

Variable booleanas



Valores de verdad

TRUE

FALSE

Tipo de dato booleano

Es un tipo de dato que sólo puede tener dos valores:

→ Verdadero (true)

→ Falso (false)

TRUE

FALSE

Variable booleana falsa



esAutoVerde →

FALSE

Variable booleana verdadera



esAutoRojo →

TRUE

Qué pasa si niego una variable booleana?



!(esAutoRojo) →

Qué pasa si niego una variable booleana?



!(esAutoRojo) →

FALSE

Cómo se declara una variable booleana?

```
boolean esMayor = true;
```

```
if ( esMayor == true ) {
```

```
    MOSTRAR ("Es mayor de edad");
```

```
} else {
```

```
    MOSTRAR ("Es menor de edad");
```

Cómo se actualiza una variable booleana

```
boolean esMayor = false;  
  
MOSTRAR (" Ingrese su edad");  
  
GUARDAR edad;  
  
if (edad >= 18) {  
    esMayor = true;  
}
```



Podemos imprimir una variable booleana?

```
boolean esMayor = false;  
  
MOSTRAR ( esMayor);
```

Va a mostrar: **false**

Esto no significa mucho para el usuario!!



Ejemplo

```
boolean esMayor = false;  
  
If ( esMayor == true) {  
    MOSTRAR ("Es mayor");  
  
else {  
    MOSTRAR ("Es menor de edad");  
  
}
```

Cómo se nombran las variables booleanas?

Como todas las variables, es recomendable siempre hacerlo de forma tal que su nombre sugiere el tipo de dato que guarda y su uso

Ej.

esMayor

esPar

estaPrendido

esCorrecto

Cómo se nombran las variables booleanas?

Todos estos nombres, sugieren que se está guardando algo con un cierto valor de verdad: verdadero o falso.

```
boolean esMayor = true
```

```
boolean esPar = false;
```

```
boolean estaPrendido = true;
```

```
boolean esCorrecto = false;
```



Métodos

Semana 3



Las funciones....

En Matemáticas: **fun**(**x**) = **2x + 1**

En Java: public static int **fun** (int **x**) {
 return (**2 * x + 1**);
 }

fun → nombre de la función

x → parámetro

2x+1 → cuerpo de la función

int → representa a los enteros



Qué es la firma de la función?

La firma de un método es la definición de las características del mismo.

Ejemplo:

```
public static void main (String args []) {}
```

```
public static int obtenerSuma (int num1, int num2){}
```



Partes de un método

- Visibilidad
- Tipo de retorno
- Nombre del método
- Parámetros

public static void mostrarDoble (int número)



Visibilidad

Si quiero que el método sea oculto a otras clases, debemos definirlo como privado... (private)

Si el método puede visto y utilizado por otras clases, entonces debemos definirlo como público... (public)

public static void mostrarDoble (int número)

private static void sumar (int num1, num2)



Tipo de retorno

- Un método puede retornar un valor o no... (void)
- El tipo de retorno puede ser cualquier tipo de dato (int, String, boolean, etc)

Ejemplo

```
public static void mostrarDoble ( int num ) { MOSTRAR (num*2); }
```

Este método solo muestra en pantalla el doble de un número... no es necesario que retorne nada, por eso, el retorno es void (sin retorno)

Ejemplo

```
public static int calcularDoble ( int num ) { return num * 2; }
```

Este método calcula y retorna el doble de un número



Nombre del método

- El nombre de los métodos debe ser descriptivo.
- Siempre comienza en minúscula
- Pueden contener varias palabras

```
public static void mostrarDoble ( int número )
```



Parámetros

Un método puede recibir 0, 1 o n parámetros.

Cada método sólo conoce los parámetros que recibe, no los de otros métodos.

```
public static void mostrarDoble ( int número ) { ..... }
```

```
public static int multiplicar ( int num1, int num2 ) { return num1 * num2;
```

Ejercicios

Cómo se utiliza un método?



Utilizando un método sin retorno

```
public static void mostrarDoble (int num) {  
    System.out.println( num * 2);  
}
```

Utilizando un método sin retorno

```
public class ejercicio1 {  
  
    public static void main (String args [] ) {  
  
        mostrarDoble(2);  
  
    }  
  
    public static void mostrarDoble (int num) {  
  
        System.out.println( num * 2);  
  
    }  
}
```

Simplemente se invoca el método dentro del main, escribiendo el nombre del método y agregando valores que recibirá el método

La variable num, vale 2 dentro del método. Se imprimirá 4 en pantalla...



Utilizando un método con retorno

```
public static int calcularDoble (int num) {  
  
    return ( num * 2);  
  
}
```



Utilizando un método con retorno

```
public class ejercicio1 {  
  
    public static void main (String args [] ) {  
  
        int doble = calcularDoble(4);  
  
    }  
  
    public static int calcularDoble (int num) {  
  
        return ( num * 2);  
  
    }  
}
```

calcularDoble retorna un valor entero; debo guardarlo en una variable...

La variable num, vale 4 dentro del método. El método retorna 8 y será asignado a la variable doble dentro del main

Puede haber varios métodos con el mismo nombre?



Sobrecarga de métodos

La sobrecarga de métodos es la creación de varios métodos con el **mismo nombre** pero con diferente cantidad y tipos de parámetros.

```
public static int calculaSuma(int x, int y) { ... }
```

```
public static int calculaSuma(int x, int y, int z){ ... }
```

```
public static int calculaSuma(int x, int y, int z, int w){ ... }
```



Ejemplo de sobrecarga

```
Public static void main ( String args [] ) {
```

```
    int suma = calcularSuma ( 2, 5);
```

```
}
```

```
public static int calculaSuma(int x, int y) { return x + y; }
```

```
public static int calculaSuma(int x, int y, int z){ return x + y + z; }
```

```
public static int calculaSuma(int x, int y, int z, int w){ return x + y + z +w; }
```



Ejemplo de sobrecarga

```
Public static void main ( String args [] ) {  
    int suma = calcularSuma ( 2, 5, 4);  
}  
  
public static int calculaSuma(int x, int y) { return x + y; }  
public static int calculaSuma(int x, int y,int z){ return x + y + z; }  
public static int calculaSuma(int x, int y, int z, int w){ return x + y + z + w; }
```



Ejemplo de sobrecarga

```
public static void main ( String args [] ) {  
    int suma = calcularSuma ( 2, 5, 3, 1);  
}  
  
public static int calculaSuma(int x, int y) { return x + y; }  
public static int calculaSuma(int x, int y,int z){ return x + y + z }  
public static int calculaSuma(int x, int y, int z, int w){ return x + y + z +w; }
```

