



**BỘ MÔN CÔNG NGHỆ PHẦN MỀM  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



# **LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

## **Bài 9. Applet**

**Nguyễn Thị Thu Trang**

# Nội dung



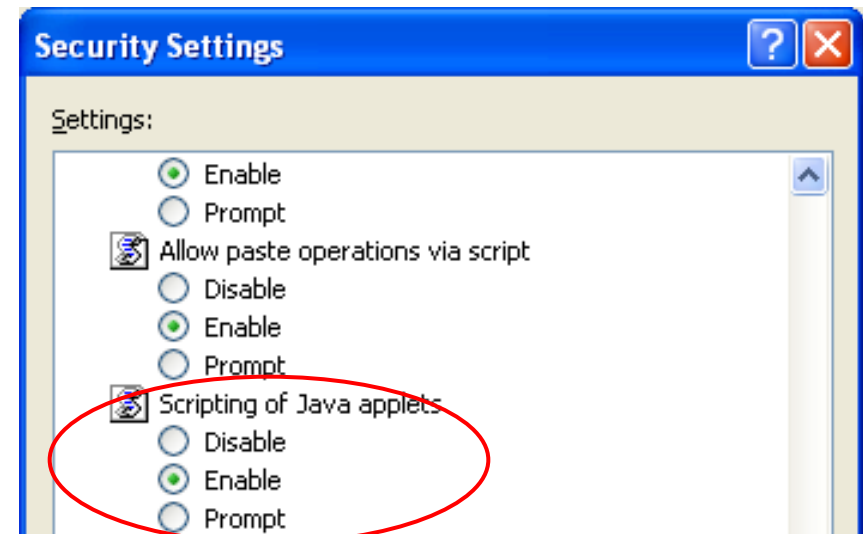
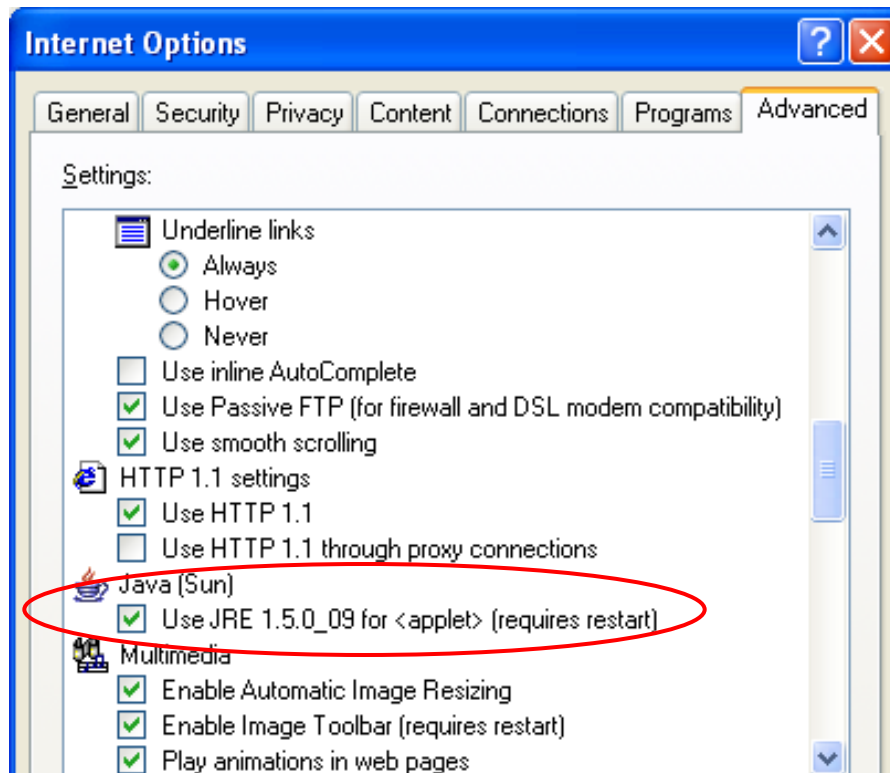
1. Tổng quan về Applet
2. Vòng đời của Applet
3. Lớp Graphics
4. Các phương thức của Applet

# 1. Tổng quan về Applet

- Applet là chương trình Java được nhúng trong mã nguồn HTML.
- Có thể chạy trên trình duyệt Web có cài JVM (JRE) sau khi tải về.
- Nhờ các applet, trang Web có thêm được sức sống vì applet có thể thể hiện multimedia, tạo được sự tương tác với user như hỏi-trả lời
- Có thể nguy hiểm, không đảm bảo tính an toàn

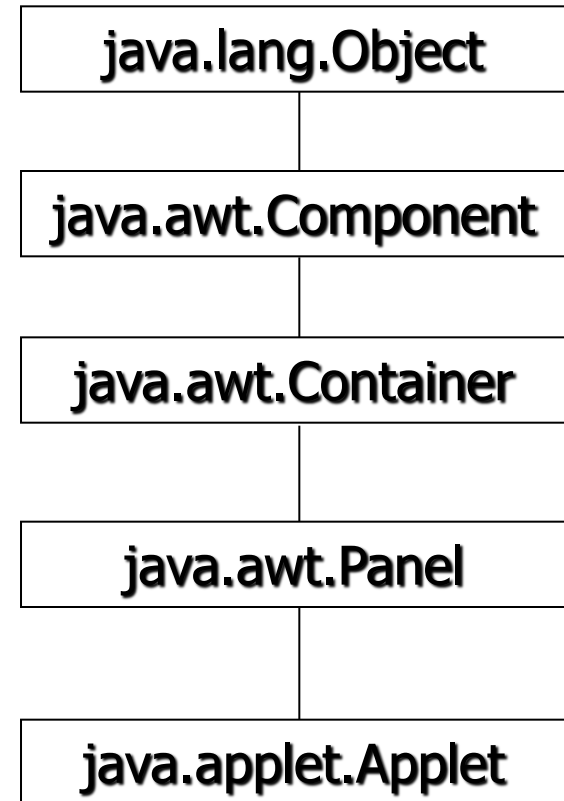
# 1. Tổng quan về Applet

- Để có thể chạy applet, trình duyệt của người dùng phải bật chức năng Java Plug-in



# 1. Tổng quan về Applet

- Để tạo 1 applet, xây dựng 1 lớp con của lớp Applet hoặc lớp JApplet.
  - `import java.applet.Applet;`  
`class MyApplet extends Applet`  
`{...}`  
hoặc
  - `import javax.swing.JApplet;`  
`class MyApplet extends JApplet`  
`{...}`
- Thường đặt tên lớp applet này trùng tên với tên file.html.



