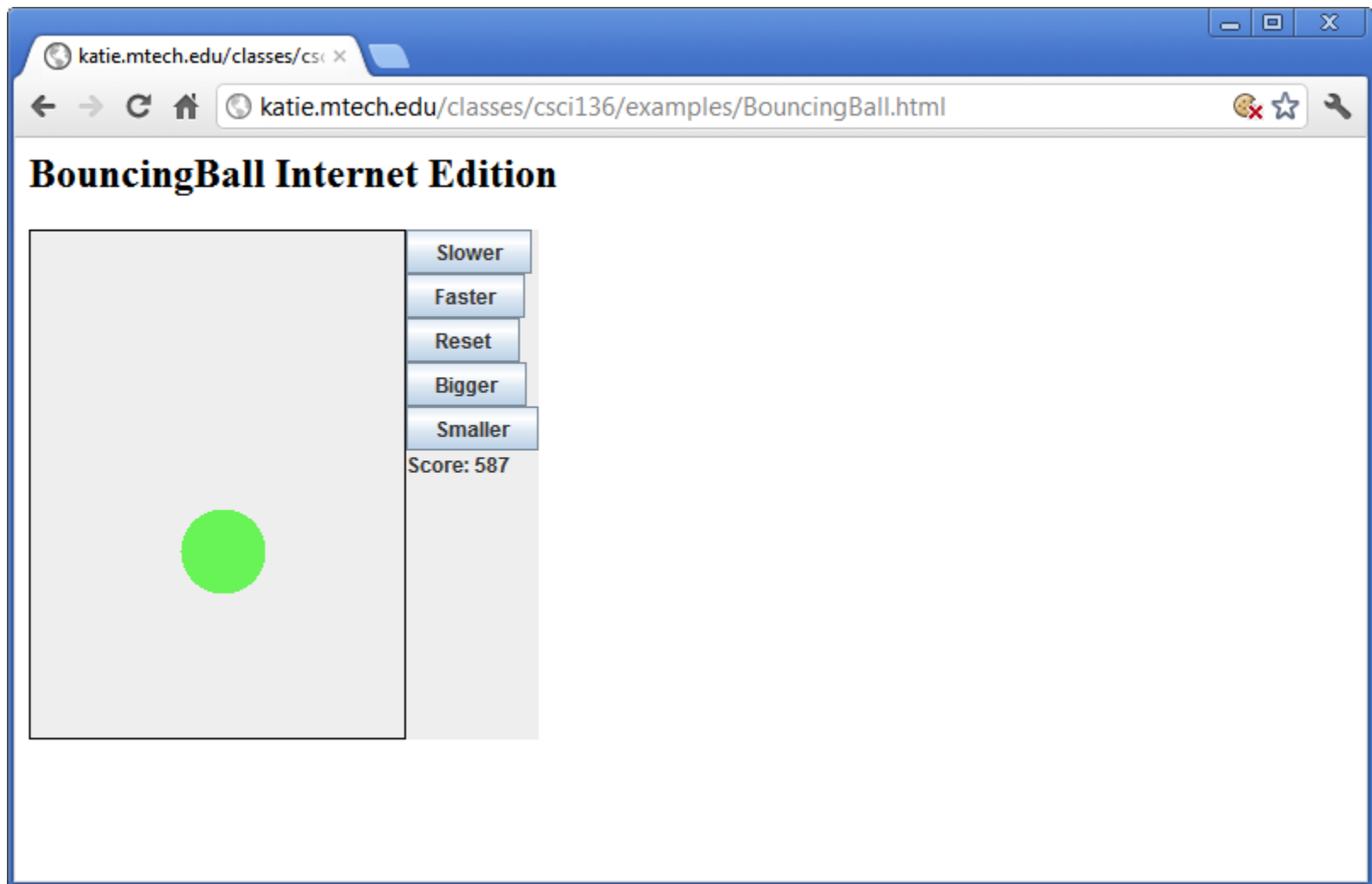
# Applets

- Applets
  - Java programs embedded in a web page
    - `<applet>` tag in the HTML
    - Appears in a box like other page elements (e.g. images)
      - But you can interact with it, pushing buttons, etc.
    - Requires a Java plug-in
    - Not support by some browsers (iPhone)
  - Code very similar to a Java desktop JFrame app
    - Class that extends JApplet
  - Operates in a security "sandbox"
    - Not allowed to access:
      - Local file system, clipboard, arbitrary web sites, etc.

