

Iniciación con Python

Clase 03 - “Tipos de datos”

¡Les damos la bienvenida!



Vamos a comenzar a grabar la clase

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

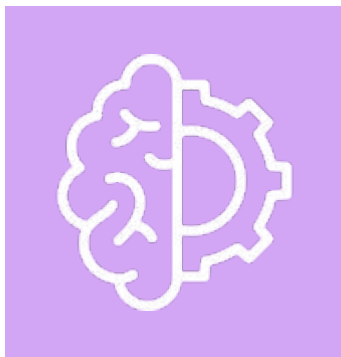
1. Conversión entre tipos de datos.
2. Operadores algebraicos.
3. Expresiones.
4. Uso de input()
5. Programas con entrada, procesamiento y salida de datos.

Clase **04.**

▶ Ruta de avance

1. Definimos los requisitos del Proyecto Integrador.
2. Menú de opciones.
3. Pedir, procesar y mostrar datos.

Pero antes...



¡Resolvamos los **ejercicios prácticos** de la clase anterior!



Conversión de tipos de datos

En ocasiones es necesario aplicar conversiones de valores entre tipos de datos para manipular los valores de forma diferente. Por ejemplo, para concatenar valores numéricos con cadenas o representar posiciones decimales en números que se iniciaron como valores enteros:

```
cadena = "1234"  
entero = 45  
flotante = 56.2  
  
int(cadena)      # convierte a entero (1234)  
float(cadena)    # convierte a coma flotante (1234.0)  
str(entero)      # convierte a cadena ("45")  
str(flotante)    # convierte a cadena ("56.2")
```

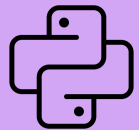


Obtención del tipo de dato

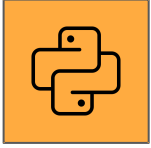
Se puede conocer el tipo de dato de cualquier objeto utilizando la función **type()**:

```
nombre = "Alan Turing"
edad = 38
notaFinal = 8.50
esEstudiante = False
domicilio = "Av. Talento Tech 1234"

print(type(nombre))      # <class 'str'>, cadena (string)
print(type(edad))        # <class 'int'>, entero (integer)
print(type(notaFinal))   # <class 'float'>, coma flotante (float)
print(type(esEstudiante)) # <class 'bool'>, lógico o booleano (boolean)
print(type(domicilio))   # <class 'str'>, cadena (string)
```



Operadores

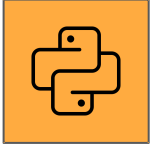


Operador de asignación

El operador de asignación = (igual) es muy importante en Python. Su función es diferente a la que habitualmente le damos en otros contextos, como la matemática.

Se lo denomina “operador de asignación” y permite asignar un valor a una variable. Su función consiste en evaluar (resolver) la expresión que se encuentra a su derecha, y asignar el resultado a la variable de la izquierda.

```
cantidad = 43          # int (enteros)
precio = 12.45         # float (reales)
nombre = "Adrián"      # string (cadenas)
encendido = True       # bool (lógicos)
```

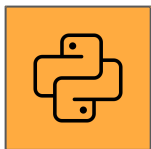



Expresiones y sentencias

Una **expresión** es una unidad de código que **devuelve un valor** y está formada por una combinación de operandos (variables y literales) y **operadores**.

```
5 + 2           # Suma del número 5 y el número 2
a < 10          # Compara si el valor de la variable a es menor que 10
b is None       # Compara si la identidad de la variable b es None
3 * (200 - c)   # Resta a 200 el valor de c y lo multiplica por 3
```

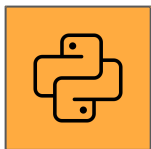
Una sentencia o declaración define una acción. Puede contener alguna(s) expresiones. Son las instrucciones que componen el código de un programa y determinan su comportamiento. Finalizan con un Enter.