# AppletTemplate

```
import java.applet.Applet;
import java.awt.*;
public class AppletTemplate extends Applet
{
    // Khai bao bien
    public void init() {
        // Khoi tao cac bien, tai cac hinh anh...
    }
    public void paint(Graphics g) {
        // Cac thao tac ve (draw)
    }
}
```

# Tạo 1 tài liệu HTML chứa applet

- Để thực thi một applet, tạo một tập tin HTML có sử dụng thẻ applet hoặc thẻ object
  - Thẻ applet/object có hai thuộc tính:
    - Width
    - Height

- Ví dụ:

```
<html>  
  <applet code="AppletTemplate" width=200  
  height=200>  
    </applet>  
</html>
```

# Applet HTML Template

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>A Template for Loading Applets</TITLE>
</HEAD>

<BODY>
<H1>A Template for Loading Applets</H1>
<P>
<APPLET CODE="AppletTemplate.class" WIDTH=320
  HEIGHT=260>
  <B>Error! You must use a Java-enabled browser.</B>
</APPLET>
</BODY>

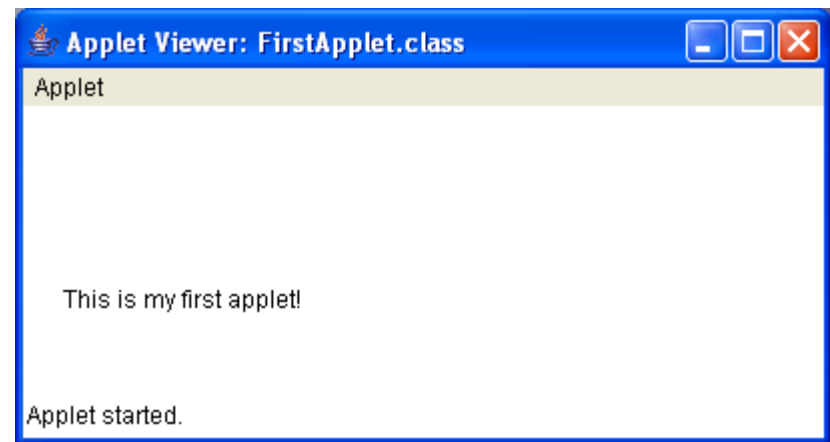
</HTML>
```



# Chương trình Applet đầu tiên

```
import java.applet.Applet;  
import java.awt.Graphics;  
public class FirstApplet extends Applet {  
    public void paint(Graphics g) {  
        g.drawString("This is my first applet!", 20, 100);  
    }  
}
```

appletviewer FirstApplet.java  
hoặc Ctrl + 3 (Textpad)

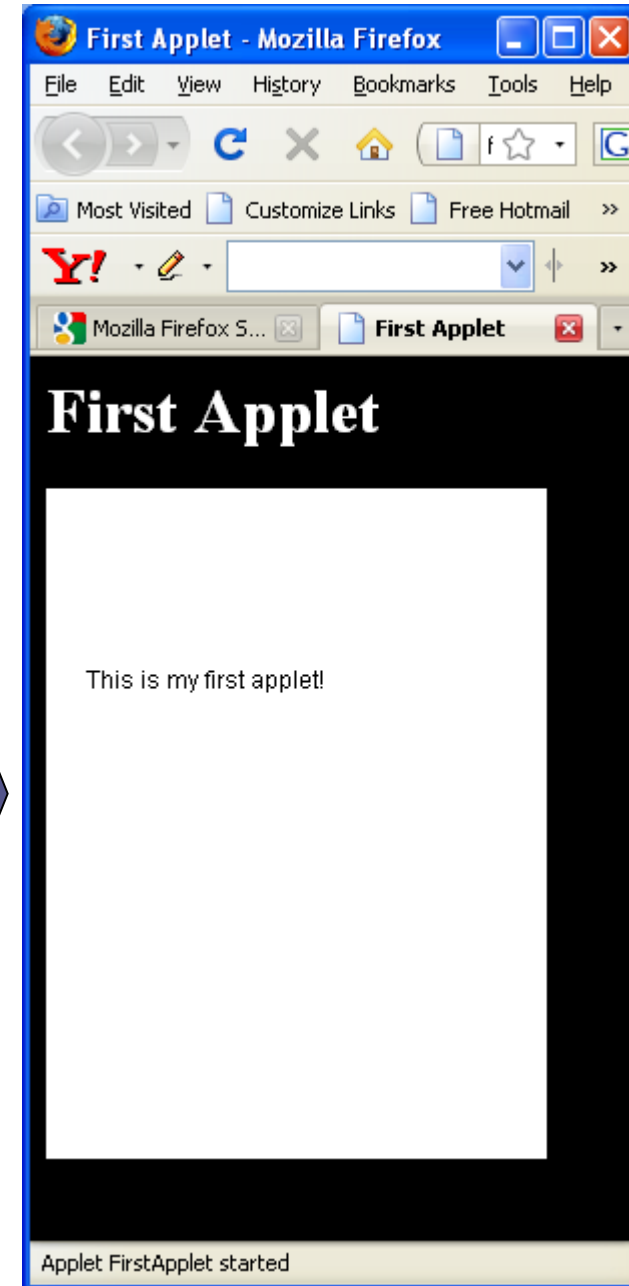
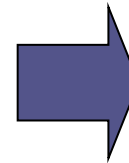


# Chương trình Applet đầu tiên (2)

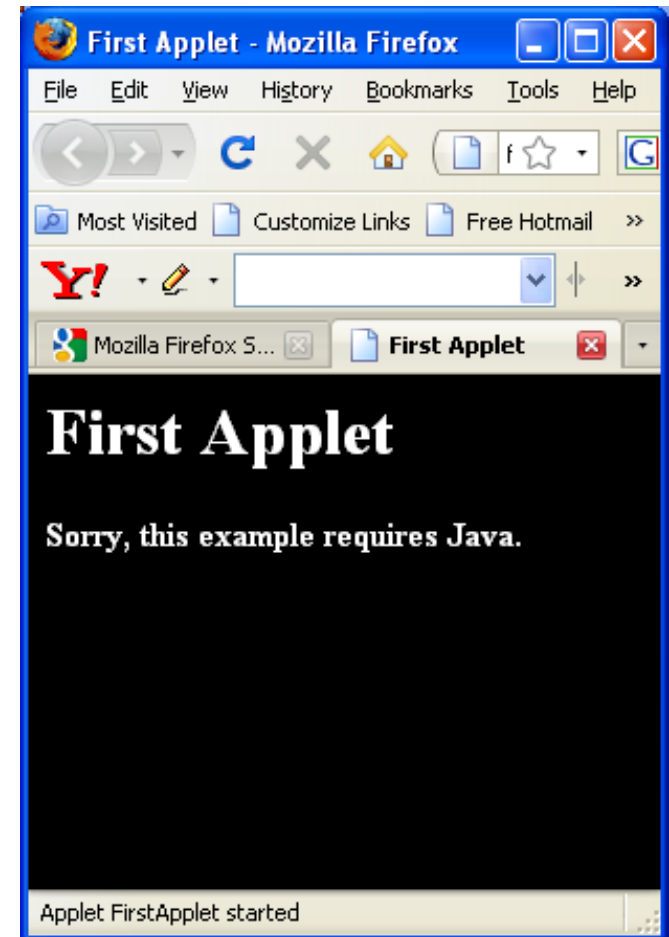
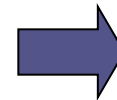
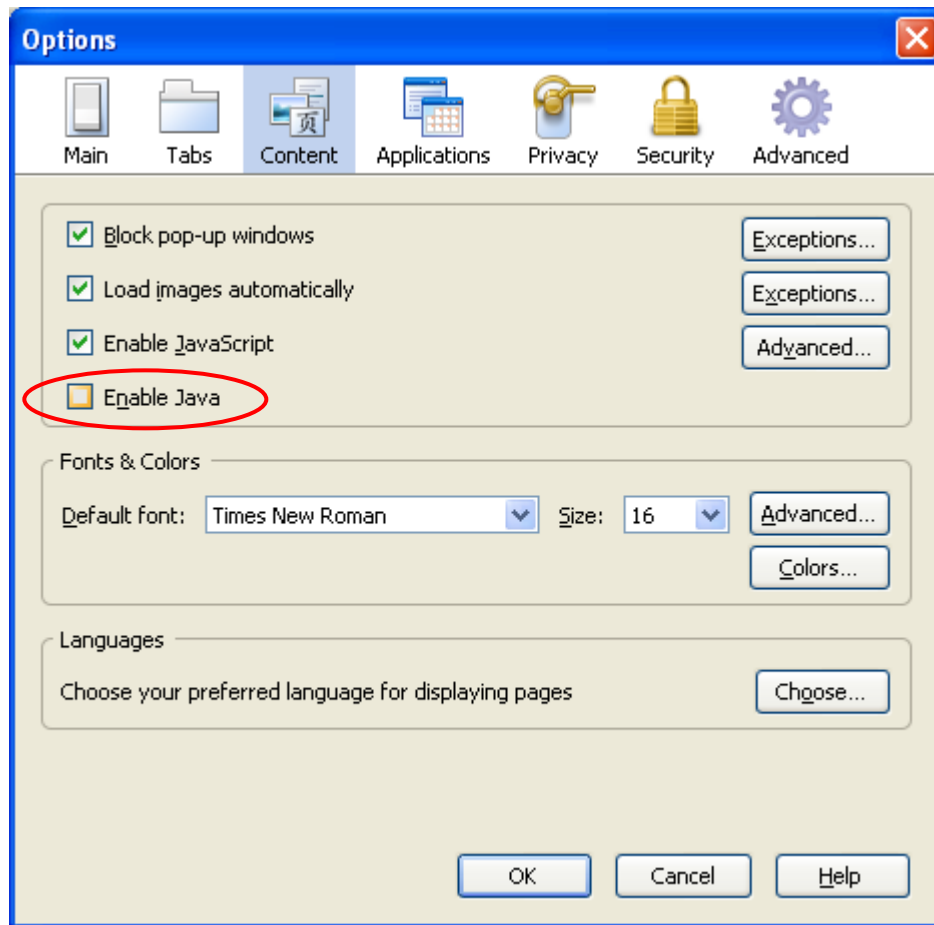
- Kế thừa từ lớp Applet
- Đồ họa từ Abstract Window Toolkit (AWT)
- Chạy sau khi biên dịch:
  - Trên cmd: `appletviewer FirstApplet.html` hoặc `appletviewer FirstApplet.java`
  - Textpad: Ctrl + 3

# Sử dụng .html

