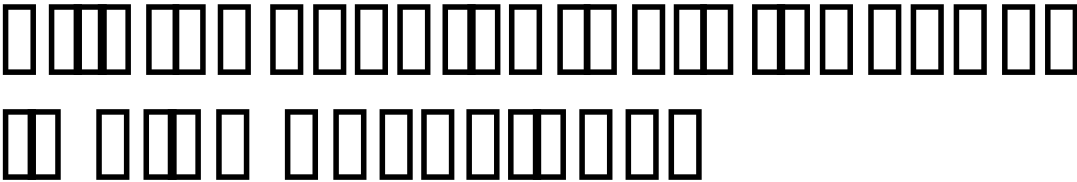
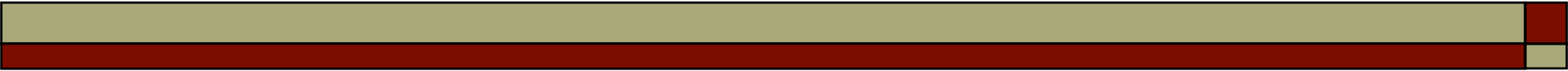
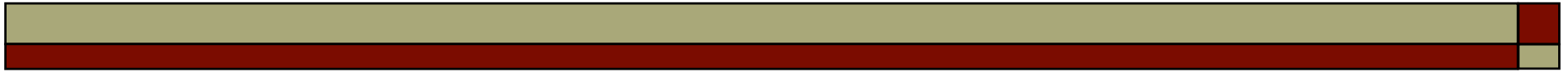


Cours 2

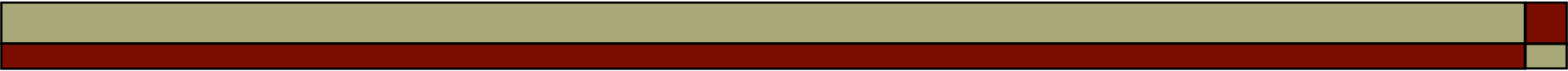
Introduction





THE HISTORY OF THE UNITED STATES

THE HISTORY OF THE UNITED STATES
FROM THE FIRST SETTLEMENTS TO THE PRESENT
BY JAMES M. SMITH
VOLUME I
THE EARLY PERIOD
FROM 1607 TO 1789
CHAPTER I
THE FIRST SETTLEMENTS
THE EARLY PERIOD
FROM 1607 TO 1789
CHAPTER II
THE SECOND PERIOD
FROM 1789 TO 1861
CHAPTER III
THE THIRD PERIOD
FROM 1861 TO 1914
CHAPTER IV
THE FOURTH PERIOD
FROM 1914 TO THE PRESENT
CHAPTER V
THE FIFTH PERIOD
FROM 1914 TO THE PRESENT
CHAPTER VI
THE SIXTH PERIOD
FROM 1914 TO THE PRESENT
CHAPTER VII
THE SEVENTH PERIOD
FROM 1914 TO THE PRESENT
CHAPTER VIII
THE EIGHTH PERIOD
FROM 1914 TO THE PRESENT
CHAPTER IX
THE NINTH PERIOD
FROM 1914 TO THE PRESENT
CHAPTER X
THE TENTH PERIOD
FROM 1914 TO THE PRESENT





- 11

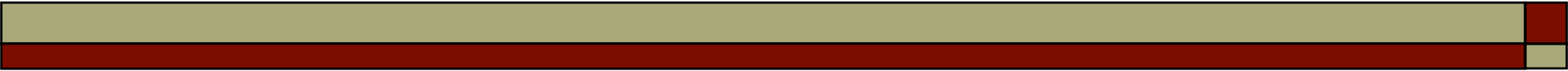
集合

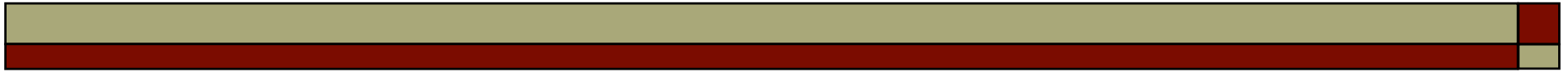
■ 集合の分類

- 順序集合
Vector
java.util.package
ArrayList
- 連結集合
LinkedList
- 数値集合
Integer
Integer