Operadores aritméticos

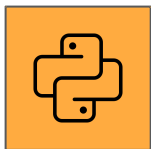
Realizan las operaciones aritméticas comunes. Requieren uno o dos operandos (operadores unarios o binarios). Se aplican las reglas de precedencia propias del álgebra:

Operador	Descripción	Ejemplo
+	Suma: Suma dos operandos.	<pre>suma = 12 + 5 print(suma) # 17</pre>
-	Resta: Resta al operando de la izquierda el valor del operando de la derecha. Utilizado sobre un único operando, le cambia el signo.	<pre>resta = 23 - 4 print(resta) # 19 numero = -32 print(numero) # -32</pre>
*	Multiplicación: Producto de dos operandos.	<pre>multiplicacion = 5 * 8 print(multiplicacion) # 40</pre>



Operadores aritméticos

Operador	Descripción	Ejemplo
/	División: Divide el operando de la izquierda por el de la derecha (el resultado siempre es un float).	<pre>division = 8 / 2 print(division) # 4.0 division2 = 14 / 5 print(division2) # 2.8</pre>
%	Operador módulo: Obtiene el resto de dividir el operando de la izquierda por el de la derecha. uno de sus usos es para saber si un número es par o impar	<pre>modulo = 8 % 2 print(modulo) # 0 (8 es par) modulo = 9 % 2 print(modulo) # 1 (9 es impar)</pre>
//	División entera: Obtiene el cociente entero de dividir el operando de la izquierda por el de la derecha.	<pre>divEntera = 10 // 6 divEntera2 = 25 // 4 print(divEntera) # 1 print(divEntera2) # 6</pre>

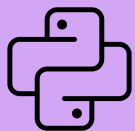


Operadores aritméticos

Operador	Descripción	Ejemplo
**	Potencia: El resultado es el operando de la izquierda elevado a la potencia del operando de la derecha.	<pre>potencia = 2 ** 3 potencia2 = 10 ** 2 print(potencia) # 8 print(potencia2) # 100</pre>

Para la **raíz cuadrada** Python no tiene operador aritmético en particular, pero podemos utilizar ** con un exponente de 0.5 (para raíz cuadrada) o fracción (para raíz cuadrada u otras raíces).

```
raiz = 9 ** 0.5
print(raiz) # 3.0
raiz = 9 ** (1/2)
print(raiz) # 3.0
raiz = 125 ** (1/3)
print(raiz) # 5.0
```



Función input()



La función input()

La función **input()** proporciona un mecanismo para que el usuario introduzca datos en nuestro programa. Muestra el cursor en la terminal, lee lo que se escribe, y cuando se presiona Enter, este contenido, en formato de cadena de caracteres, se puede asignar a una variable.

```
nombre = input("Ingrese su nombre: ")
```

Variable que recibe
el valor ingresado

Función input

Mensaje para ayudar al
operador



La función input()

Dado que **input()** devuelve únicamente valores tipo string, es necesario realizar una conversión a algún formato numérico si se requiere operar matemáticamente con esos valores. Para ello, usamos las funciones **int()** y **float()**:

```
num1 = input("Ingrese un número: ")  
numero = float(num1)  
resultado = numero * 2  
print(numero, "x 2 =", resultado)
```

Ingrese un número: 20
20.0 x 2 = 40.0

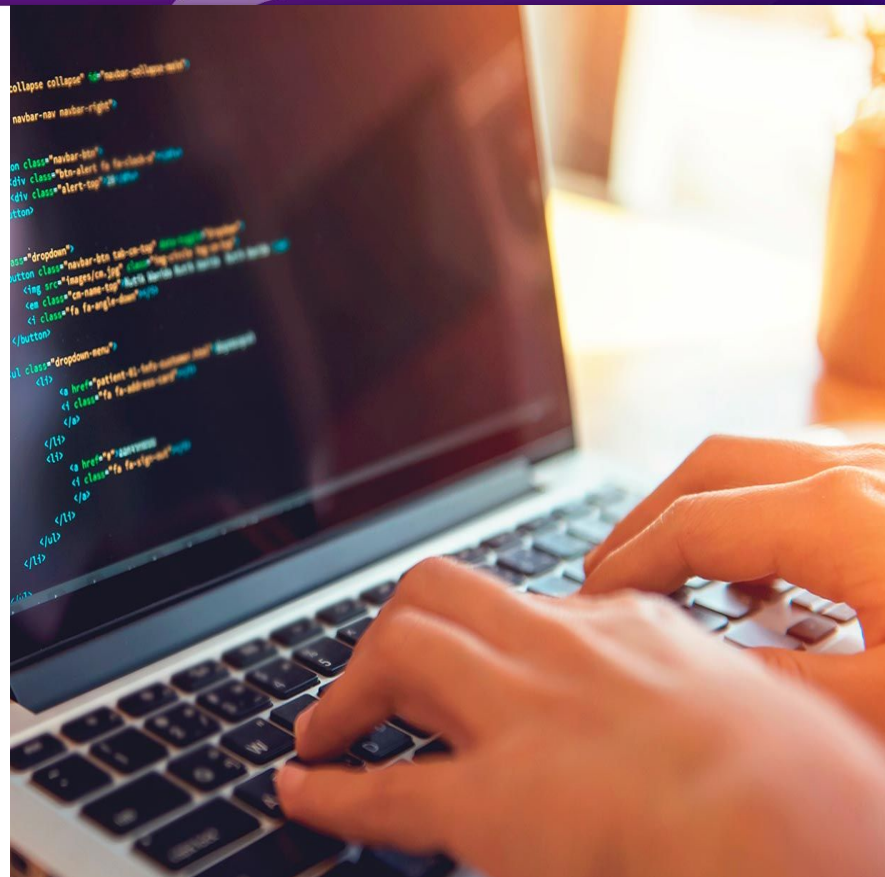
La cadena ingresada en el **input()** se convierte en un número de coma flotante, se almacena en “numero”, se guarda en “resultado” su producto con 2, y se muestra en la terminal usando **print()**.