```
<HTML>
<HEAD>
  <TITLE>First Applet</TITLE>
</HEAD>
<BODY BGCOLOR="BLACK" TEXT="WHITE">
<H1>First Applet</H1>
<P>
<APPLET CODE="FirstApplet.class"
        WIDTH=250
        HEIGHT=335>
  <B>Sorry, this example
        requires Java.</B>
</APPLET>
</BODY>
</HTML>
```



# Mozilla Firefox

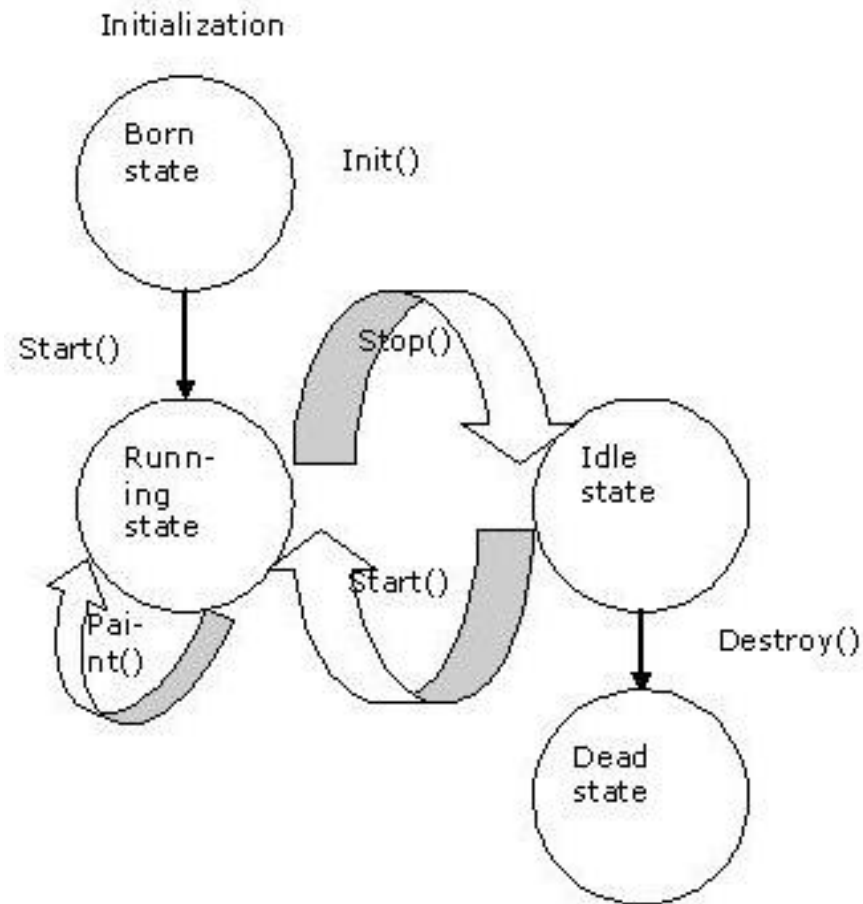


# Nội dung

1. Tổng quan về Applet
- ⇒ 2. Vòng đời của Applet
3. Lớp Graphics
4. Các phương thức của Applet

## 2. Vòng đời của Applet

- **init()** : Chỉ được gọi khi applet bắt đầu thực thi
- **start()** : Được gọi sau init() hoặc khi applet được trở lại
- **paint()** : Được gọi sau khi trình duyệt init và start, và được gọi lại khi trình duyệt được vẽ lại
- **stop()** : Được gọi khi applet được thu nhỏ
- **destroy()** : Được gọi khi applet kết thúc (tắt trình duyệt)

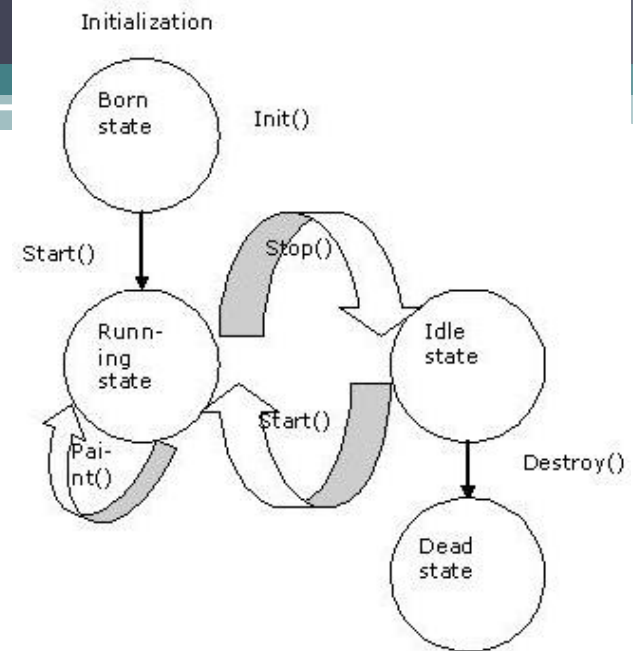


# Ví dụ

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

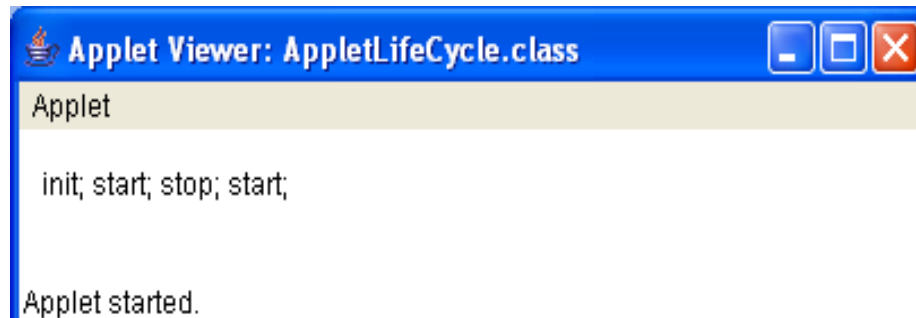
public class AppletLifeCycle
    extends Applet {
    String str = "";

    public void init() {
        str += "init; ";
    }
    public void start() {
        str += "start; ";
    }
    public void stop() {
        str += "stop; ";
    }
}
```



```
public void destroy() {
    System.out.println("destroy");
}

public void paint(Graphics g) {
    g.drawString(str, 10, 25);
}
```



