

# Iniciación con Python

Clase 02 - "Hola mundo"

# ¡Les damos la bienvenida!



Vamos a comenzar a grabar la  
clase

## Clase 01.

### ► Bienvenida

1. Introducción al curso.
2. Conceptos básicos: hardware, software, programa.
3. Concepto de algoritmo.
4. Introducción a Python

## Clase 02.

### ► Intro a Python

1. Visual Studio Code + Python.
2. "Hola Mundo" en Python.
3. Sintaxis básica de Python.
4. Variables.
5. Tipos de datos simples.

## Clase 03.

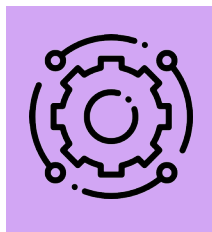
### ► Datos en Python

1. Conversión entre tipos de datos.
2. Operadores algebraicos.
3. Expresiones.
4. Uso de input()
5. Programas con entrada, procesamiento y salida de datos.

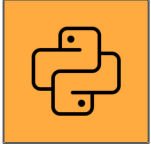


## ¿Ya viste la “**Introducción al programa**” disponible en el campus virtual?

La visualización y resolución de un breve cuestionario es de carácter obligatorio para desbloquear los contenidos de las primeras 2 clases



# Herramientas



# Instalación de Python

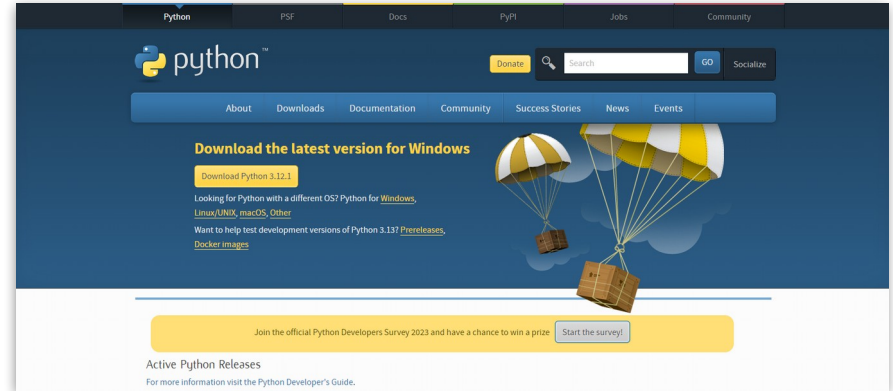
Necesitamos instalar el intérprete Python desde su página oficial.

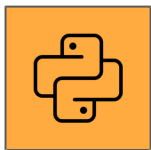
Ingresamos a Downloads y hacemos clic en el botón Download Python.

Asegúrate de descargar la última versión (o como mínimo la versión 3.8.x).



[Página oficial de Python](#)





# Instalación de Python

Procedemos con la instalación realizando los pasos típicos para instalar aplicaciones en nuestro Sistema Operativo.

**Nota:** Si usas Windows, recuerda instalar seleccionado la opción *"Add python.exe to PATH"*.





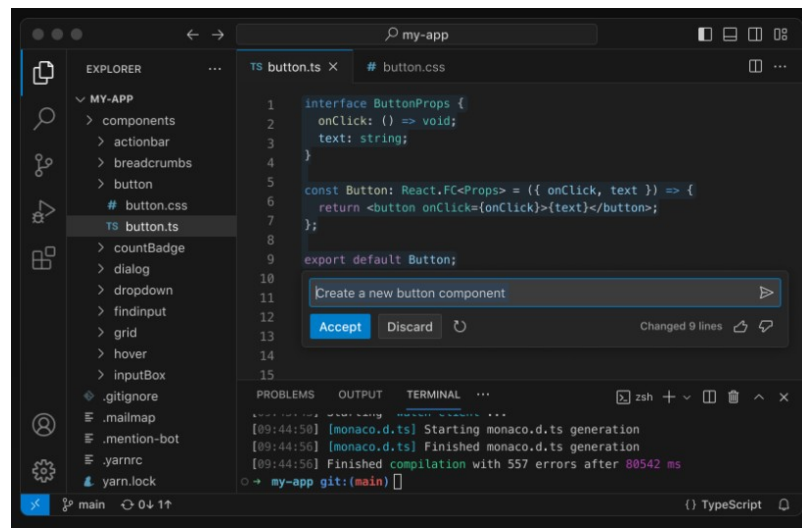
# Instalación de VSCode

Para escribir código utilizaremos un editor de texto llamado **Visual Studio Code**.