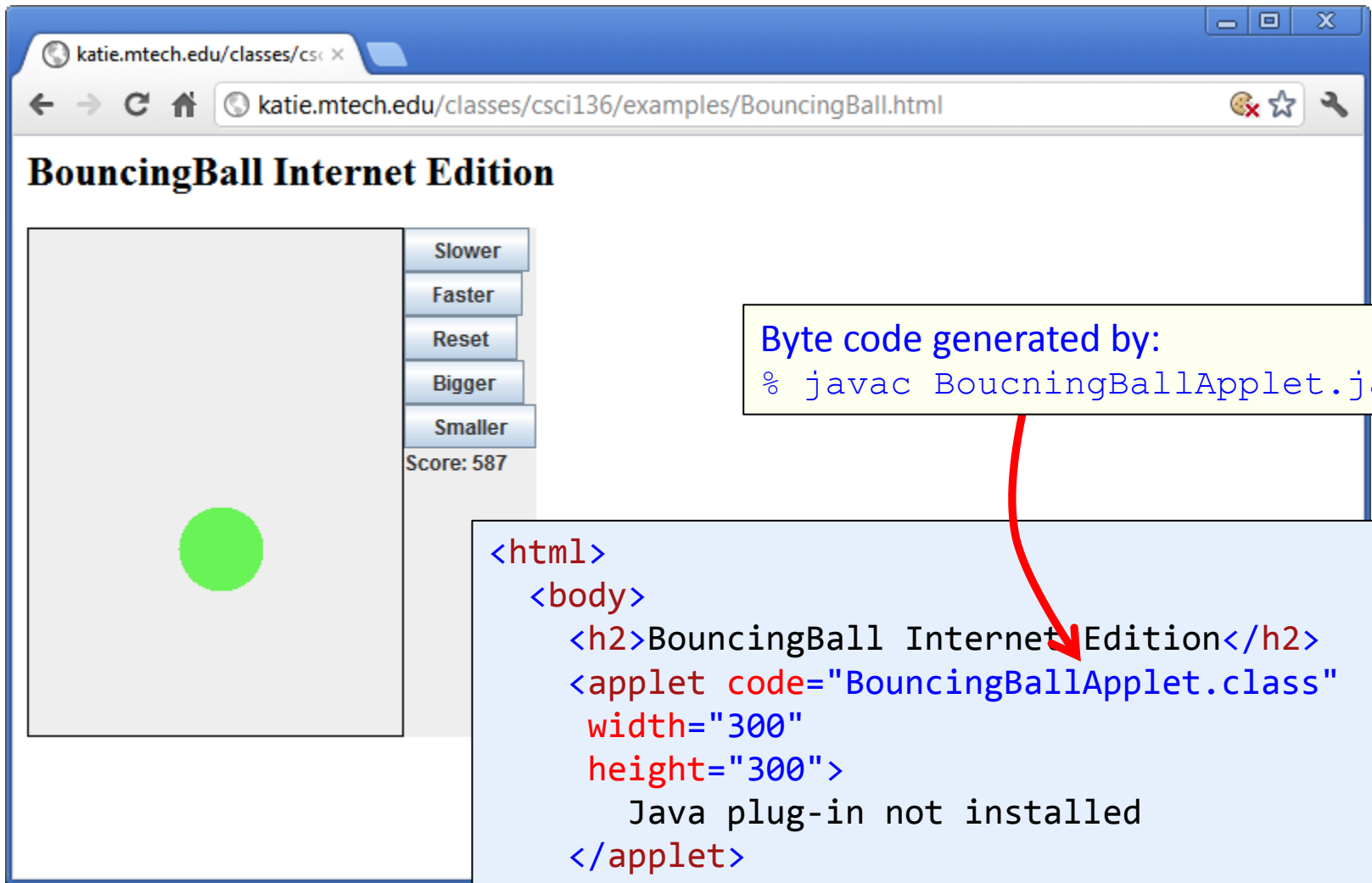
# Example applet: plug-in approval



# Example applet: running



# Example applet: HTML code



Byte code generated by:  
`% javac BoucningBallApplet.java`

```
<html>
  <body>
    <h2>BouncingBall Internet Edition</h2>
    <applet code="BouncingBallApplet.class"
      width="300"
      height="300">
      Java plug-in not installed
    </applet>
  </body>
</html>
```

