

Curso de iniciación a la programación



Criterios del curso

- ¿Para quién?

Criterios del curso

- ¿Para quién?
- Teoría / práctica

Criterios del curso

- ¿Para quién?
- Teoría / práctica
- Lenguaje

Objetivos - Primer Día

- Programar: ¿qué es? / Esquema general
- Herramientas
- Variables, Condicionales, Bucles, Funciones

Programar

Programar

Conseguir que un ordenador haga lo que
nosotros queramos

Lenguajes de programación

Lenguajes de programación

Formas distintas de expresar lo que queremos
que haga el ordenador

Distintos paradigmas

Distintos paradigmas

Un paradigma de programación representa un enfoque particular o filosofía para diseñar soluciones.

Distintos paradigmas

- Imperativo

Distintos paradigmas

- Imperativo
- Orientado a objetos

Distintos paradigmas

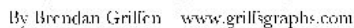
- Imperativo
- Orientado a objetos
- Funcional

Distintos paradigmas

- Imperativo
- Orientado a objetos
- Funcional
- Lógico

Distintos paradigmas

- **Imperativo**
- **Orientado a objetos**
- Funcional
- Lógico



Moss



Moss



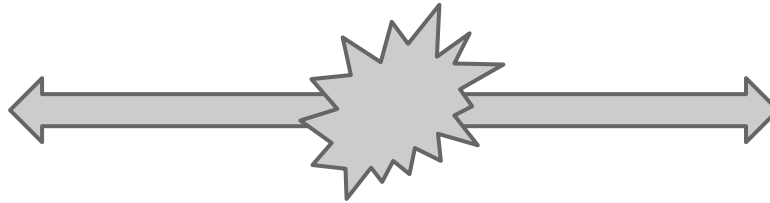
Código fuente



Moss



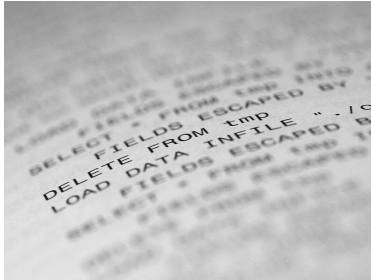
Código fuente



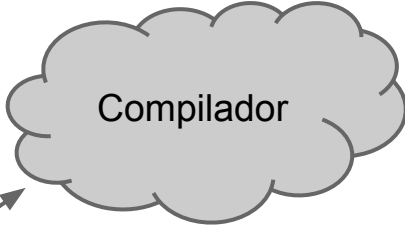
Moss



Código fuente



Compilador



Binario



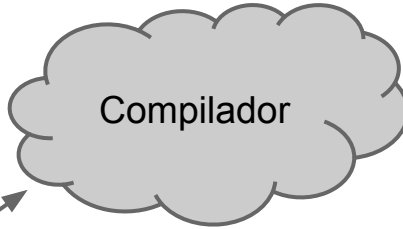
Moss



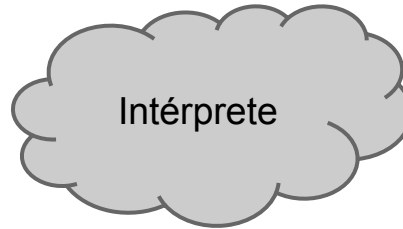
Binario



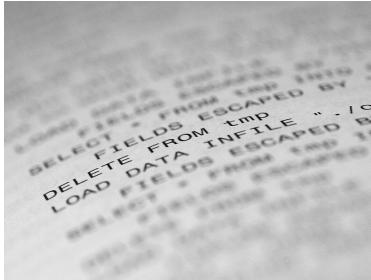
Compilador



Intérprete



Código fuente



Moss



Binario



Compilador



Código fuente



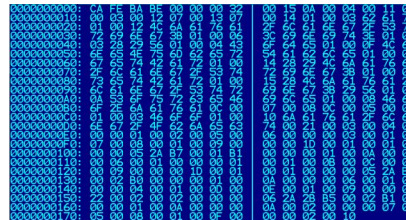
Intérprete



Compilador



Bytecode



Intérprete



Herramientas

- **Editor de texto**

- Linux: Kate, gedit, Nano
- Windows: Notepad, Notepad++
- OSX: TextEdit (“Format” -> “Make Plain Text”)
- Cualquier sistema: Vim, Emacs, Sublime Text

Según el lenguaje:

- **Intérprete** (Python, JavaScript, ...)
o bien
- **Compilador** (Java*, C/C++, ...)