# Nội dung

1. Tổng quan về Applet
2. Vòng đời của Applet
- ⇒ 3. Lớp Graphics
4. Các phương thức của Applet



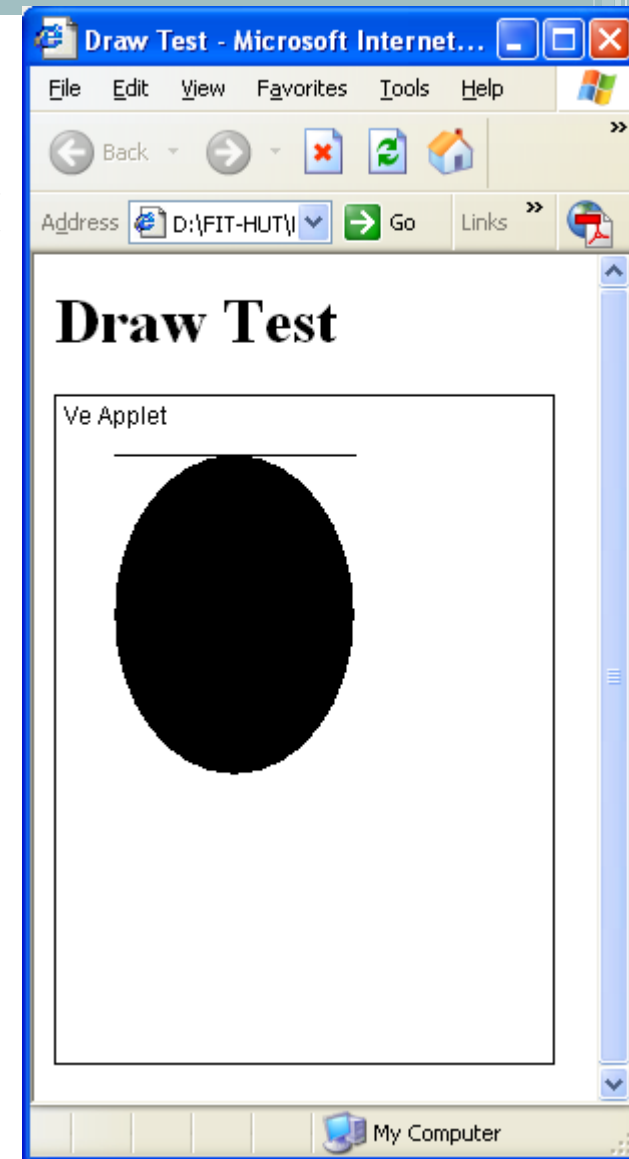
# Các phương thức trong Graphics

- **drawString(string, left, bottom)**
  - Vẽ một xâu với font chữ và màu hiện tại với góc trái (left) và đáy (bottom) của xâu được chỉ ra
- **drawRect(left, top, width, height)**
  - Vẽ đường viền hình chữ nhật (1-pixel border) với màu hiện tại
- **fillRect(left, top, width, height)**
  - Vẽ hình chữ nhật đặc với màu hiện tại
- **drawLine(x1, y1, x2, y2)**
  - Vẽ một đường với độ dày 1 pixel từ tọa độ (x1, y1) đến tọa độ (x2, y2)