# JApplet methods

- **Override some JApplet methods**
  - Otherwise you get default implementation from parent

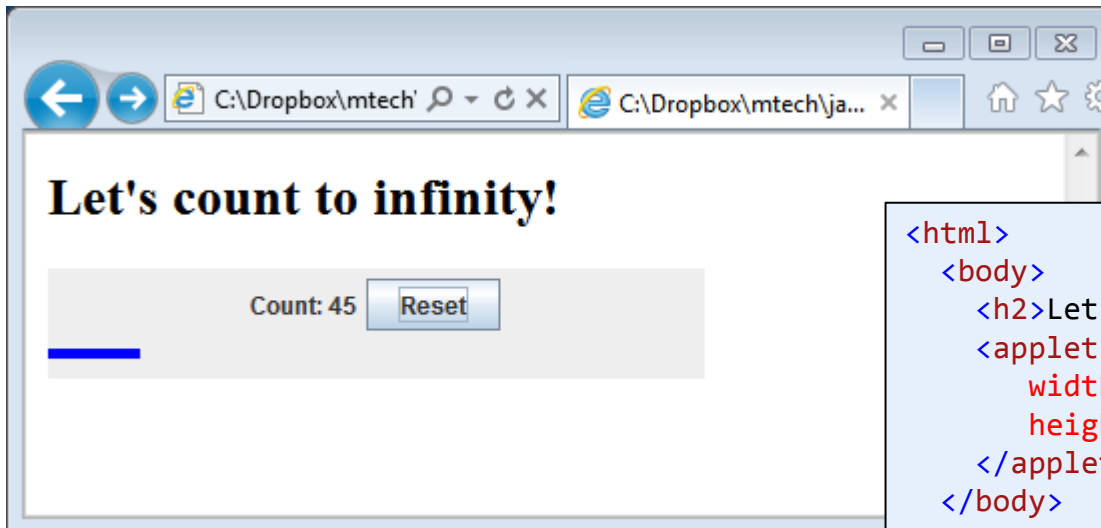
Method	Description
<code>void init()</code>	Called once by the applet container. This should initialize widgets, load images, sounds, threads, etc.
<code>void start()</code>	Called after <code>init</code> completes. Also called whenever user browses to another site and later returns to page. May start animations or threads.
<code>void paint(Graphics g)</code>	Called after <code>init</code> and <code>start</code> . Also whenever window resized, moves, or <code>repaint</code> called.
<code>void stop()</code>	Called when user leaves the applet's web page for another page. May stop animations or threads.
<code>void destroy()</code>	Called when browser closed, cleans up resources.



# Applet example

- Counter applet

- Increase count by 1 every second
  - Create a Thread to manage this
- Reset button sets back to zero
  - Hook up an ActionListener to do reset
- Bar graph showing current count
  - Use paint method to draw directly, no need for a custom JPanel



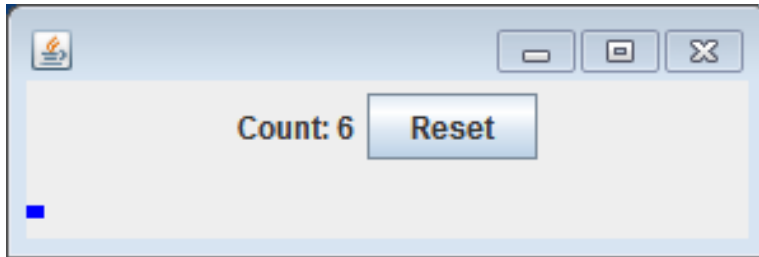
*CounterApplet.java*

```
<html>
  <body>
    <h2>Let's count to infinity!</h2>
    <applet code="CounterApplet.class"
            width="300"
            height="50">Java plugin not installed
    </applet>
  </body>
</html>
```