- 000





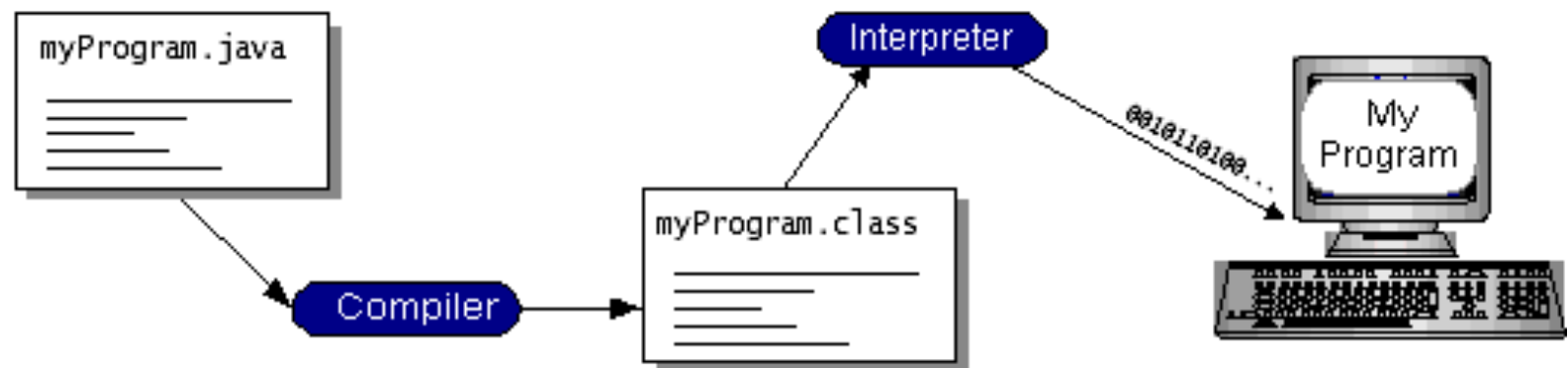
- 000

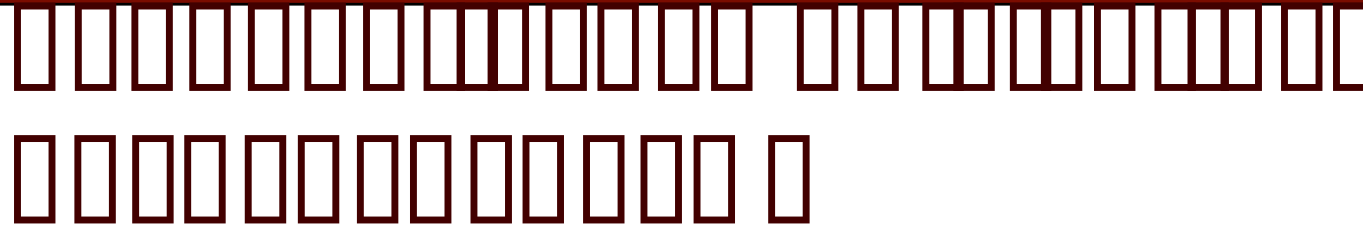


Java Architecture

- Java is a high-level, object-oriented programming language that is platform-independent and runs on a virtual machine.
- Java code is compiled into bytecode, which is then executed by the Java Virtual Machine (JVM).

□ `javac` is the Java compiler.
□ `java` is the Java interpreter.





- 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
- 1 2 3 4 5 6 7 8 9 10 11 12

■ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

- 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Java Program

```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

HelloWorldApp.java.

Compiler

Interpreter

Interpreter

Interpreter



Win32



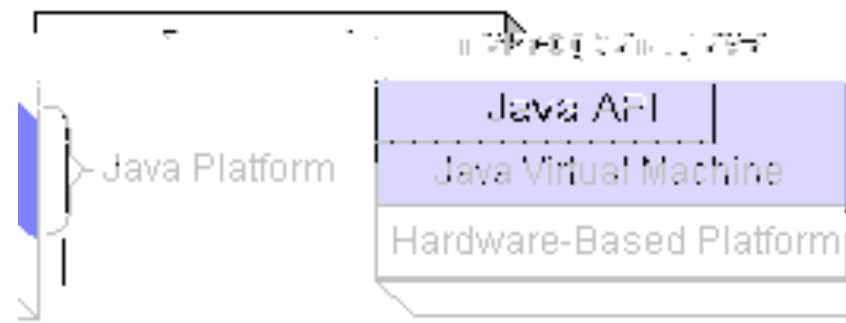
Solaris

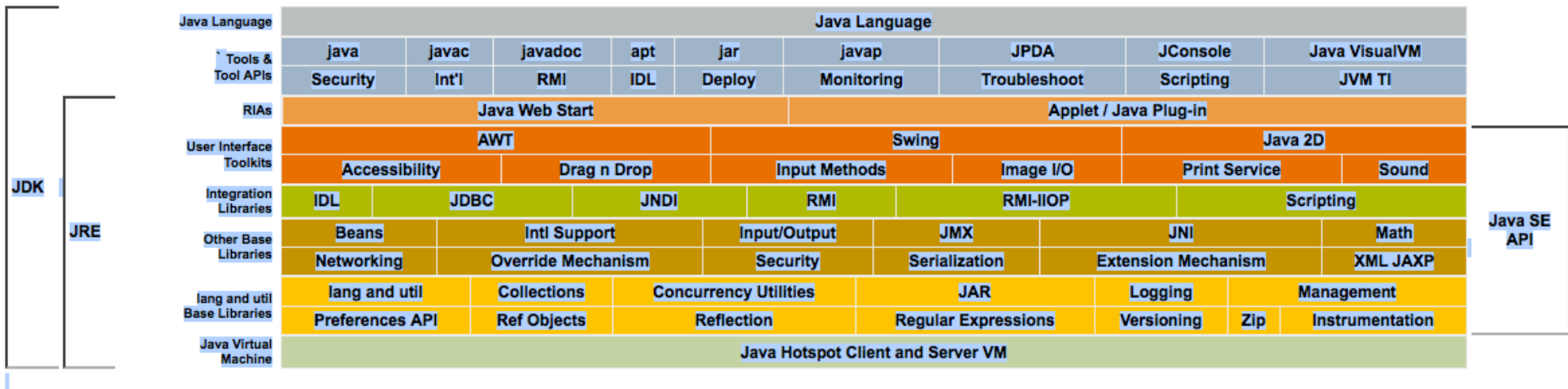


MacOS

Java Platform

- Java Platform은 Java API, Java Virtual Machine, Hardware-Based Platform로 구성된다.
- Java API는 Java Virtual Machine를 통해 실행되는 Java 프로그램에 대한 인터페이스를 제공한다.
- Java Virtual Machine는 Java API를 실행하고, Hardware-Based Platform을 통해 실행되는 Java 프로그램을 실행한다.





Java SE 6 API Documentation



- □ □



□ □□□□□□□ Appli.java:

```
/**
 * Une application  basique...
 */
class Appli {
    public static void main(String[] args) {
        System.out.println("Bienvenue en L3...");
        //affichage
    }
}
```

编译与运行

- 编译源文件 `App1.java`
 - 编译选项
 - `javac App1.java`
 - 编译生成字节码文件 `App1.class`
 - 运行字节码文件
 - `java App1`
 - 运行选项
 - 运行选项包括：运行类名、主类名、主方法名、主参数等
- Exception in thread "main" java.lang.NoClassDefFoundError:
- 运行选项包括：运行类名、主类名、主方法名、主参数等
 - 运行选项包括：运行类名、主类名、主方法名、主参数等



- 000

PrintStream

PrintStream

- out is a PrintStream object that is created by the System class
- println is a method of the PrintStream class
- out.println is a statement that prints a line of text to the standard output stream

■ out.println is a statement that prints a line of text to the standard output stream



- 000



- 11



□ □ □ □ □ □ □ □ □ □ □ □

