

Java Standard Web Programming

Módulo 1

Variables y constantes

Introducción

Para manejar y/o manipular información, dentro de un sistema, necesitamos **almacenarla en algún lugar mientras se procesa**. Ese lugar será una base de datos o un archivo que persista en el tiempo para que, luego, se pueda recuperar.

Para este propósito los lenguajes de programación tienen: **variables** (pueden cambiar su valor a los largo del tiempo) y **constantes** (no cambian su valor una vez asignado) que **se almacenarán de forma temporal en la memoria de la computadora**.

Al espacio de memoria que guardará un dato (variable) se le debe asignar un identificador "**Nombre**".



Identificador

Se puede decir que gran parte del tiempo que dedicamos a programar es para darle un **nombre a las variables y constantes**. Aunque se puede asignar cualquier identificador, es recomendable **utilizar nombres significativos que describan para qué se usarán**.

Ejemplos

- **Variables**

numero1, edad, nombre, \$contador, etc.

- **Constantes**

IMPUESTO, VERSION, y otros.

Reglas de nomenclatura:

- No deben contener espacios.
- Los únicos caracteres permitidos son el guión bajo “_” y el “\$”.
- Pueden contener números y letras.
- Si posee números, estos no deben ir al inicio del identificador.

También es recomendable usar las **convenciones** cuando se necesitan **nombres compuestos**.

Para variables compuestas

- **Lower Camel Case o Snake Case:**
`razonSocial`, `nombre_apellido`.

Para constantes

- **Upper Case con Snake Case:**
`DIAS_SEMANA`, `URL_BUSCADOR`, etc.



Sintaxis

Recordemos que **sintaxis** es el conjunto de reglas que brinda un lenguaje para trabajar con él.



La forma en que se **declaran variables y constantes** en Java es:

Sin darle valor inicial

1. Palabra reservada final (solo para constantes).
2. Tipo de dato.
3. Identificador.
4. Punto y coma.

```
//Constantes
final tipoDato IDENTIFICADOR;

//Variables
tipoDato identificador;
```

Con un valor inicial

1. Palabra reservada final (solo para constantes).
2. Tipo de dato.
3. Identificador.
4. Valor.
5. Punto y coma.

```
//Constantes
final tipoDato IDENTIFICADOR = valor;

//Variables
tipoDato identificador = valor;
```

Tipos de datos

Primitivos

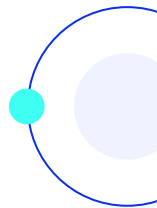
Son los elementales y fundamentales dentro de Java. Se pueden dividir en:

- numéricos enteros,
- numéricos con punto flotante,
- booleanos y
- de carácter.

Objetos

Son tipos de estructuras o datos especiales, que se comprenderán mejor más adelante.

Por lo pronto, se verá **el más usado de todos los objetos en Java: el *string*.**



Tipos de datos



	Tipo	Definición	Espacio en Memoria	Valores
Primitivos	byte	Entero	1 byte	-128 a 127
	short	Entero	2 bytes	-32768 a 32767
	int	Entero	4 bytes	$2 \cdot 10^9$
	long	Entero	8 bytes	Muy grande
	float	Decimal simple	4 bytes	Muy grande
	double	Decimal doble	8 bytes	Muy grande
	char	Carácter simple	2 bytes	[caracteres simples]
	boolean	Verdadero y Falso	1 byte	[true/false]
Objetos	String	Cadena de caracteres	Depende de la cantidad de caracteres	Cadena de Caracteres



¿Cómo asignamos valores?

- **Numéricos enteros:** literal del número.
- **Números decimales:** literal del número con el punto como separador de decimales. Algo particular que pasa con los `float` es que necesitan la letra “f” al final del número, ya que para Java todos los datos con decimales son de tipo **double** y así los diferenciamos.
- **Booleanos:** literales `true` o `false`.
- **Carácter:** carácter encerrado entre ‘ ’.
- **Cadena de caracteres:** cadena encerrada entre “ ”.

```
//Numeros

byte miByte = 5;

short miShort = 12596;

int miInt = 10;

long miLong = 1000000000;

float miFloat = 10.58f;

double miDouble = 10.58;

//Booleanos

boolean miBoolean = false;

//Caracteres

char miChar = 'P';

//Cadena de caracteres

String miString = "Hola Mundo";
```



Secuencias de escape

Suministran un mecanismo que permite expresar códigos de caracteres que, de otro modo, no se permitirían en los `string`.



```
System.out.println("Primera linea \n Segunda linea \t despues del tabulado...");
```



Secuencias de Escape

Secuencia	Descripción
<code>\n</code>	Salto de línea. Sitúa el cursor al principio de la línea siguiente.
<code>\b</code>	Retroceso. Mueve el cursor un carácter atrás en la línea actual.
<code>\t</code>	Tabulador horizontal. Mueve el cursor hacia adelante una distancia determinada por el tabulador.
<code>\r</code>	Ir al principio de la línea. Mueve el cursor al principio de la línea actual.
<code>\f</code>	Nueva página. Mueve el cursor al principio de la siguiente página.
<code>\"</code>	Comillas. Permite mostrar por pantalla el carácter comillas dobles.
<code>\'</code>	Comilla simple. Permite mostrar por pantalla el carácter comilla simple.
<code>\\</code>	Barra inversa.

**¡Sigamos
trabajando!**