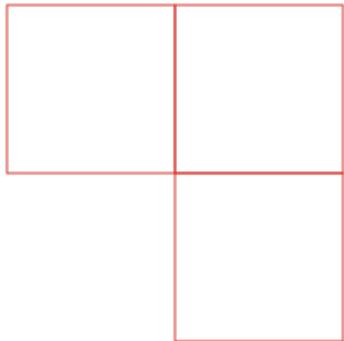


# 3

## Variables y constantes

Ve más allá





# CONTENIDOS

## ■ Unidad 1: Fundamentos del lenguaje Java

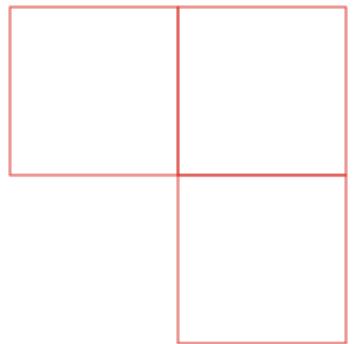
1. Nuestro primer programa
2. Sentencias y errores
- 3. Variables y constantes**
4. Tipos de datos
5. Operadores y expresiones
6. Entrada y Salida

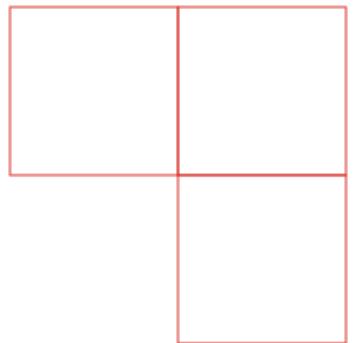




# ÍNDICE

- Variables y constantes
  - 1.Qué son las variables
  - 2.Variables en Java
  - 3.Atributos Vs Variables locales
  - 4.Constantes



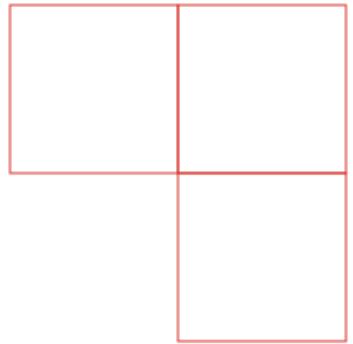


# ¿Qué son las variables?

Una variable no es más ni menos que una zona de memoria donde se puede almacenar información del tipo que desea el programador. Vamos a conocer para qué se utilizan, los nombres que debemos darles, la diferencia entre variables locales y atributos, y entre variables y constantes.

Las variables se utilizan para almacenar un valor que necesitarás más tarde. Una forma sencilla de imaginarte una variable, podría ser un trozo de papel donde has apuntado el número de la plaza del parking donde has dejado el coche.





# Variables en Java

- En Java las variables se declaran poniendo primero el tipo y luego el nombre de la variable, con esta sintaxis:

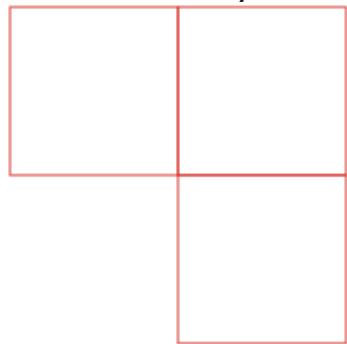
```
<tipo> <nOMBRE>;
```

- Por ejemplo, para almacenar la plaza de parking nos puede venir bien un tipo de datos entero (sin decimales), con el nombre de `plazaParking`, de la siguiente manera:

```
int plazaParking;
```

- Lo que hemos hecho ha sido declarar la variable. Si más tarde, queremos darle un valor, podemos hacer:

```
plazaParking = 30;
```



# Variables en Java

Elegir un buen nombre de variable hace que tu código sea más fácil de leer por un humano, y por lo tanto, más fácil de comprender.

Desde el primer momento debes acostumbrarte a no poner nombres como x,y,z o ya que no dice nada sobre el objetivo de esa variable. Hay que buscar nombres lo más significativos posibles.

En Java es sensible a mayúsculas y minúsculas y se utiliza CamelCase para juntar palabras, es decir, que se pone la primera letra en mayúsculas y el resto en minúsculas.

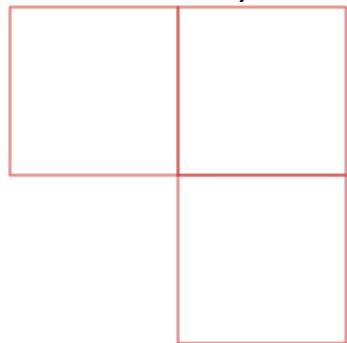
Se recomienda utilizar el alfabeto anglosajón para evitar problemas de codificación de caracteres. Se considera una buena práctica utilizar nombres en inglés.

