

# TIPOS de DATOS

1. Tipos de Datos
2. Representación de los Tipos de Datos en JAVA

[Ejemplos](#)

## Tipos de Datos Simples

- **Datos Numéricos:** Permiten representar valores escalares de forma numérica, esto incluye a los números enteros y a los reales. Este tipo de datos permiten realizar operaciones aritméticas simples.
- **Datos Alfanuméricos (string):** Es una secuencia de caracteres alfanuméricos que permiten representar valores identificables de forma descriptiva, esto incluye nombres de personas, descripciones, etc. Es posible representar datos numéricos como alfanuméricos, pero estos perderán su capacidad matemática (no se pueden realizar operaciones).
- **Datos Lógicos:** Son aquellos que solo pueden tener 2 valores: “cierto” o “false”, ya que representan el resultado de una comparación entre dos datos: numéricos o alfanuméricos.

# TIPOS DE DATOS - JAVA

- Tipos de datos:
  - Enteros
  - Decimales (Flotantes)
  - Caracteres (Char)
  - Cadenas de Caracteres (String)
  - Booleanos

[Ejemplos](#)

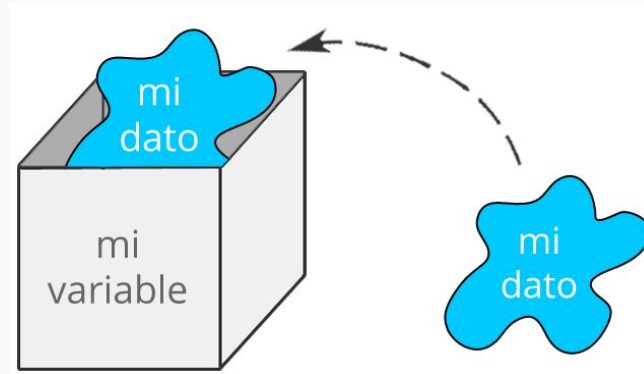
# ESTRUCTURAS SIMPLES

1. Variables
2. Constantes
3. Operadores
  - Asignación
  - Aritméticos
4. Palabras Reservadas
  - static, class. new, main, etc

[Ejercicios](#) | [Ejemplos](#)

## Variables

- Son espacios reservados en la memoria que, como su nombre indica, pueden cambiar de contenido a lo largo de la ejecución de un programa.
- Una variable corresponde a un área reservada en la memoria principal de la computadora.



## Constantes

- Es un valor que **no** puede ser alterado/modificado durante la ejecución de un programa, únicamente puede ser leído.
- Corresponde a una longitud fija de un área reservada en la memoria principal de la computadora,, donde el programa almacena valores fijos.

## Expresiones

- Las expresiones son combinaciones de variables, constantes o símbolos de operación, paréntesis o nombres de funciones especiales.

$$a + ( b + 3 ) / c$$

- Cada expresión toma un valor que se determina tomando los valores de las variables y constantes implicadas y la ejecución de las funciones indicadas.
- Una expresión consta de operadores y operandos.
- Según sea el tipo de datos que manipulas, se clasifican en:
  - **Aritméticas**
  - **Relacionales**
  - **Lógicas**

## Operadores

- Son elementos que relacionan de forma diferente, los valores de una o más variables y/o constantes.
- Los operadores permite manipular valores.
- Según sea el tipo de datos que manipulas, se clasifican en:
  - **Aritméticas**
  - **Relacionales**
  - **Lógicas**



# ESTRUCTURAS SIMPLES - OPERADORES ARITMÉTICOS

## Operadores Aritméticos

- Los operadores aritméticos permiten la realización de operaciones matemáticas con los valores de variables o constantes.

Operación	Operador	Ejemplo	Acción	Resultado
<b>Suma</b>	+	$X + Y$	Suma X más Y	12
<b>Resta</b>	-	$X - Y$	Resta X menos Y	8
<b>Multiplicación</b>	*	$X * Y$	Multiplica X por Y	20
<b>División</b>	/	$X / Y$	Divide X entre Y, regresa el cociente	5
<b>Potencia</b>	pow	pow(X,Y)	Eleva X a la Y	100
<b>Residuo</b>	%	$X \% Y$	Regresa el residuo de X entre Y	0
<b>Raíz cuadrada</b>	sqrt	sqrt(Y)	Raíz cuadrada de Y	1.4142

# ESTRUCTURAS SIMPLES - OPERADORES ARITMÉTICOS

## *Prioridad de los Operadores Aritméticos*

Todas las expresiones entre paréntesis se evalúan primero. Las expresiones con paréntesis anidados se evalúan de dentro a fuera, el paréntesis mas interno se evalúa primero.