# Các phương thức trong Graphics

- **drawOval, fillOval**
  - Vẽ hình elip (viền/đặc) với các tham số chỉ ra là hình chữ nhật bao quanh elip (oval)
- **drawPolygon, fillPolygon**
  - Vẽ hình đa giác (viền/đặc) với các điểm được xác định bởi các mảng hoặc một đối tượng `Polygon` (một lớp lưu một loạt các điểm) được chỉ ra trong tham số
  - Mặc định hình đa giác được vẽ khép kín, để vẽ hình đa giác mở sử dụng phương thức `drawPolyline`
- **drawImage**
  - Vẽ một hình ảnh
  - Hình ảnh có thể có định dạng JPEG hoặc GIF (gồm GIF89A)

```
import java.applet.Applet;  
import java.awt.Graphics;  
  
public class DrawTest extends Applet{  
    public void paint(Graphics g){  
        //Ve hcn xq vung hien thi applet  
        g.drawRect(0, 0,  
            getWidth()-1,getHeight()-1);  
  
        //Ve xau trong hcn  
        g.drawString("Ve Applet", 5, 15);  
  
        g.fillOval(30, 30, 120, 160);  
  
        g.drawLine(30, 30, 150, 30);  
    }  
}
```

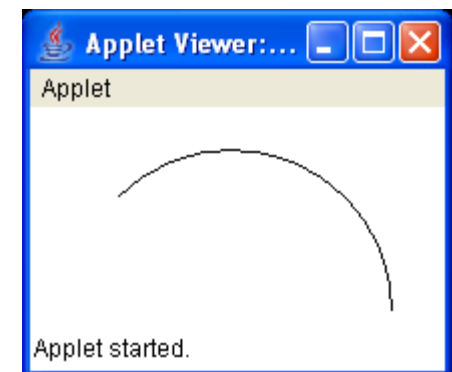


# Các phương thức trong Graphics

- `drawArc(int x, int y, int w, int h, int degrees0, int degrees1)`
- `fillArc(int x, int y, int w, int h, int degree0, int degree1)`
- `drawPolygon(int x[], int u[], int n)`
- `fillPolygon(int x[], int y[], int n)`
- `drawPolyline(int x[], int y[], int n)`
- `Color getColor()`
- `Font getFont()`
- `setFont(Font f)`
- ...

# Ví dụ

```
import java.applet.Applet;  
import java.awt.Graphics;  
  
public class DrawArc extends Applet {  
  
    public void paint(Graphics g) {  
        g.drawArc(20, 20, 160, 160, 0, 135);  
    }  
}
```



# Graphics Font

- `setFont, getFont`
  - Chỉ ra font được sử dụng để vẽ văn bản
  - Xác định kích thước của ký tự thông qua `FontMetrics` (trong Java 2 sử dụng `LineMetrics`)
  - Thiết lập font cho đối tượng `Graphics` không tồn tại qua các lời gọi kế tiếp của `paint`
  - Thiết lập font của window (ví dụ gọi phương thức `setFont`) cho việc thay đổi lâu dài cho đối tượng `Graphics`.

# Graphics Color

- setColor, getColor
  - Chỉ ra màu mặt trước (foreground color) cho các thao tác vẽ
  - Mặc định, đối tượng Graphics nhận màu mặt trước từ màu của window.
  - AWT định nghĩa 16 màu (`Color.red`, `Color.blue`...) hoặc bạn có thể tạo ra màu của riêng mình bằng `new Color(r, g, b)`
  - Thay đổi màu của đối tượng Graphics chỉ ảnh hưởng đến đối tượng Graphics tương ứng
    - Để thay đổi màu lâu dài, gọi phương thức `setForeground` của applet

# Ví dụ

```
import java.applet.Applet;  
import java.awt.Color;  
import java.awt.Graphics;  
import java.awt.Font;  
  
public class BlueString extends Applet {  
  
    public void paint(Graphics g) {  
        g.setFont(new Font("SansSerif", Font.BOLD, 30));  
        g.setColor(Color.blue);  
        g.drawString("Blue String", 100, 50);  
    }  
}
```





# Tải một hình ảnh

- Đăng ký Image (trong phương thức `init`)

```
Image image = getImage(getCodeBase(), "file");  
Image image = getImage(url);
```

- Nếu URL là tuyệt đối, cần sử dụng `try/catch`

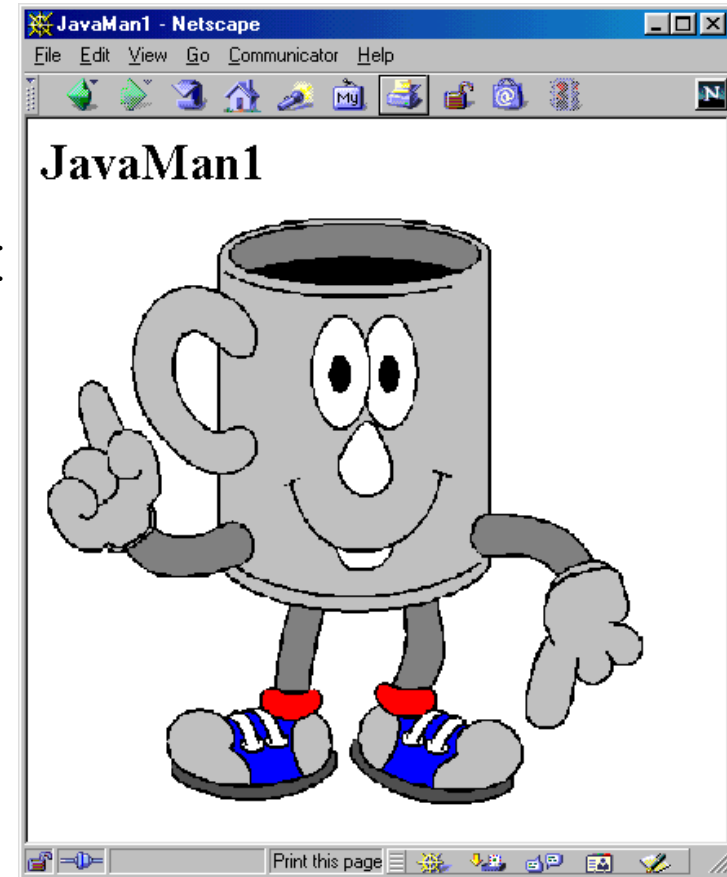
- Vẽ một hình ảnh (trong phương thức `paint`)

```
g.drawImage(image, x, y, window);  
g.drawImage(image, x, y, w, h, window);
```