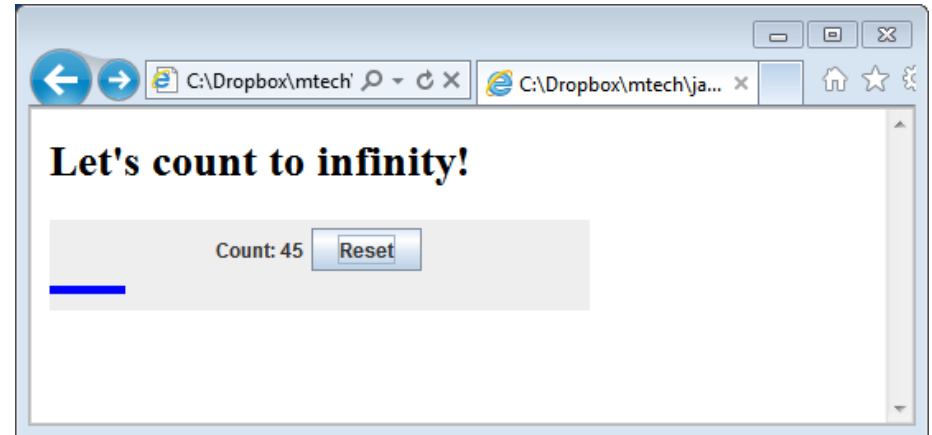
# For comparison...

- Counter application

- Same goal, but as an application extending JFrame
- Code differences:
  - Moved widget setup to constructor
  - Layout setup a little more complicated
    - Must create a custom JPanel1 for the graph
  - Create a main method that created object



*Counter.java*



*CounterApplet.java*



*PanicApplet.java*

# Multimedia applets

- Using images in your applet

- Put in same directory as applet's \*.class file:

```
img = getImage(getDocumentBase(), "dont_panic.png");
```

- Or on a web server somewhere:

```
img = getImage(new URL(  
    "http://katie.mtech.edu/classes/csci136/dont_panic.png"));
```

- Drawing in the paint(Graphics g) method:

```
if (img != null)  
    g.drawImage(img, x, y, this);
```

# Multimedia applets

- Playing sounds in your applet

- Put in same directory as applet's \*.class file:

```
audio = getAudioClip(getDocumentBase(), "boink.wav");
```

- Or on a web server somewhere:

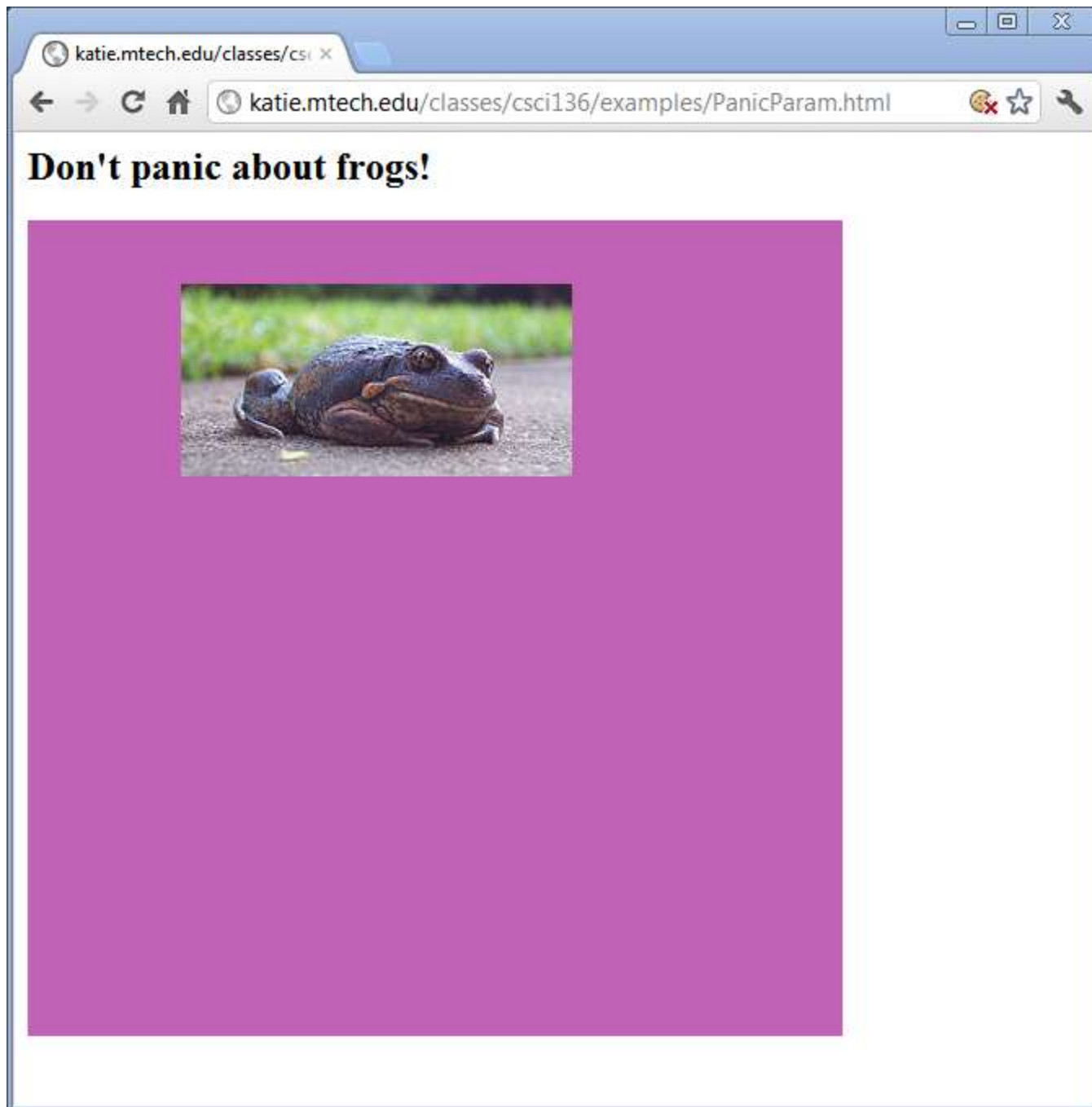
```
img = getAudioClip(  
    "http://katie.mtech.edu/classes/csci136/boink.wav");
```

Method	Description
void play()	Starts playing this audio clip.
void stop()	Stops playing this audio clip.
void loop()	Starts playing this audio clip in a loop.

# Getting input

- Applets may take input
  - Specified as parameters in the HTML
  - Name/value string pairs
    - Parse into other data type as necessary
- Example:
  - Make flashing image/sound applet configurable
  - Different images, sounds, and time delay