## Variable Names



Use a variable to  
store a value that you  
need later:

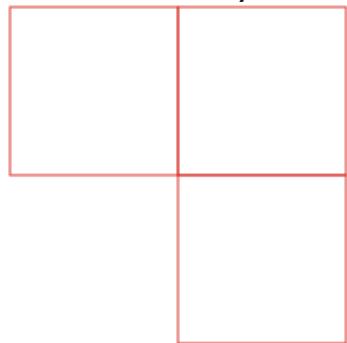




# Variables en Java

Por convención, los nombres que debes utilizar son:

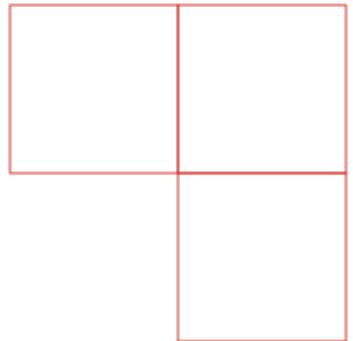
- **Clases e interfaces:** Deben comenzar en mayúscula.
  - Por ejemplo: Bicycle
- **Métodos:** Deben comenzar en minúscula y ser verbos.
  - Por ejemplo: changeGear
- **Variables:** Deben comenzar en minúscula y ser nombres.
  - Por ejemplo: speed
- **Constantes:** Mayúsculas y separados por "\_".
  - Por ejemplo: POSITIVE\_INFINITY
- **Paquetes:** Deben estar escritos en minúscula.
  - Por ejemplo: java.lang



# Atributos y variables locales

Veamos el siguiente código:

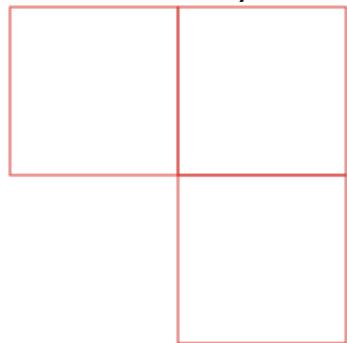
```
class suma {  
    static int n1=50;          // Atributo  
    public static void main(String [] args) {  
        int n2=30, suma=0;    // variables locales  
        suma=n1+n2;  
        System.out.println("LA SUMA ES: " + suma);  
    }  
}
```



# Atributos y variables locales

Las variables declaradas en el bloque de la clase como n1 se consideran atributos, mientras que las variables n2 y suma pertenecen al método main y solo pueden ser utilizados en el mismo. Las variables declaradas en el bloque de código de un método son variables locales, que se crean cuando el bloque se declara, y se destruyen cuando finaliza la ejecución de dicho bloque.

¿Cuándo utilizar un atributo o una variable local? Todo depende del tiempo que queremos recordar el valor de la variable. Los atributos también se conocen como variables globales y los utilizamos cuando queremos guardar algo durante más tiempo y queremos que sea accesible desde cualquier lugar de nuestro programa. Las variables locales, por otro lado, son el equivalente de la memoria a corto plazo.



# Atributos y variables locales

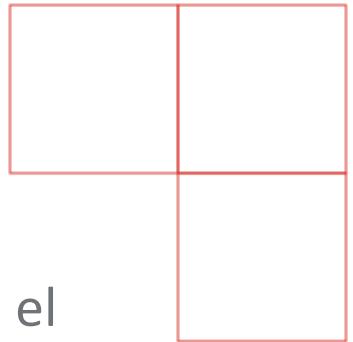
Quizás recuerdes que hoy comiste pizza, pero no lo recordarás por mucho tiempo. Sin embargo, si tienes un animal en casa, siempre deberías recordar que debes cuidar de él y darle de comer.

Lo primero sería una variable local, mientras que lo segundo sería un atributo.

Si estamos creando la clase Coche, por ejemplo, un atributo podría ser la cantidad de combustible que tienes en el depósito, mientras que una variable local sería los litros que has utilizado en el último trayecto.

Local and Instance Variables





# Constantes

Las constantes son variables cuyo valor no deseamos que cambie. Se declaran siguiendo el siguiente formato:

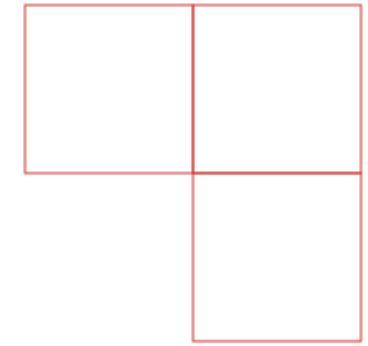
```
final [static] <tipo de datos> <nombre de la constante> = <valor>;
```

Donde el calificador final identificará que es una constante, la palabra static (opcional) sirve para indicar que el atributo pertenece a la clase y no al objeto, por lo todos los objetos tendrán el mismo valor. Esto último lo entenderemos mejor más adelante, cuando trabajemos la orientación a objetos.

