



**Business**  
School

# Iniciación a Python.

**Marta Ramírez**

Especialidad Data Science y Big data

---

- ¿Qué es Python?
- Instalación Python 3.X e IDE para su manejo. Características.
- Conceptos básicos.
- Tipos de datos, operadores de asignación, aritméticos, etc.
- Funciones predefinidas, math int float str.
- Lectura y escritura desde el terminal usando *print* e *input*.

# I. Introducción a Python

---

## Introducción

Python

### Características

- Lenguaje interpretado o de script.
- Tipado dinámico.
- Orientado a objetos.

# I. Introducción a Python

---

## Introducción

## Python

### Ide's

- Anaconda

Código abierto.

Paquetes Conda.

Administrador entornos.

# I. Introducción a Python

## Introducción

## Python

### Conceptos básicos

Para crear comentarios:

- a. de una línea
- b. de más de una línea “texto”.

Palabras reservadas:

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

# I. Introducción a Python

---

## Introducción

Python

### Conceptos básicos

#### Declaración de variables

Para asignar un valor a una variable se ejecutará la sentencia:

**variable = expresión**

# I. Introducción a Python

## Introducción

## Python

### Tipos de datos

En Python los tipos básicos se dividen en:

- Números (enteros, reales y complejos).
- Cadenas de texto.
- Valores booleanos.

Para saber el tipo de dato usamos la función **type()**:

```
nombre="hola"  
print(type(nombre)) nos devuelve str
```

# I. Introducción a Python

## Introducción

## Python

### Operaciones

Operación	Operador	Aridad	Asociatividad	Precedencia
Exponenciación	<b>**</b>	Binario	Por la derecha	1
Identidad	<b>+</b>	Unario	—	2
Cambio de signo	<b>-</b>	Unario	—	2
Multiplicación	<b>*</b>	Binario	Por la izquierda	3
División	<b>/</b>	Binario	Por la izquierda	3
Módulo (o resto)	<b>%</b>	Binario	Por la izquierda	3
Suma	<b>+</b>	Binario	Por la izquierda	4
Resta	<b>-</b>	Binario	Por la izquierda	4
Igual que	<b>==</b>	Binario	—	5
Distinto de	<b>!=</b>	Binario	—	5
Menor que	<b>&lt;</b>	Binario	—	5
Menor o igual que	<b>&lt;=</b>	Binario	—	5
Mayor que	<b>&gt;</b>	Binario	—	5
Mayor o Igual que	<b>&gt;=</b>	Binario	—	5
Negación	<b>not</b>	Unario	—	6
Conjunción	<b>and</b>	Binario	Por la izquierda	7
Disyunción	<b>or</b>	Binario	Por la izquierda	8

**Tabla** Características de los operadores Python. El nivel de precedencia 1 es el de mayor prioridad.



# I. Introducción a Python

## Introducción

### Python

#### Funciones predefinidas

##### **Funciones matemáticas:**

- `abs( )`: calcula el valor absoluto de un número.
- `float( )`: convierte un número o cadena numérica a flotante.
- `int( )`: convierte un número o cadena numérica entera a entero.
- `Str()`: convierte un numero en una cadena.
- `round( )`: redondeo. Puede usarse con uno o dos argumentos.

No necesitamos importar ninguna librería.

# I. Introducción a Python

## Introducción

## Python

### Funciones matemáticas

Necesitamos importar la librería math → `import math`

<code>sin(x)</code>	Seno de $x$ , que debe estar expresado en radianes.
<code>cos(x)</code>	Coseno de $x$ , que debe estar expresado en radianes.
<code>tan(x)</code>	Tangente de $x$ , que debe estar expresado en radianes.
<code>exp(x)</code>	El número $e$ elevado a $x$ .
<code>ceil(x)</code>	Redondeo hacia arriba de $x$ (en inglés, «ceiling» significa techo).
<code>floor(x)</code>	Redondeo hacia abajo de $x$ (en inglés, «floor» significa suelo).
<code>log(x)</code>	Logaritmo natural (en base $e$ ) de $x$ .
<code>log10(x)</code>	Logaritmo decimal (en base 10) de $x$ .
<code>sqrt(x)</code>	Raíz cuadrada de $x$ (del inglés «square root»).

# I. Introducción a Python

## Introducción

## Python

### Lectura y escritura

- Lectura: `variable = input ("mensaje")`. Todo lo que se lee por teclado es de tipo string.
- Castear el valor introducido:
  - `variable = int(input ("mensaje"))`
  - `variable = float(input ("mensaje"))`

# I. Introducción a Python

## Introducción

### Python

#### Lectura y escritura

Escritura de los datos:

- `print ("mensaje" + variable)`, si la variable es de tipo string.
- `print ("mensaje" + str(variable))`, si la variable no es de tipo string.
- `print ("mensaje", variable)`, si la variables es string o cualquier tipo de dato.

## Conclusiones

---

Para avanzar en la material **se debe realizar los casos prácticos**, tanto los que se han resuelto en clase como los que se dejan pendientes para su realización.