- Có thể vẽ một phần của hình ảnh
- Sử dụng applet (**this**) cho tham số `window`

# Tải một hình ảnh từ URL tương đối

```
import java.applet.Applet;  
import java.awt.*;  
/** An applet that loads an image  
    from a relative URL. */  
public class JavaMan1 extends Applet {  
    private Image javaMan;  
    public void init() {  
        javaMan =  
            getImage(getCodeBase(),  
                    "images/Java-Man.gif");  
    }  
    public void paint(Graphics g) {  
        g.drawImage(javaMan, 0, 0, this);  
    }  
}
```



# Tải một hình ảnh từ URL tuyệt đối

```
import java.applet.Applet;
import java.awt.*;
import java.net.*;
...
private Image javaMan;
public void init() {
    try {
        URL imageFile =
            new URL("http://www.corewebprogramming.com" +
                    "/images/Java-Man.gif");
        javaMan = getImage(imageFile);
    } catch (MalformedURLException mue) {
        showStatus("Bogus image URL.");
        System.out.println("Bogus URL");
    }
}
```

# Nội dung

1. Tổng quan về Applet
2. Vòng đời của Applet
3. Lớp Graphics
- ⇒ 4. Các phương thức của Applet

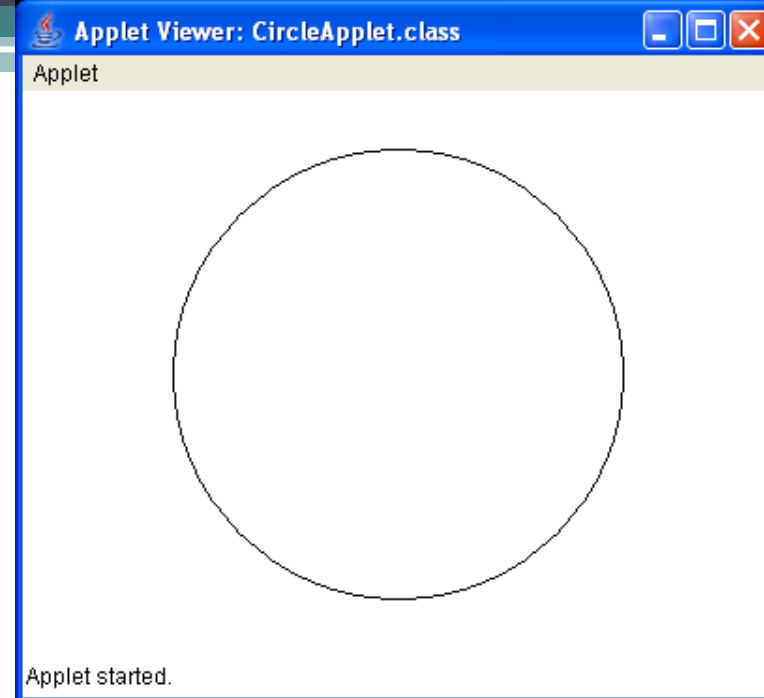
# Các phương thức khác của Applet

- `getCodeBase`, `getDocumentBase`
  - Lấy về URL của:  
Applet file - `getCodeBase`  
HTML file - `getDocumentBase`
- `getParameter(String name)`
  - Lấy về giá trị từ phần tử `PARAM` trong HTML file
- `getSize()`
  - Trả về **Dimension** (width, height) của applet
- `getGraphics()`
  - Lấy về đối tượng **Graphics** hiện tại của applet
  - Đối tượng **Graphics** không tồn tại giữa các lần gọi `paint()`

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

public class Circle extends Applet{

    public void paint(Graphics g) {
        Dimension d = getSize();
        int xc = d.width / 2;
        int yc = d.height / 2;
        int radius = (int) ((d.width < d.height) ?
                               0.4 * d.width : 0.4 * d.height);
        g.drawOval(xc - radius, yc - radius,
                   2 * radius, 2 * radius);
    }
}
```



# Các phương thức khác (2)

- `showDocument()`

```
getAppletContext().showDocument(java.net.URL url)
getAppletContext().showDocument(java.net.URL url, String
                                targetWindow)
```

Hiển thị một tài liệu trên trình duyệt

- `showStatus(String str)`

- Hiển thị một xâu trong dòng trạng thái cuối trình duyệt

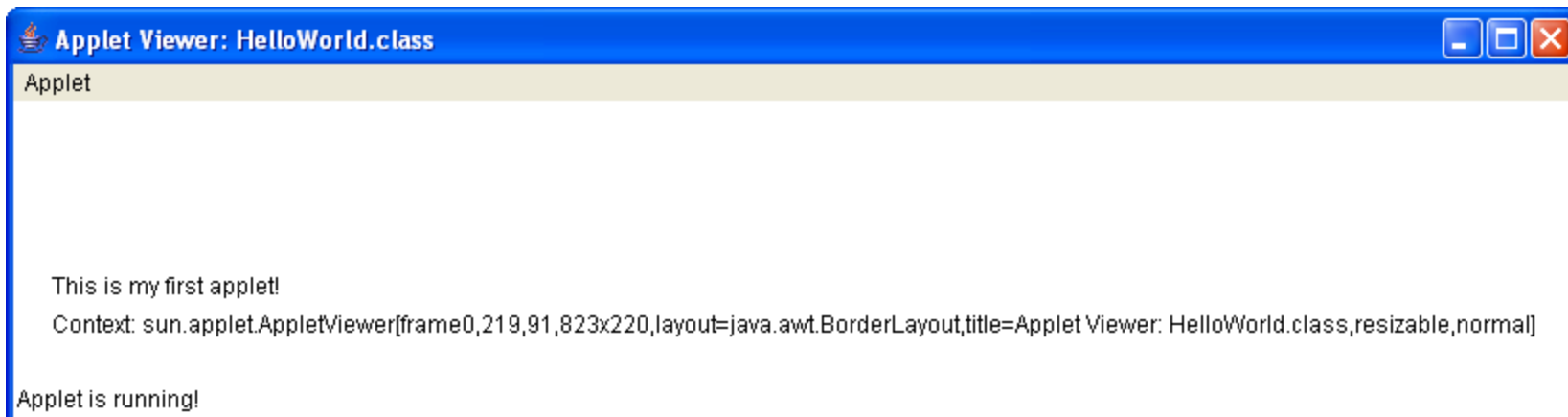
- `Cursor getCursor(), setCursor(Cursor cur)`

- Lấy về hoặc thiết lập `Cursor` cho chuột, ví dụ như `CROSSHAIR_CURSOR`, `HAND_CURSOR`, `WAIT_CURSOR`



# Ví dụ

