

Tipos de datos  
Jerarquía de operadores  
Expresiones matemáticas

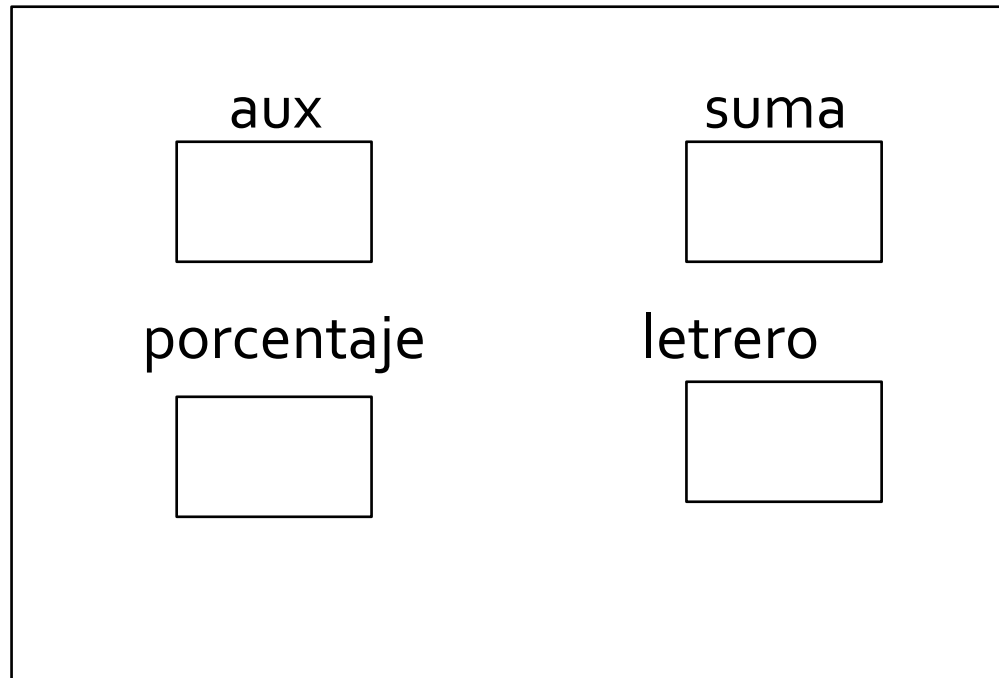
# Tipos de datos

- Los datos a procesar por una computadora, deben almacenarse en celdas de memoria para su posterior utilización. Llamaremos **identificador al nombre que se les da a las celdas de memoria**. Con las siguientes reglas:
  - El primer carácter que forma un identificador debe ser una letra o guion bajo
  - Los demás caracteres pueden ser letras o dígitos.
  - Ejemplos:  
aux    suma    acum    a1    variable    variable1    \_edad

# Tipos de datos

Es como una caja, que le ponemos un nombre, y en esa caja podemos almacenar cualquier valor que se requiera.

MEMORIA



# Palabras reservadas - Key words

```
>>> help("keywords")
```

Here is a list of the Python keywords. Enter any keyword to get more help.

and	elif	if	print
as	else	import	raise
assert	except	in	return
break	exec	is	try
class	finally	lambda	while
continue	for	not	with
def	from	or	yield
del	global	pass	

# Tipos de Datos: **int**, **float**

Para almacenar valores numéricos:

- Enteros, nos permiten almacenar números sin decimales
- Flotantes, nos permiten almacenar números con decimales
- **Ejemplos:**
- **a = 5**      5 es un número entero - **int**
- **b = 7.5**    7.5 es un número flotante - **float**

.

# Tipos de Datos: **str**

Para almacenar texto, se usa el tipo de dato string y el valor que se asigna debe estar entre comillas dobles o entre apóstrofes .