Editor de texto



- ¿Sirve el Word?

Editor de texto



- ¿Sirve el Word?
- No.

Editor vs IDE (entorno de desarrollo)

Editor vs IDE (entorno de desarrollo)

- **Editor de texto** (Notepad, gedit, ...)
 - Sencillo
 - Genérico
 - Ofrece poca o ninguna ayuda

Editor vs IDE (entorno de desarrollo)

- **Editor de texto** (Notepad, gedit, ...)
 - Sencillo
 - Genérico
 - Ofrece poca o ninguna ayuda
- **IDE** (Eclipse, DrJava, Visual Studio, ...)
 - Complejo
 - Especializado en uno o varios lenguajes
 - Ofrece mucha ayuda (suele incluir editor, compilador/intérprete, depurador, etc)

Java



- Linux familia Arch: **sudo pacman -S jdk8-openjdk**
- Linux familia Debian: **sudo apt-get install openjdk-8-jdk**

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Plantilla

```
public class IntroProg {  
    public static void main(String args[]) {  
        System.out.println("Hola, mundo!");  
    }  
}
```

**Guardar en “IntroProg.java” (por ejemplo)
y comprobar que funciona**

Datos básicos

- Números:
 - Enteros (int): 14, 29, -10, 0, ...
 - Reales (float): 2.7, 19.0, -120.3, ...
- Caracteres (char): 'a', 'w', ...
- Strings (String): "Hola, mundo!", "142", ...
- Booleanos (boolean): true, false.

Operaciones básicas

- `2 + 8`
- `2.5 * 2`
- `4 / 0`
- `9 % 2`
- `"Hola, " + "mundo!"`
- `true && true`
- `true && false`
- `true || false`
- `1 < 2`
- `1 == 2`
- `(1 > 2) || (1 != 2)`

Variables (I)

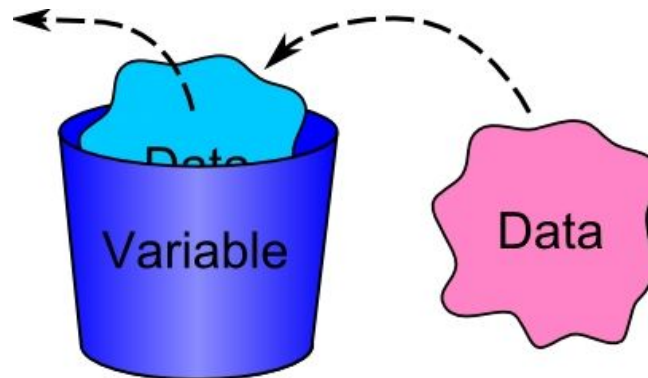
```
> int a = 3
```

```
> a                (=> 3)
```

```
> a + 7            (=> 10)
```

```
> a = 10
```

```
> a + 7            (=> 17)
```



Variables (II)

```
> int a = 3
```

```
> int b = 100
```

```
> a + b           (=> 103)
```

```
> b = a + b
```

```
> a               (=> 3)
```

```
> b               (=> 103)
```

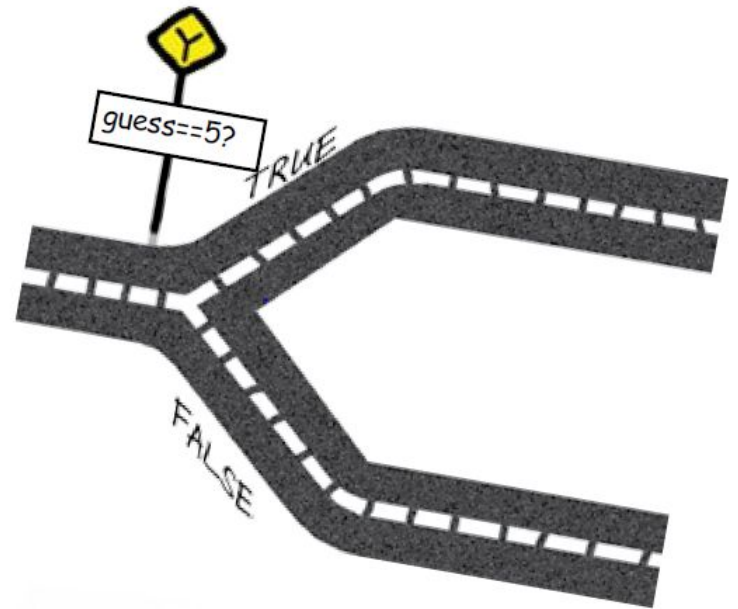
Ejemplo...