ENTRADA, PROCESO, SALIDA

Ya tenemos los elementos necesarios para escribir programas en Python que puedan cumplir las tres etapas que requiere un algoritmo: ingreso de datos, procesamiento y salida.

Esto es posible utilizando **input()** para realizar la petición de datos al usuario, **variables** y **operadores aritméticos** para realizar el procesamiento de esos datos, y finalmente la función **print()** para proporcionar los datos de salida al usuario.

Veamos algunos ejemplos...





Suma de dos números

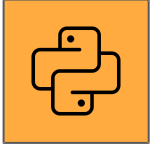
Este ejemplo simple ilustra cómo se pueden utilizar las funciones `input()`, operadores aritmético y `print()` para escribir un programa simple que suma dos números proporcionados por el usuario:

```
numero1 = float(input("Ingresa el primer número: "))
numero2 = float(input("Ingresa el segundo número: "))

suma = numero1 + numero2

print("La suma de", numero1, "y", numero2, "es", suma)
```

```
Ingresa el primer número: 12
Ingresa el segundo número: 7
La suma de 12.0 y 7.0 es 19.0
```



Suma de dos números

Explicación del programa:

- **Ingreso de datos:** El programa solicita al usuario que ingrese dos números. Los valores ingresados se almacenan en las variables llamadas “numero1” y “numero2”.
- **Proceso:** El programa realiza la suma de los dos números utilizando el operador + y guarda el resultado en la variable llamada “suma”.
- **Salida de datos:** Finalmente, el programa muestra el resultado de la suma utilizando print() con el formato necesario para que claro y fácil de entender.



Promedio de tres notas

Aquí vemos un ejemplo un poco más complejo que realiza el cálculo del promedio de tres notas:

```
nota1 = float(input("Ingresa la primera nota: "))
nota2 = float(input("Ingresa la segunda nota: "))
nota3 = float(input("Ingresa la tercera nota: "))

promedio = (nota1 + nota2 + nota3) / 3

print()
print("El promedio de las tres notas es:", promedio)
```

```
Ingresa la primera nota: 8
Ingresa la segunda nota: 6
Ingresa la tercera nota: 4
```

```
El promedio de las tres notas es: 6.0
```



Promedio de tres notas

Explicación del programa:

- **Ingreso de datos:** El programa solicita al usuario que ingrese tres notas, que se almacenan en las variables “nota1”, “nota2”, y “nota3”.
- **Proceso:** Se calcula el promedio de las tres notas sumando los valores y dividiendo por 3. El resultado se guarda en la variable “promedio”.
- **Salida de datos:** El promedio calculado se muestra en la terminal.

¡Vamos a la práctica! 🚀



Ejercicios Prácticos



Optativos | No entregables

Operaciones básicas

1. Crea un programa que solicite al usuario dos números enteros.
2. Realiza las siguientes operaciones: suma, resta, multiplicación, y módulo.
3. Muestra el resultado de cada operación en un formato claro y amigable.

Asegúrate de incluir mensajes personalizados que expliquen cada resultado, por ejemplo: "La suma de tus números es: X".

Calculadora de propinas

Escribe un programa en Python que calcule la propina que se debe dejar en un restaurante.

El script debe solicitar al usuario el monto total de la cuenta y el porcentaje de propina que desea dejar.

Utilizando operadores aritméticos, calcula la cantidad de propina y el total a pagar (incluyendo la propina). Finalmente, muestra los resultados en la pantalla.