Vemos el siguiente ejemplo:

```
final double PI=3.141592;
```

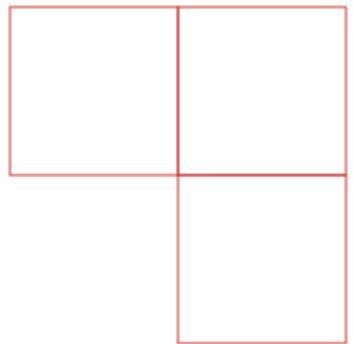
Estamos declarando la variable PI como constante (final) de tipo real (double) y además, estamos utilizando la declaración para darle un valor inicial.



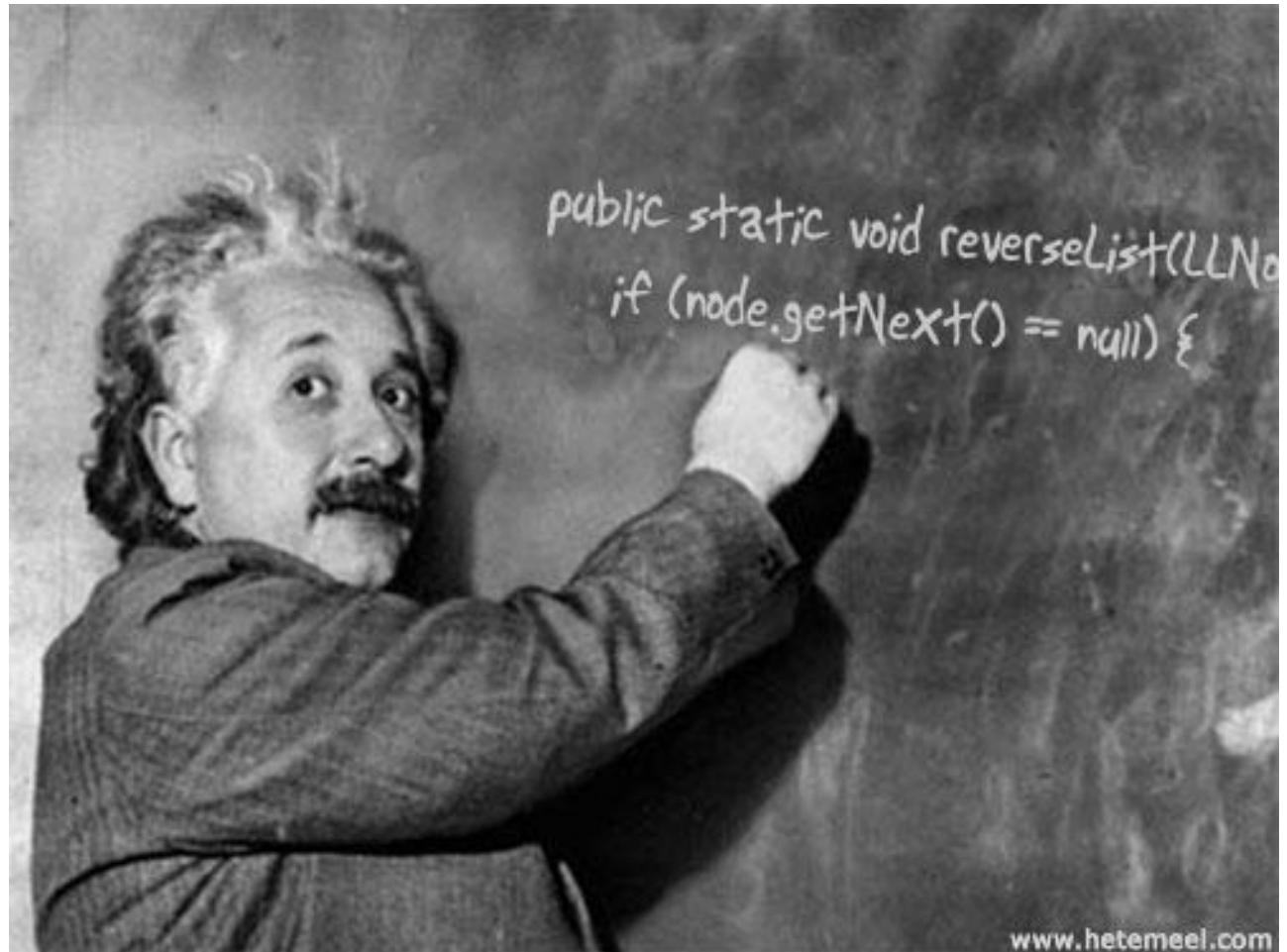
# Constantes

Las constantes se utilizan en datos que nunca varían (IVA, PI, etc.). Utilizando constantes y no variables nos aseguramos de que su valor no va a poder ser modificado nunca.

También utilizar constantes permite centralizar el valor de un dato en una sola línea de código (si se quiere cambiar el valor del IVA se hará solamente en una línea en vez de si se utilizase el literal 18 en muchas partes del programa).



# APUNTE...





**Universidad  
Europea**

# GRACIAS

Pedro J. Camacho

[Universidadeuropea.com](http://Universidadeuropea.com)

Ve más allá