Lo descargamos desde su página oficial, y seguimos los pasos necesarios para su instalación, asegurándonos de crear un ícono de acceso en el escritorio o menú de nuestra computadora.



[Página oficial de VSCode](#)





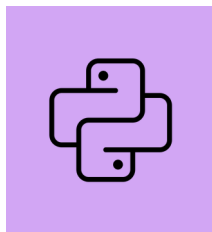


# Extensiones para VSCode

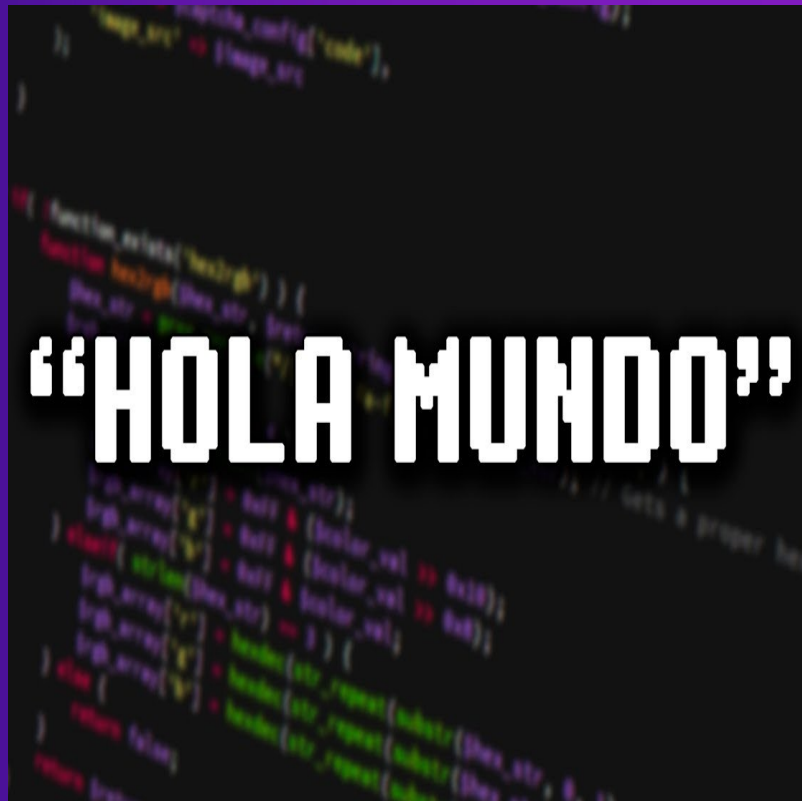


Una **extensión** es un módulo de software que agrega funcionalidades adicionales. Instalaremos dos:

1. **Spanish Language Pack for Visual Studio Code**, que traduce a nuestro idioma la interfaz del editor.
2. **Python extension for Visual Studio Code**, que proporciona una serie de herramientas específicas para programar con Python.