```
/**
 * Une applet basique...
 */
import java.applet.Applet;
import java.awt.Graphics;
public class MonApplet extends Applet {
    public void paint(Graphics g){
        g.drawString("Bienvenue en en L3...", 50,25);
    }
}
```



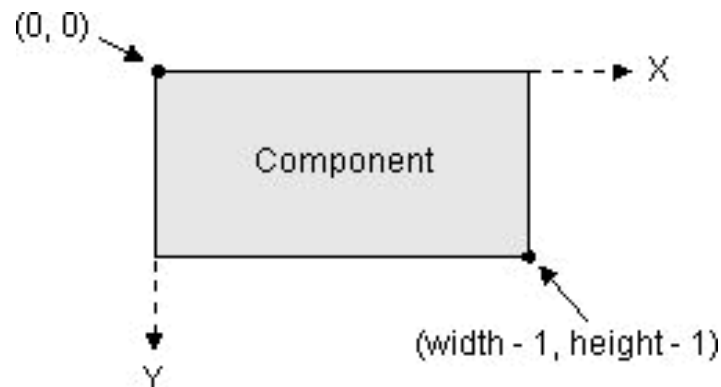
- 000



- 000

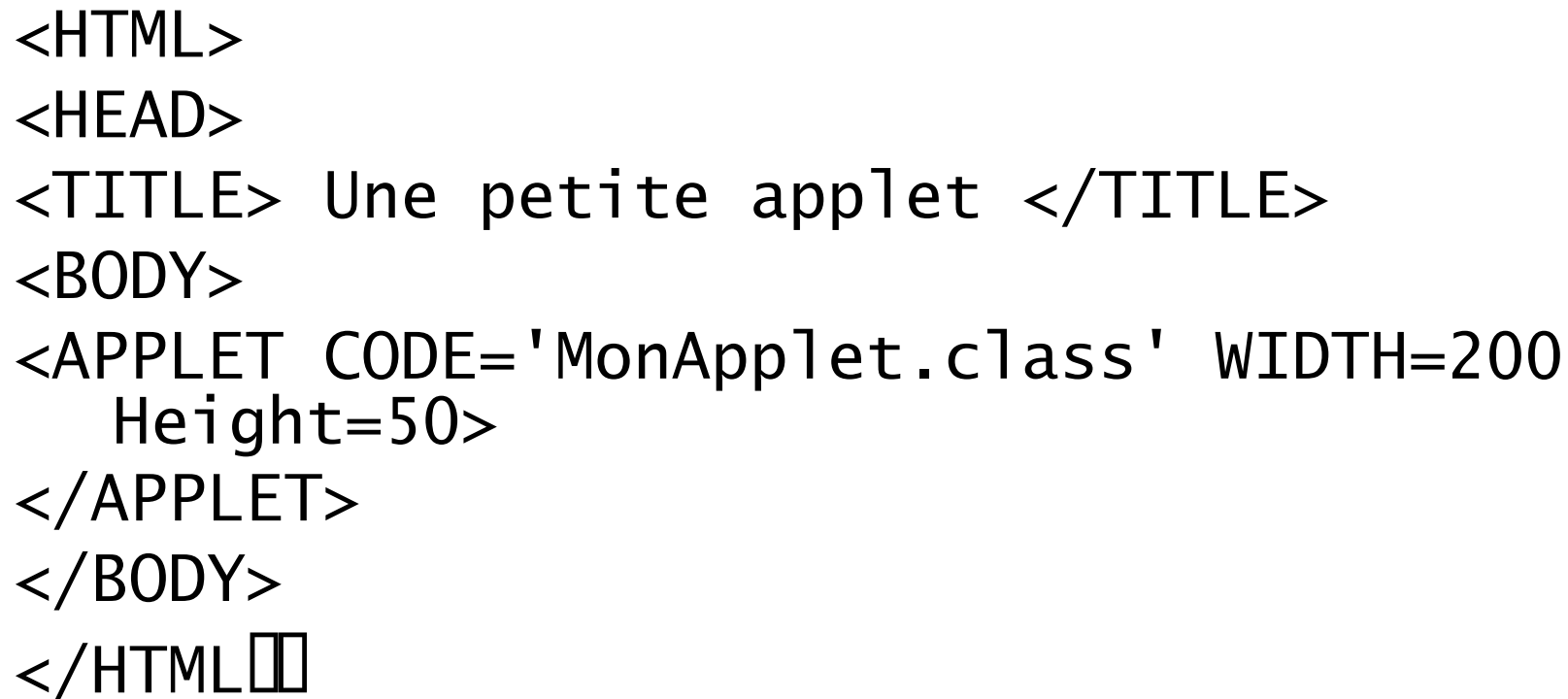
GUI Programming

- `paint start`
`init` `paint`
`paint start`
- `Graphics g` `paint`
`drawString`
`drawString`





- □ □



□ □□ □□

□ □ □□□□□□□□ □□□□ □□□□ □□□

□ □ □□ □□□□

■ <HTML> </HTML>

■ □□□□□

□ <a target="_blank" href="http://
www.liafa.jussieu.f/~hf">page de hf

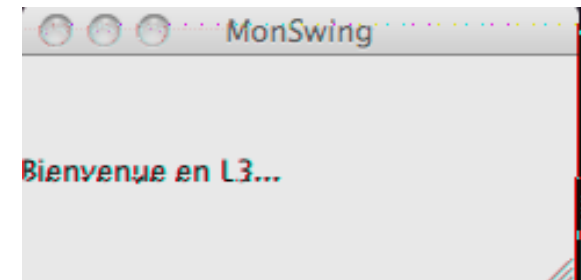
□ □□□□

<APPLET CODE='MonApplet.class' WIDTH=200
Height=50>
</APPLET>



000000 000 00000000

```
/**
 * Une application basique... avec interface graphique
 */
import javax.swing.*;
public class MonSwing {
    private static void creerFrame() {
        //Une formule magique...
        JFrame.setDefaultLookAndFeelDecorated(true);
        //Creation d'une Frame
        JFrame frame = new JFrame("MonSwing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //Afficher un message
        JLabel label = new JLabel("Bienvenue en L3...");
        frame.getContentPane().add(label);
        //Afficher la fenêtr
        frame.pack();
        frame.setVisible(true);
    }
    public static void main(String[] args) {
        creerFrame();
    }
}
```





- 000