

20200528

May 28, 2020

1 Input

```
[1]: # Capturar el nombre del usuario
print("Hola, ¿cómo te llamas?")
nombre = input()
print(f"Hola, tu nombre es {nombre}")
```

Hola, ¿cómo te llamas?

Alexander

Hola, tu nombre es Alexander

```
[2]: nombre = input("Hola, ¿cuál es tu nombre? ")
print(f"Hola, tu nombre es {nombre}")
```

Hola, ¿cuál es tu nombre? Felipe

Hola, tu nombre es Felipe

```
[3]: edad = input("Hola, ¿cuál es tu edad?")
type(edad)
nacimiento = edad - 2020
```

Hola, ¿cuál es tu edad? 89

```

      □
↳ -----
TypeError                                Traceback (most recent call↳
↳ last)
```

```
<ipython-input-3-0486c96ff752> in <module>
      1 edad = input("Hola, ¿cuál es tu edad?")
      2 type(edad)
----> 3 nacimiento = edad - 2020
```

TypeError: unsupported operand type(s) for -: 'str' and 'int'

```
[4]: edad = input("Hola, ¿cuál es tu edad?")
     type(edad)
```

Hola, ¿cuál es tu edad? 89

[4]: str

```
[5]: edad = int(edad)
     type(edad)
```

[5]: int

```
[6]: nacimiento = 2020 - edad
     nacimiento
```

[6]: 1931

```
[8]: edad = int(input("Hola, ¿cuál es tu edad?"))
     type(edad)
     print(f"Tu edad es {edad}")
```

Hola, ¿cuál es tu edad? b

```

      □
↪-----

      ValueError                                Traceback (most recent call
↪last)

      <ipython-input-8-c34e893a9f14> in <module>
----> 1 edad = int(input("Hola, ¿cuál es tu edad?"))
      2 type(edad)
      3 print(f"Tu edad es {edad}")
```

ValueError: invalid literal for int() with base 10: 'b'

1.1 Funciones

```
[12]: def sumar(a, b):
      """Función para realizar la suma de dos números"""
      return a + b
```

```
[14]: resultado = sumar(58, 12)
      resultado
```

```
[14]: 70
```

```
[15]: sumar(89, 'hola')
```

```

      □
↳ -----

      TypeError                                Traceback (most recent call↳
↳ last)

      <ipython-input-15-d21b112c8aed> in <module>
      ----> 1 sumar(89, 'hola')

      <ipython-input-12-d6d4d771a974> in sumar(a, b)
           1 def sumar(a, b):
           2     """Función para realizar la suma de dos números"""
      ----> 3     return a + b

      TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
[16]: sumar("hola", "mundo")
```

```
[16]: 'holamundo'
```

```
[17]: def Saludar():
      print("Hola, bienvenido a la clase de Python")

      Saludar()
```

```
Hola, bienvenido a la clase de Python
```

```
[18]: def Saludar():
      return "Hola, bienvenido a la clase de Python"

      print(f"{Saludar()}")
```

```
Hola, bienvenido a la clase de Python
```

```
[19]: def sumar(a, b):
      return a + b
```

```
[20]: sumar(b=89, a=25)
```

```
[20]: 114
```

```
[21]: def restar(a, b):  
      return a - b
```

```
[22]: restar(12, 8)
```

```
[22]: 4
```

```
[23]: restar(b=4, a=12)
```

```
[23]: 8
```

```
[24]: restar(b=50, a=5)
```

```
[24]: -45
```

```
[25]: def saludar(nombre=None):  
      if nombre is None:  
          return "Hola, bienvenido a la clase de Python"  
      else:  
          return f"Hola {nombre}, bienvenido a la clase de Python"
```

```
[26]: saludar()
```

```
[26]: 'Hola, bienvenido a la clase de Python'
```

```
[27]: saludar("pepito")
```

```
[27]: 'Hola pepito, bienvenido a la clase de Python'
```

```
[28]: saludar(58)
```

```
[28]: 'Hola 58, bienvenido a la clase de Python'
```

1.2 Funciones con argumentos indeterminados por posición

```
[29]: def indeterminado(*args):  
      for arg in args:  
          print(arg)
```

```
[31]: indeterminado("Hola", 8.9, 5, [9, 8, 7], ("Mundo", "saludo"), True)
```

```
Hola  
8.9
```

```
5
[9, 8, 7]
('Mundo', 'saludo')
True
```

```
[32]: def diccionario(**kwargs):
      for kwarg in kwargs:
          print(f"{kwarg} => {kwargs[kwarg]}")
```

```
[33]: diccionario(validar=False, nombres=["Edwin", "Camila", "Eliana"], edades=(33, 22, 28))
```

```
validar => False
nombres => ['Edwin', 'Camila', 'Eliana']
edades => (33, 22, 28)
```

1.3 Funciones lambda

```
[34]: def sumar(a, b):
      return a + b

suma = lambda a, b : a + b
```

```
[35]: sumar(12, 5)
```

```
[35]: 17
```

```
[36]: suma(12, 5)
```

```
[36]: 17
```

```
[37]: restar = lambda a, b : a - b
```

```
[38]: restar(56, 12)
```

```
[38]: 44
```

```
[39]: restar(b=56, a=12)
```

```
[39]: -44
```

```
[40]: mayor = lambda a, b : a if a > b else b
```

```
[41]: mayor(-12, 12)
```

```
[41]: 12
```

```
[44]: def mayor_def(a, b):  
      if a > b:  
          return a  
      else:  
          return b
```

```
[45]: mayor_def(-12, 12)
```

```
[45]: 12
```

1.4 Referencia

```
[49]: def doblar(num):  
      num *= 2  
      return num
```

```
[53]: n = 10  
      print(n)
```

```
10
```

```
[51]: doblar(n)
```

```
[51]: 20
```

```
[52]: print(n)
```

```
10
```

```
[56]: print(f"{n} => {doblar(n)} => {n}")
```

```
10 => 20 => 10
```

```
[57]: n = 10  
      print(n)  
      doblar(n)  
      print(n)
```

```
10
```

```
10
```

```
[ ]: def dividir(a, b):  
      resultado
```

```
[58]: a = 1  
      id(a)
```

```
[58]: 140705022058896
```