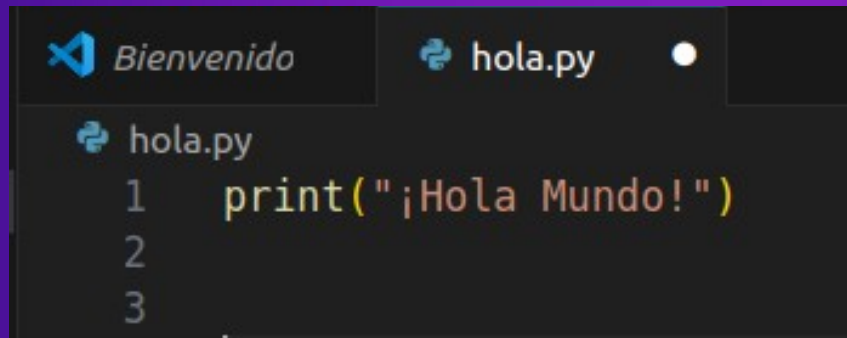
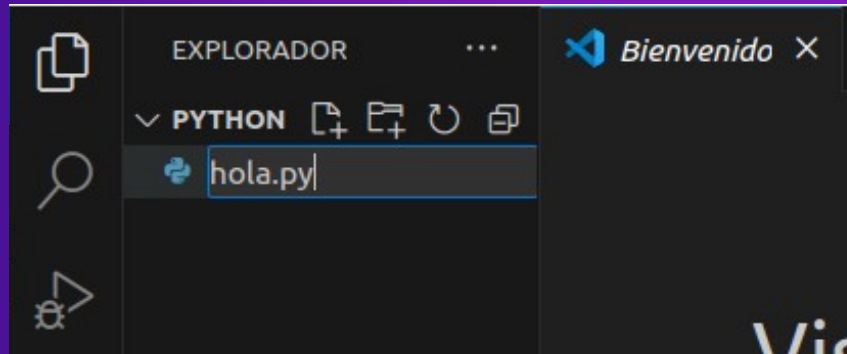
# ¡Hola, Mundo!



# HOLA MUNDO

Escribir un programa que muestre el texto **"Hola Mundo"** es una tradición en la programación y se trata de uno de los programas más simples que pueden escribir en cualquier lenguaje de programación.

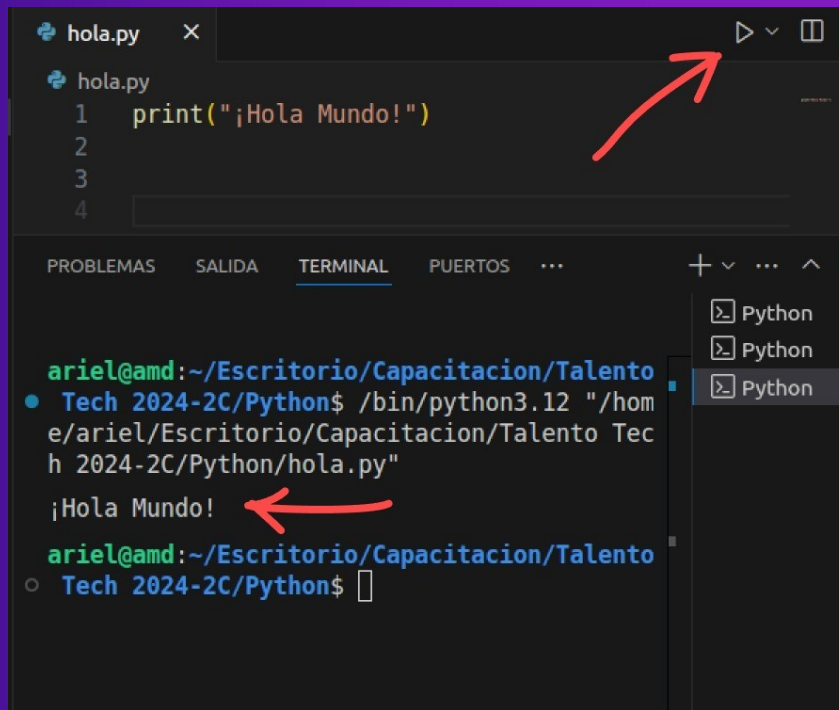
Comenzaremos a aprender un Python respetando esta tradición: escribiremos nuestro propio programa capaz de imprimir o mostrar en la pantalla el mensaje "Hola Mundo".



# HOLA MUNDO

Primero, creamos un nuevo archivo llamado "hola.py" en el explorador de VSCode, utilizando la herramienta que vemos en la imagen de la izquierda, o el menú de opciones "Archivo".