“Ve a comprar el pan, hijo. Pide una baguette y si no hay pide candeal. Si no hay ni una ni la otra pues coje una pistola.”

Condicionales

Se comprueba una <condición> para decidir qué parte de <código> ejecutar.

```
if (<condición>) {  
    <código1>  
} else {  
    <código2>  
}
```



Volviendo al ejemplo...

```
String compra;  
int baguettes = 0;  
int candeals = 3;  
int pistolas = 3;  
  
if (baguettes > 0) {  
    compra = "baguette";  
} else if (candeals > 0) {  
    compra = "candeal";  
} else {  
    compra = "pistola";  
}
```

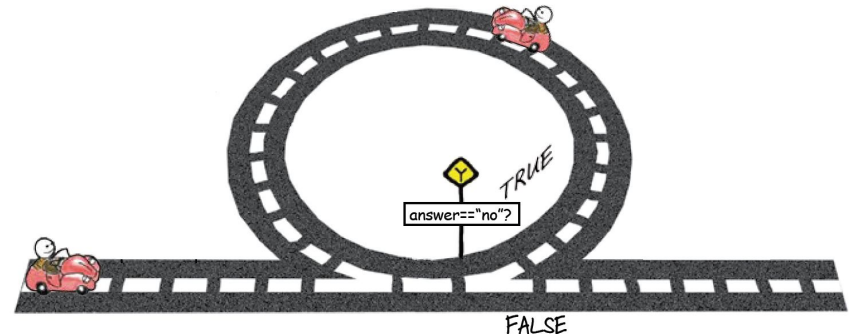
Y si...

“Castigado, escribe en la pizarra 100 veces ‘No lo volveré a hacer’.”

Bucles

Mientras se cumpla una <condición> el <código> se ejecuta.

```
while (<condición>) {  
    <código>  
}
```



while

```
int veces = 0;
boolean fin = false;

while (fin == false) {
    veces = veces + 1;
    System.out.println("No lo volveré a hacer.");
    if (veces == 100) {
        fin = true;
    }
}
```

Funciones

Vamos a calcular el factorial de un número: 80!

```
int a = 80 * 79 * 78 * 77 * ... * 3 * 2 * 1
```

¡Mucho trabajo!



Funciones

Creamos `<código>` con `<nombre>` para poder guardarlo y utilizarlo libremente las veces necesarias.

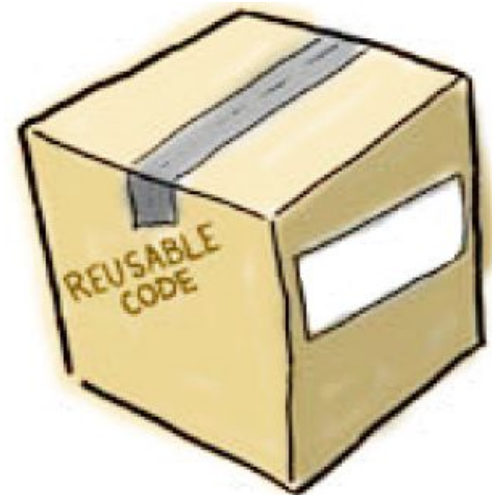
Las funciones pueden recibir `<datos>` y devolver el `<resultado>` de las operaciones que realicen

Las funciones tienen un `<resultado>` pueden recibir `<datos>` y devolver el `<resultado>` de las operaciones que realicen

```
int factorial (int n)
```

Funciones

```
int factorial (int n) {  
    int resultado = 1;  
    while (n>1) {  
        resultado = resultado*n;  
        n = n-1;  
    }  
    return resultado;  
}
```



¿Dudas / preguntas?

¡Gracias!