APPLET

Gianna Bellè

Aprile 1999

APPLICAZIONE

Che cosa è un'applicazione

è un programma java *stand-alone*

```
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello, world.");
  }
}
```

Compilazione: `javac HelloWorld.java`

Running: `java HelloWorld.class`

Output: `Hello, world.`

Che cosa è un applet

è un particolare tipo di programma Java,
inteso per essere incorporato in una pagina Web

- richiede un browser Java-abilitato (versioni di Java includono appletviewer, un mini-browser che ignora HTML eccetto parte relativa agli applets)
- gira *localmente* (sulla macchina client su cui il Web browser sta girando)
- è vincolato a criteri di sicurezza (no accesso al file system locale, no connessioni di rete con host diversi da quello di provenienza, no lancio di altri programmi)

Creazione di un applet

1. creare la classe Java:
   definisce il comportamento dell'applet
2. creare il documento HTML associato:
   associa all'applet una particolare regione rettangolare della pagina Web

Java Template

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

public class AppletTemplate extends Applet {

  // Dichiarazioni di variabili

  public void init() {
    //Inizializzazione di variabili,
    //caricamento immagini, etc.
  }

  public void paint(Graphics g) {
    // Operazioni di disegno
  }
}
```

HTML Template

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
  <TITLE>Un Template per Caricare Applets</TITLE>
</HEAD>

<BODY>
<H1>Un Template per Caricare Applets</H1>
<P>
<OBJECT CODETYPE="application/java"
        CLASSID="java:AppletTemplate.class"
        WIDTH=120 HEIGHT=60>
  <B>Errore! Devi usare un browser abilitato Java.</B>
</OBJECT>

</BODY>
</HTML>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
...
<APPLET CODE="AppletTemplate.class" WIDTH=120 HEIGHT=60>
  <B>Errore! Devi usare un browser abilitato Java.</B>
</APPLET>
...
```


Esempio

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">

<!-- This appears in Core Web Programming from        -->
<!-- Prentice Hall Publishers, and may be freely used -->
<!-- or adapted. 1997 Marty Hall, hall@apl.jhu.edu.   -->

<HTML>
<HEAD>
  <TITLE>Jumping Java</TITLE>
</HEAD>

<BODY BGCOLOR="BLACK" TEXT="WHITE">
<H1>Jumping Java</H1>
<P>
<APPLET CODE="JavaJump.class" WIDTH=400 HEIGHT=500>
  <B>Sorry, this example requires Java.</B>
</APPLET>

</BODY>
</HTML>
```

Esempio

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

// This appears in Core Web Programming from
// Prentice Hall Publishers, and may be freely used
// or adapted. 1997 Marty Hall, hall@apl.jhu.edu.

/** An applet that draws an image. */

public class JavaJump extends Applet {
  private Image jumpingJava;

  public void init() {
    setBackground(Color.white);
    setFont(new Font("Helvetica", Font.BOLD, 18));
    jumpingJava = getImage(getDocumentBase(),
                           "images/Jumping-Java.gif");
    add(new Label("Great Jumping Java!"));
    System.out.println("Yow! I´m jiving with Java.");
  }

  public void paint(Graphics g) {
    g.drawImage(jumpingJava, 0, 25, this);
  }
}
```


Applet

Ciclo di vita

public void init() Chiamato appena il browser crea l'istanza dell'applet. Non viene più chiamato a meno che non sia stato cancellato dalla memoria dal browser.

public void start() Chiamato dopo `init` ma prima della prima invocazione di `paint`. Se l'applet viene fermato e fatto ripartire, `start` viene chiamato di nuovo.

public void paint(Graphics g) Questo è il posto dove l'utente mette il disegno. E' invocato dopo il completamento di `init` e `start` e quando il browser pensa sia necessario che lo schermo venga ridisegnato. Si puo' chiamare `repaint` da un altro metodo per dire al browser ti chiamare `paint` appena può.

public void stop() Chiamato quando l'utente lascia la pagina o minimizza il browser.

public void destroy() Chiamato quando l'applet sta per essere permanentemente distrutto (e.g. browser is shut down)

Altri metodi

Äpplet eredita da `Panel`, `Container` and `Component`

Lettura dei parametri di un applet: HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">

<HTML>
<HEAD>
  <TITLE>Customizable HelloWWW Applet</TITLE>
</HEAD>

<BODY>
<H1>Customizable HelloWWW Applet</H1>
<P>
<OBJECT CODETYPE="application/java"
        CLASSID="java:HelloWWW2.class" WIDTH=400 HEIGHT=40>
  <PARAM NAME="BACKGROUND" VALUE="LIGHT">
</OBJECT>
<P>
<OBJECT CODETYPE="application/java"
        CLASSID="java:HelloWWW2.class" WIDTH=400 HEIGHT=40>
  <PARAM NAME="BACKGROUND" VALUE="DARK">
</OBJECT>
<P>
<OBJECT CODETYPE="application/java"
        CLASSID="java:HelloWWW2.class" WIDTH=400 HEIGHT=40>
</OBJECT>

</BODY>
</HTML>
```


Lettura dei parametri di un applet: Java

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

public class HelloWWW2 extends Applet {
  public void init() {
    setFont(new Font("Helvetica", Font.BOLD, 30));
    String backgroundType = getParameter("BACKGROUND");
    if (backgroundType != null &&
        backgroundType.equals("LIGHT")) {
      setBackground(Color.white);
      setForeground(Color.black);
    } else if (backgroundType != null &&
               backgroundType.equals("DARK")) {
      setBackground(Color.black);
      setForeground(Color.white);
    } else {
      setBackground(Color.gray);
      setForeground(Color.darkGray);
    }
  }

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

Controllo caricamento immagini

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

public class BetterImageBox extends Applet {
  private int imageWidth, imageHeight;
  private Image image;

  public void init() {
    String imageName = getParameter("IMAGE");
    if (imageName != null)
      image = getImage(getDocumentBase(), imageName);
    else
      image = getImage(getDocumentBase(), "error.gif");
    setBackground(Color.white);
    MediaTracker tracker = new MediaTracker(this);
    tracker.addImage(image, 0);
    try {
      tracker.waitForAll();
    } catch(InterruptedException ie) {}
    if (tracker.isErrorAny())
      System.out.println("Error while loading image");

    // This is safe: image is fully loaded
    imageWidth = image.getWidth(this);
    imageHeight = image.getHeight(this);
  }
```



```
  public void paint(Graphics g) {
    g.drawImage(image, 0, 0, this);
    g.drawRect(0, 0, imageWidth, imageHeight);
  }
}
```