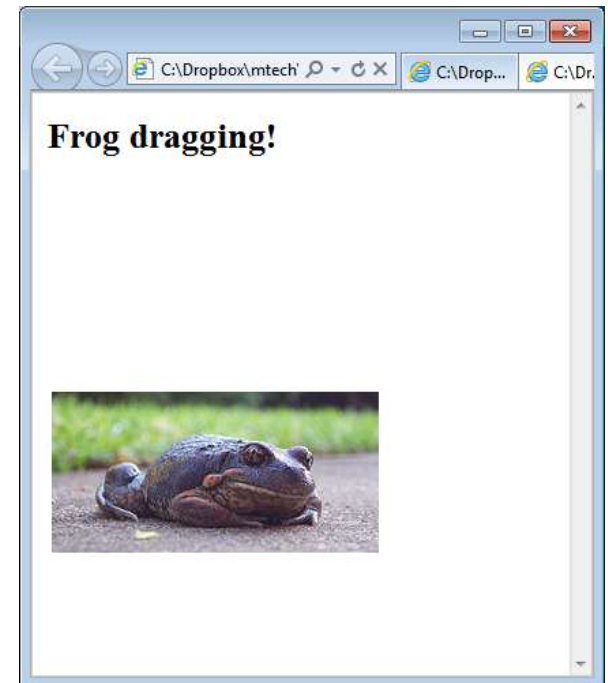
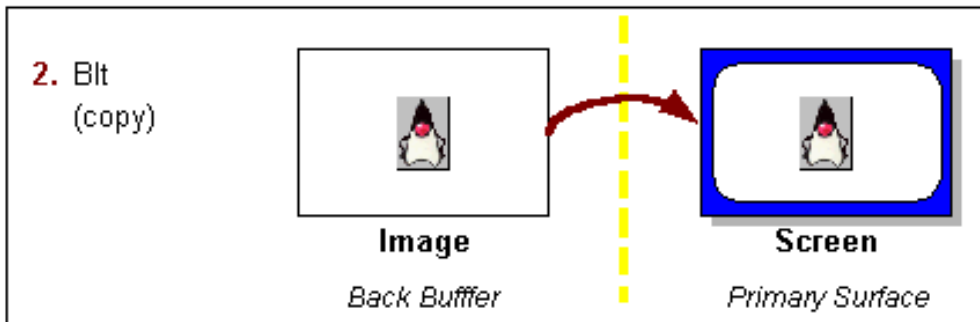
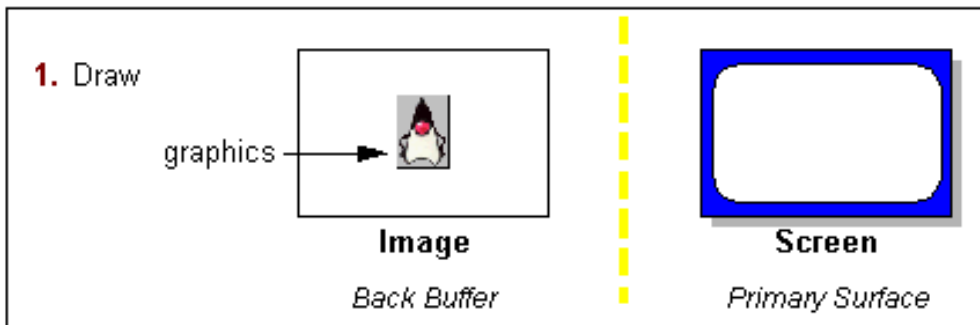
```
<html>
  <body>
    <h2>Don't panic about frogs!</h2>
    <applet code="PanicAppletParam.class" width="500" height="500">
      <param name="image" value="frog.jpg">
      <param name="sound" value="frog.wav">
      <param name="delay" value="500">
      Java plugin not installed
    </applet>
  </body>
</html>
```



# Double buffering

- Problem: PanicApplet flickers a lot
- Solution: Use double buffering
  - Technique that prevents flicker
  - Draw things to a background buffer first
  - Then quick copy to screen

## Double Buffering





```

...
private Image img;
private int x = 0;
private int y = 0;

public void init()
{
    img = getImage(getDocumentBase(),
                  "frog.jpg");
    addMouseListener(this);
}

public void paint(Graphics g)
{
    g.clearRect(0,
               0,
               getWidth(),
               getHeight());
    if (img != null)
        g.drawImage(img, x, y, this);
}
...

```

*DragFrog.java*

```

...
private Image img;
private int x = 0;
private int y = 0;
private Graphics buffer;
private Image offscreen;

public void init()
{
    img = getImage(getDocumentBase(),
                  "frog.jpg");
    addMouseListener(this);

    offscreen = createImage(this.getWidth(),
                           this.getHeight());
    buffer = offscreen.getGraphics();
}

public void paint(Graphics g)
{
    buffer.clearRect(0, 0,
                   getWidth(), getHeight());
    if (img != null)
        buffer.drawImage(img, x, y, this);
    g.drawImage(offscreen, 0, 0, this);
}
...

```

*DragFrogDouble.java*

# Summary

- Java Applets
  - Like a Java GUI application, but inside a web browser
  - Must extend JApplet instead of JFrame
    - Implement some of the methods:
      - init, start, stop, destroy, paint
  - Multimedia applets
    - Showing images, playing audio
  - Getting input using <param> tags
  - Double buffering to prevent flicker