Ejemplos :

- direccion = "Calle X #33 Esquina Calle Y"
- mensaje = 'Python es "el mejor" lenguaje'

**NOTA:** Podemos introducir cualquier letra, número o símbolo en un texto

# Tipos de Datos: **lógico**

El tipo de dato lógico o booleano, solo puede tomar dos valores: verdadero (True) o falso (False)

Ejemplos :

- Dato\_logico1 = True
- Dato\_logico2 = False

# Operadores aritméticos

Operador Aritmético	Operación	Ejemplo	Resultado
**	Potencia	$4^{**}3$	64
*	Multiplicación	$8.25 * 7$	57.75
/	División	$15 / 4$	3.75
+	Suma	$125.78 + 62.50$	188.28
-	Resta	$65.30 - 32.33$	32.97
mod	Módulo (residuo)	$15 \text{ mod } 2$	1
div	División entera	$17 \text{ div } 3$	5



# Operadores aritméticos

```
#  
num1 = int(input("Dame el primer numero"))  
num2 = int(input("Dame el segundo numero"))  
div = num1 // num2  
residuo = num1 % num2  
print("La división entera es ",div)  
print("El residuo es ",residuo)
```

# Jerarquía de operadores aritméticos

Operador	Jerarquía	Operación
**	(mayor) ↓ (menor)	Potencia
*,/,mod,div		Multiplicación, división, módulo, división entera
+, -		Suma, resta

## Jerarquía de operadores aritméticos

```
2 #  
3 #  
4 base = int(input("Dame la base"))  
5 altura = int(input("Dame la altura"))  
6 area = base*altura/2  
7 print("El area es ",area)
```

# Jerarquía de operadores aritméticos

- Si una expresión contiene subexpresiones entre paréntesis, estas se evalúan primero.
- Los operadores aritméticos se aplican teniendo en cuenta la jerarquía y ***de izquierda a derecha***.

$$5 + 9 * 2 - 40 / 5$$

$$5 + 18 - 40 / 5$$

$$5 + 18 - 8$$

$$23 - 8 = 15$$

# Jerarquía de operadores aritméticos

Shell x

Python 3.7.2 (bundled)

```
>>> 5 + 9 * 2 - 40/5
```

```
15.0
```

```
>>> |
```

# Operadores relacionales

Operador	Operación	Ejemplo	Resultado
=	Igual que	'hola' = 'lola'	FALSO
<>	Diferente que	'a' <> 'b'	VERDADERO
<	Menor que	7 < 15	VERDADERO
>	Mayor que	22 > 11	VERDADERO
<=	Menor o igual que	30 <= 22	FALSO
>=	Mayor o igual que	20 >= 55	FALSO

# Operadores Relacionales

Los operadores relacionales se utilizan para comparar y regresan dos posibles valores: Verdadero o Falso

$$\underbrace{(7^{**}2)}_{49} > \underbrace{(25 * 2)}_{50}$$

Falso

# Operadores lógicos

Los operadores lógicos son utilizados para combinar declaraciones condicionales

Operator	Description	Example
not	Negación	<code>not(x &lt; 5 and x &lt; 10)</code>
or	Regresa Verdadero si al menos uno de los enunciados es Verdadero	<code>x &lt; 5 or x &lt; 4</code>
and	Regresa Verdadero si todos los enunciados son Verdaderos	<code>x &lt; 5 and x &lt; 10</code>





# Expresiones matemáticas

¿Cómo se traduce una  
fórmula matemática a  
código de programación?

# Expresiones matemáticas

Multiplicación implícita

$$4xyz \rightarrow 4 * x * y * z$$

Multiplicación con paréntesis

$$VP(1 + i) \rightarrow VP * (1 + i)$$

Divisiones

$$\frac{x + 2}{x - 2} \rightarrow (x + 2) / (x - 2)$$

Potencias (exponentes)

$$x^5 \rightarrow x ** 5$$

Raíces matemáticas

$$\sqrt{x} = x^{1/2} \rightarrow x ** 0.5$$

# Expresiones matemáticas

$$VF = VP (1 + i)^N$$

$$VF = VP * (1 + i) ** N$$