Dentro de una misma expresión los operadores se evalúan en el siguiente orden.

1.- ^ Exponenciación

2.- \*, /, mod Multiplicación, división, modulo.

3.- +, - Suma y resta.

Los operadores en una misma expresión con igual nivel de prioridad se evalúan de izquierda a derecha.

Ejemplos:

$$4 + 2 * 5 = 14$$

$$23 * 2 / 5 = 9.2$$

$$3 + 5 * (10 - (2 + 4)) = 23$$

$$3.5 + 5.09 - 14.0 / 40 = 5.09$$

$$2.1 * (1.5 + 3.0 * 4.1) = 28.98$$

$$46 / 5 = 9.2$$

$$3 + 5 * (10 - 6) = 3 + 5 * 4 = 3 + 20 = 23$$

$$3.5 + 5.09 - 3.5 = 8.59 - 3.5 = 5.09$$

$$2.1 * (1.5 + 12.3) = 2.1 * 13.8 = 28.98$$

# ESTRUCTURAS SIMPLES - OPERADORES RELACIONALES

## Operadores Relacionales

Operador	Operación	Ejemplo	Resultado
=	Igual que	"hola"= "ola"	Falso
<>	Diferente	"a" <> "b"	Verdadero
<	Menor que	7 < 15	Verdadero
>	Mayor que	22 > 11	Verdadero
<=	Menor o igual que	15 <= 22	Verdadero
>=	Mayo o igual que	33 >= 20	Verdadero

# ESTRUCTURAS SIMPLES - OPERADORES LÓGICOS

## Operadores Lógicos

- Estos operadores se utilizan para establecer relaciones entre valores lógicos.
- Estos valores pueden ser resultado de una expresión relacional.

### *Operadores Lógicos*

And      Y  
Or        O  
Not      Negación

#### *Operador And*

<i>Operando1</i>	<i>Operador</i>	<i>Operando2</i>	<i>Resultado</i>
T	AND	T	T
T		F	F
F		T	F
F		F	F

## Operadores Lógicos

Operador	Significado
Operador &&	Operador lógico and
Operador	Operador lógico or
Operador !	Operador lógico not

# ESTRUCTURAS SIMPLES - OPERADORES LÓGICOS

Ejemplo:

Antioquia gana si Nacional Gana Medellín Gana y Envigado Gana.

Nacional	Medellín	Envigado	Antioquia
G	G	G	G
P	G	G	P
G	P	P	P

**Operador Or**

Operando1	Operador	Operando2	Resultado
T	OR	T	T
T		F	T
F		T	T
F		F	F

Recuerde que con or se cumple si alguna cumple. Utilice el mismo ejemplo.

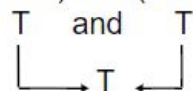
**Operador Not**

**Operando Resultado**

T	F
F	T

Ejemplos:

$(a < b)$  and  $(b < c)$   
 $(10 < 20)$  and  $(20 < 30)$



# ESTRUCTURAS SIMPLES - OPERADORES LÓGICOS

## Prioridad de los Operadores Lógicos

Not  
And  
Or

## Prioridad de los Operadores en General

- 1.- ( )
- 2.- ^
- 3.- \*, /, Mod, Not
- 4.- +, -, And
- 5.- >, <, >=, <=, <>, =, Or

## Ejemplos:

a = 10 b = 12 c = 13 d = 10

1)  $((a > b) \text{ or } (a < c)) \text{ and } ((a = c) \text{ or } (a >= b))$

F      T      F      F

└─ T ─┘      └─ F ─┘

└──────── F ─────────┘

2)  $((a >= b) \text{ or } (a < d)) \text{ and } ((a >= d) \text{ and } (c > d))$

F      F      T      T

└─ F ─┘      └─ T ─┘

└──────── F ─────────┘

3)  $\text{not } (a = c) \text{ and } (c > b)$

T      F      T

└─ T ─┘      └─ T ─┘

└──────── T ─────────┘