Luego, en la pestaña que aparece en el área de trabajo escribimos `print("¡Hola Mundo!")`. Ten cuidado en respetar exactamente la sintaxis que se vé en la imagen.



```
hola.py x
hola.py
1 print("¡Hola Mundo!")
2
3
4
```

PROBLEMAS SALIDA TERMINAL PUERTOS ...

```
ariel@amd:~/Escritorio/Capacitacion/Talento Tech 2024-2C/Python$ /bin/python3.12 "/home/ariel/Escritorio/Capacitacion/Talento Tech 2024-2C/Python/hola.py"
¡Hola Mundo!
ariel@amd:~/Escritorio/Capacitacion/Talento Tech 2024-2C/Python$
```

# HOLA MUNDO

El programa se ejecuta haciendo click en el ícono que aparece en la esquina superior derecha de la ventana de Visual Studio Code.

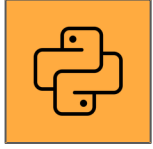
Vamos a ver que aparece una zona nueva en el editor: la terminal de Python. Esa terminal es la interfaz que utiliza Python para comunicarse con el usuario.

Como se ve en la imagen, aparece la ruta al archivo que hemos ejecutado, datos del usuario, y también la salida de nuestro programa: "¡Hola Mundo!"

A pesar de su sencillez, ¡este es un gran primer paso en nuestro camino a convertirnos en programadores!

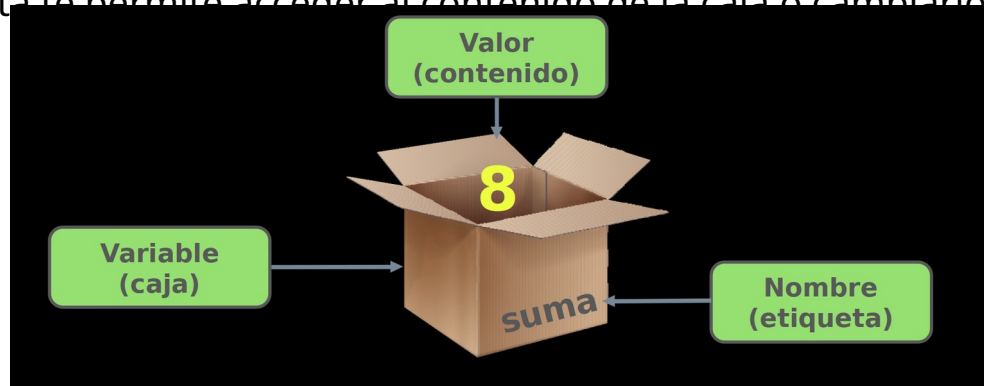


# Variables



# Variables en Python

En programación, una **variable** es como una "caja" en la memoria de tu computadora donde puedes guardar datos temporales que tu programa puede usar y modificar mientras se ejecuta. Piensa en una variable como en un contenedor que tiene una etiqueta con un nombre único; esta etiqueta te permite acceder al contenido de la caja o cambiarlo.





# Nombre de las variables

Los nombres de variables, funciones, etc, deben respetar las siguientes convenciones:

- Pueden ser cualquier combinación de letras (mayúsculas y minúsculas), dígitos y el carácter guión bajo (\_), pero no puede comenzar por un dígito. Se escriben en minúsculas, separando las palabras con el guión bajo.
- Se recomienda usar nombres que sean expresivos. Por ejemplo, **contador** es mejor que simplemente **c**.
- Python es "case sensitive" diferencia entre mayúsculas y minúsculas
- `and, as, assert, break, class, continue, def, del, elif, else, except, False, finally, for, from, global, if, import, in, is, lambda, None, nonlocal, not, or, pass, raise, return, True, try, yield, while, with`





# Convenciones de nombres

Algunos nombres de variables **válidos** y **recomendados**:

suma

total

imp\_final

\_saldo

area12

Algunos nombres de variables **válidos** pero **no recomendados**:

Suma

arearombo

ImpFinal

w123rt41

años

Algunos nombres de variables **no válidos** (Python reporta error):

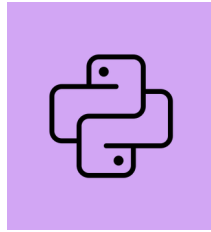
mi saldo

2pesos

for

21%IVA

\$a\_pagar



# print()



# La función print()

La función **print()** permite mostrar datos en la terminal. Recibe entre paréntesis lo que se conoce como **parámetros**, que pueden ser **variables** y/o **literales**, separados por comas. Las cadenas pueden delimitarse con comillas simples o dobles.

Luego de imprimir, print() realiza un salto de línea (pasa a la línea siguiente). Si se usa print() sin argumentos, sólo se muestra la línea en blanco:

```
print("Linea 1")  
print()  
print("Linea 2")
```

```
Linea 1  
  
Linea 2
```

Salida por pantalla

Argumento



# La función print()

En este ejemplo, además de la creación y asignación de valores a variables y el uso de algunos operadores, vemos como utilizar print() para mostrar literales y valores de variables en la terminal:

```
nombre = "Talentos Tech"  
a = 20  
b = 30  
promedio = (a + b) / 2  
  
print("Nombre:", nombre)  
print("La suma de", a, "y", b, "es", a+b)  
print("Promedio:", promedio)
```

Declaración  
de variables

Operaciones y  
asignación

```
Nombre: Talentos Tech  
La suma de 20 y 30 es 50  
Promedio: 25.0
```

Salida por pantalla



# La función print(): end

**end=' '** es un parámetro opcional de la función **print()** y determina qué se debe imprimir al final de cada llamada a la función. Por defecto es un carácter de nueva línea ('\n'), lo que significa que **cada llamada a print() agrega automáticamente una nueva línea** al final. Al cambiar end a una cadena nula ( ' ') con un espacio en blanco, modificamos su comportamiento:

```
print("¡Hola Mundo!", end=" ")  
print("Aprendiendo Python.")
```

```
¡Hola Mundo! Aprendiendo Python.
```

Salida por pantalla



# La función print(): \n

**\n** (nueva línea) representa un carácter especial que **se utiliza para insertar una nueva línea en una cadena de texto**. Cuando escribimos **\n** dentro de una cadena y la imprimimos, el texto que le sigue se coloca en una nueva línea:

```
print("Escribimos\nen\nvarias\nlíneas.")
```

```
Escribimos  
en  
varias  
líneas.
```

Salida por pantalla



# La función print(): \t

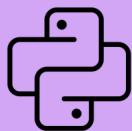
**\t** (tabulación) se utiliza para insertar un tabulador (o tabulación) en una cadena de texto. Cuando Python encuentra **\t** dentro de una cadena a imprimir, inserta un espacio equivalente al de una tabulación.

```
print("Nombre \tApellido \tEdad")  
print("Juan \tFernández \t32")
```

Nombre	Apellido	Edad
Juan	Fernández	32

Salida por pantalla

El comportamiento exacto de **\t** depende de la longitud de las cadenas que lo rodean. La tabulación agrega un número fijo de espacios, y si las cadenas son más largas o más cortas de lo esperado, la alineación puede verse afectada.



# Tipos de datos simples



# TIPOS DE DATOS.

En Python, un "tipo de dato" se refiere a la categoría o clasificación de los valores que una variable puede almacenar.

Cada valor en Python pertenece a un **tipo de dato específico**, y estos tipos de datos determinan cómo se almacenan los datos en la memoria y qué operaciones se pueden realizar en ellos.

Los tipos de datos en Python son fundamentales para el manejo de variables y la realización de operaciones en el código.



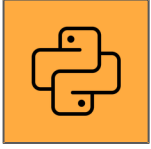
```
nombre = "Talento Tech" # string  
  
numero = 1234             # int  
  
importe = 35.75           # float  
  
activo = False            # bool
```

## TIPOS DE DATOS.

Python es un lenguaje de programación que **no requiere especificar el tipo de dato** de una variable al declararla.