```
import java.applet.Applet;
import java.awt.Graphics;
public class TestApplet extends Applet {
    public void paint(Graphics g) {
        g.drawString("This is my first
applet!",
                                20, 100);
        showStatus("Applet is running!");
        g.drawString("Context: " +
                                getAppletContext(), 20, 120);
    }
}
```



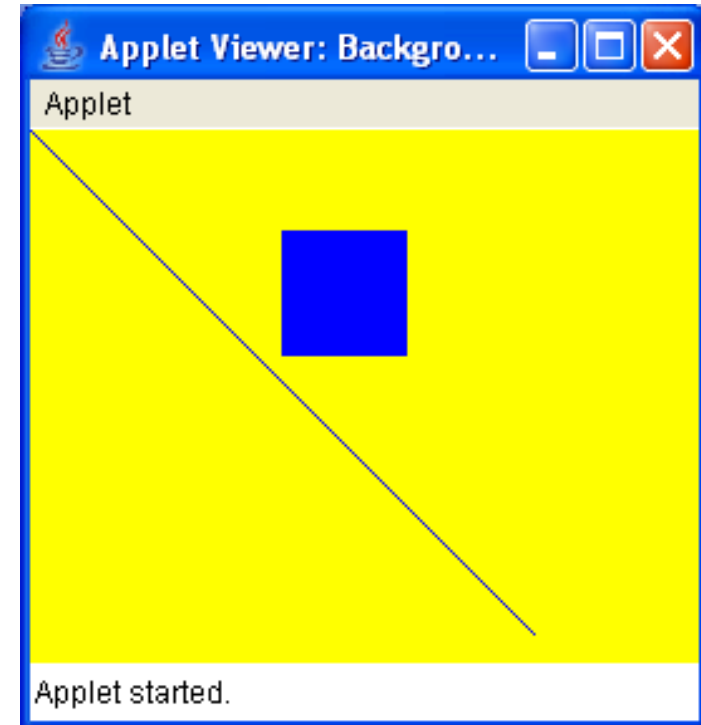


# Các phương thức khác (3)

- **getAudioClip, play**
  - Lấy về tệp âm thanh từ một vị trí từ xa và chơi nó
  - JDK 1.1 chỉ hỗ trợ .au. Java 2 còn hỗ trợ thêm MIDI, .aiff và .wav
- **setBackground, setBackground**
  - Lấy về/thiết lập màu nền cho applet
  - Lớp `SystemColor` cung cấp truy cập đến các màu của màn hình nền
- **getForeground, setForeground**
  - Lấy về/thiết lập màu vẽ của applet (màu mặc định cho các thao tác vẽ - draw)

# Ví dụ

```
import java.applet.*;  
import java.awt.*;  
public class BackgroundForeground extends Applet {  
  
    public void paint(Graphics g) {  
        setBackground(Color.yellow);  
        setForeground(Color.blue);  
        g.drawLine(0, 0, 200, 200);  
        g.fillRect(100, 40, 50, 50);  
    }  
}
```



# Phần tử APPLET trong HTML file

```
<applet code="..." width=xxx height=xxx  
...>
```

```
...  
</applet>
```

- Các thuộc tính bắt buộc
  - **CODE**
    - Chỉ ra tên của lớp Java cần tải
    - Tên tệp được thông dịch với thư mục của trang HTML hiện tại (mặc định) trừ khi CODEBASE được cung cấp
  - **WIDTH và HEIGHT**
    - Chỉ ra vùng mà applet sẽ hiển thị
    - Các giá trị được đưa ra theo pixel hoặc theo phần trăm của cửa sổ trình duyệt

# Phần tử APPLET trong HTML file (2)

- Các thuộc tính khác
  - ALIGN, HSPACE, và VSPACE
    - Kiểm soát vị trí và khoảng cách với viền theo pixel
  - ARCHIVE
    - Chỉ ra các tệp JAR (tệp zip với phần mở rộng .jar) chứa tất cả các lớp và hình ảnh được sử dụng bởi applet
    - Tiết kiệm đáng kể thời gian tải về nhiều tệp class
  - NAME
    - Tên của applet dùng cho việc liên kết với các applet khác hoặc với JavaScript
  - MAYSCRIPT (không chuẩn)
    - Cho phép JavaScript điều khiển applet

# Ví dụ: Thiết lập thuộc tính Param

```
<h1>Customizable HelloWWW Applet</h1>
```

```
<applet code="HelloWWW2.class" width=400 height=40>  
  <param name="BACKGROUND" value="LIGHT">  
  <b>Error! You must use a Java-enabled browser.</b>  
</applet>
```

```
<applet code="HelloWWW2.class" width=400 height=40>  
  <param name="BACKGROUND" value="DARK">  
  <b>Error! You must use a Java-enabled browser.</b>  
</applet>
```

```
<applet code="HelloWWW2.class" width=400 height=40>  
  <b>Error! You must use a Java-enabled browser.</b>  
</applet>
```

# Applet HelloWorld2 đọc các tham số

- Sử dụng `getParameter(name)` để lấy về giá trị của phần tử PARAM, tên của tham số có phân biệt chữ hoa chữ thường

```
import java.applet.Applet;
import java.awt.*;
public class HelloWorld2 extends Applet {
    public void init() {
        setFont(new Font("SansSerif", Font.BOLD, 30));
        Color background = Color.GRAY;
        Color foreground = Color.DARK_GRAY;
        String backgroundType = getParameter("BACKGROUND");
```

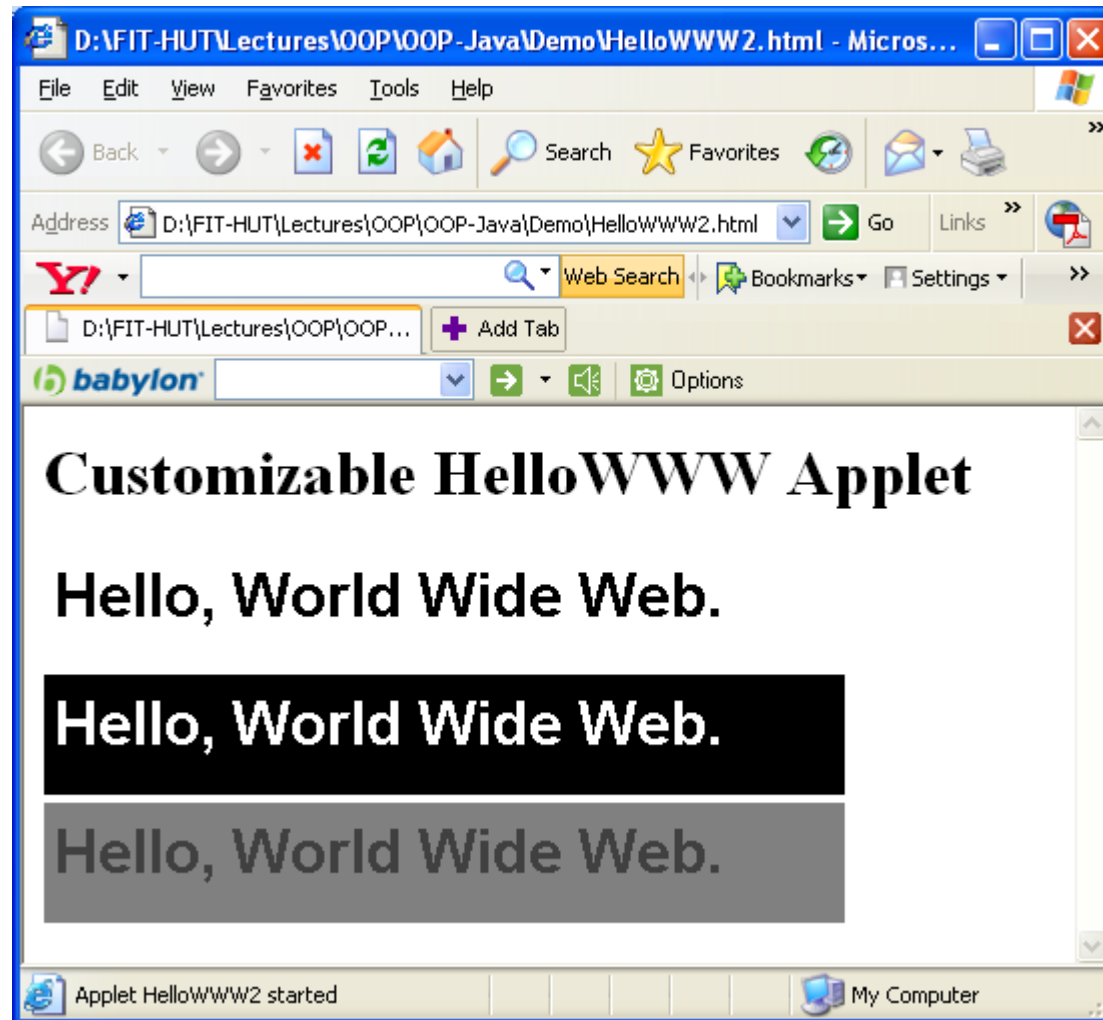
# Applet HelloWWW2 đọc các tham số

```
if (backgroundType != null) {
    if (backgroundType.equalsIgnoreCase("LIGHT")) {
        background = Color.WHITE;
        foreground = Color.BLACK;
    } else if (backgroundType.equalsIgnoreCase("DARK")) {
        background = Color.BLACK;
        foreground = Color.WHITE;
    }
}

setBackground(background);
setForeground(foreground);
}

public void paint(Graphics g) {
    g.drawString("Hello, World Wide Web.", 5, 35);
}
}
```

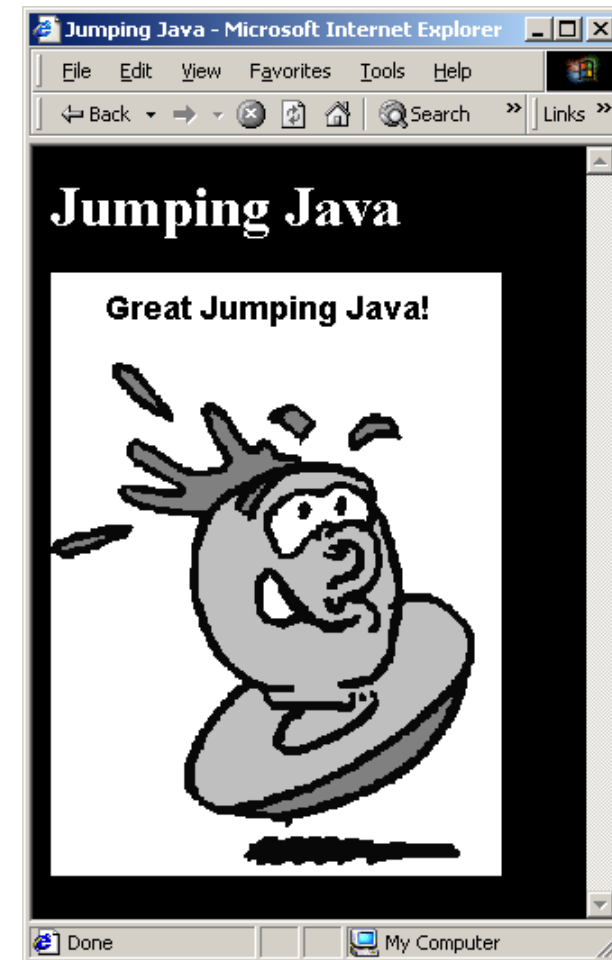
# Lấy tham số từ HTML – Kết quả





# Sử dụng JavaJump.html

```
<HTML>
<HEAD>
  <TITLE>Jumping Java</TITLE>
</HEAD>
<BODY BGCOLOR="BLACK" TEXT="WHITE">
<H1>Jumping Java</H1>
<P>
<APPLET CODE="JavaJump.class"
  WIDTH=250
  HEIGHT=335>
  <param name="image" value="jumping.gif">
  <B>Sorry, this example requires
  Java.</B>
</APPLET>
</BODY>
</HTML>
```



# JavaJump.java

```
import java.applet.Applet;
import java.awt.*;
/** An applet that draws an image. */
public class JavaJump extends Applet{
    private Image jumpingJava;
    public void init() {
        setBackground(Color.white);
        setFont(new Font("SansSerif", Font.BOLD, 18));
        jumpingJava = getImage(getDocumentBase(),
                               getParameter("image"));
        add(new Label("Great Jumping Java!"));
        System.out.println("Yow! I'm jiving with
Java.");
    }

    public void paint(Graphics g) {
        g.drawImage(jumpingJava, 0, 50, this);
    }
}
```