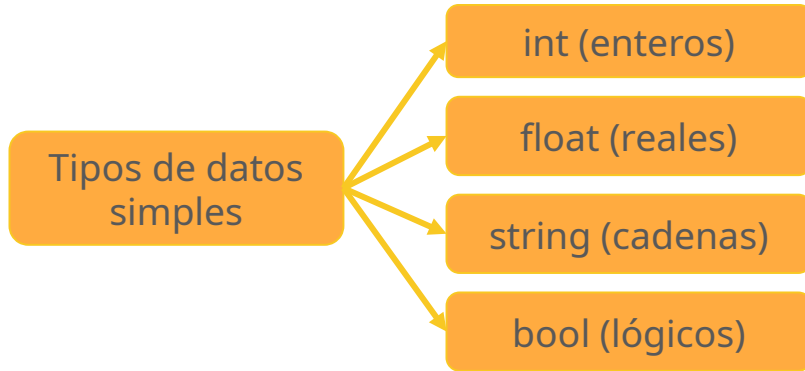
Python infiere automáticamente el tipo de dato en tiempo de ejecución según el valor que se le asigna a la variable. Esto proporciona flexibilidad y simplifica la escritura de código.

Recordemos que la forma de asignar el dato a una variable es a través del **operador de asignación "="**

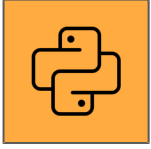


# Tipos de datos

Existen una gran variedad de datos en **Python**. Entre ellos algunos que se consideran "*tipos de datos básicos*":

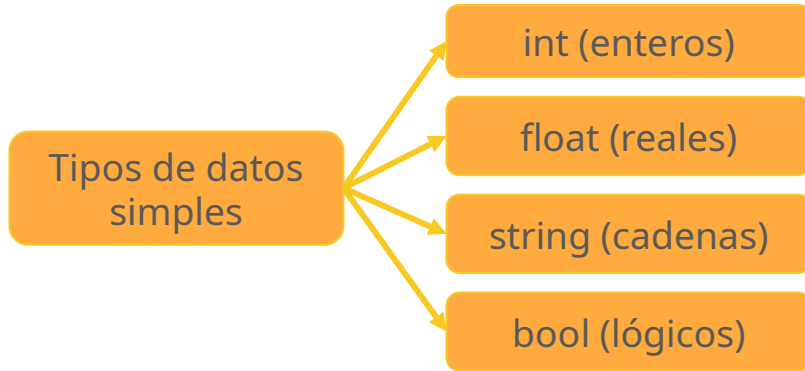


```
cadena = "Hola Mundo" # string
entero = 3              # int
flotante = 12.3        # float
logico = True          # boolean
```



# Tipos de datos

Existen una gran variedad de datos en **Python**. Entre ellos algunos que se consideran "*tipos de datos básicos*":



```
cadena = "Hola Mundo" # string
entero = 3              # int
flotante = 12.3        # float
logico = True          # boolean
```

# ¡Vamos a la práctica!



# Ejercicios Prácticos

## Optativos | No entregables



### Lista de la compra

Usando la instrucción `print()` y lo visto en la clase, escribir un programa que muestre en la terminal la lista de productos que necesitamos comprar en el supermercado.

Tips:

- Podés usar tabuladores para ordenar la lista.
- Recordá que si necesitas una línea en

### Ingreso promedio

Escribir un programa que guarde en variables el monto del ingreso de cada uno de los primeros 6 meses del año.

Luego, calcular y guardar en otra variable el promedio de esos valores.

Por último, mostrar una leyenda que diga "El ingreso promedio en el semestre es de xxxxx" donde "xxxxx" es el valor calculado.



# ¡NUEVO CUESTIONARIO EN CAMPUS!

La resolución del cuestionario es de carácter obligatorio para desbloquear los contenidos de las próximas 2 clases