

1. Tipos de Datos Básicos

Tipo	Ejemplo	Descripción
int	123	Números enteros.
float	3.14	Números decimales (flotantes).
str	"Hola"	Cadenas de texto.
bool	True , False	Valores lógicos.
list	[1, 2, 3]	Lista (mutable, ordenada).
tuple	(1, 2, 3)	Tupla (inmutable, ordenada).
dict	{"a": 1, "b": 2}	Diccionario (clave-valor).
set	{1, 2, 3}	Conjunto (únicos, no ordenados).

2. Operadores

Aritméticos:

Operador	Ejemplo	Resultado
+	5 + 3	8
-	5 - 3	2
*	5 * 3	15
/	5 / 2	2.5
//	5 // 2	2 (división entera)
%	5 % 2	1 (módulo)
**	2 ** 3	8 (potencia)

Lógicos:

Operador	Descripción	Ejemplo	Resultado
and	Ambos verdaderos	True and False	False
or	Al menos uno verdadero	True or False	True
not	Negación	not True	False

Comparación:

Operador	Descripción	Ejemplo	Resultado
==	Igual a	5 == 5	True
!=	No igual a	5 != 3	True
>	Mayor que	5 > 3	True
<	Menor que	5 < 3	False
>=	Mayor o igual	5 >= 3	True
<=	Menor o igual	5 <= 3	False

3. Estructuras de Control

Condicionales:

python Copiar código

```
if condición:
    # código si es True
elif otra_condición:
    # código si otra_condición es True
else:
    # código si ninguna condición es True
```

Bucles:


- for :

python Copiar código

```
for elemento in iterable:
    # código para cada elemento
```

Ejemplo:


python

 Copiar código

```
for i in range(5): # range(start=0, stop=5, step=1)
    print(i) # 0, 1, 2, 3, 4
```

- **while :**

python

 Copiar código

```
while condición:
    # código mientras condición sea True
```

break , **continue** , **pass** :


Palabra clave	Descripción
break	Termina el bucle inmediatamente.
continue	Salta a la siguiente iteración del bucle.
pass	No hace nada (útil como marcador de posición).



4. Funciones

Definición de funciones:


python

 Copiar código

```
def nombre_función(param1, param2=valor_por_defecto):
    # código
    return valor # opcional
```

Ejemplo:

python

 Copiar código


```
def suma(a, b=0):
    return a + b

print(suma(5, 3)) # 8
print(suma(5))   # 5
```

lambda :

Funciones anónimas de una sola línea:

python

 Copiar código


```
doble = lambda x: x * 2
print(doble(4)) # 8
```



5. Manejo de Listas

Operaciones básicas:


python

 Copiar código

```
lista = [1, 2, 3]
lista.append(4)      # Agrega al final
lista.pop()          # Elimina el último
lista.remove(2)       # Elimina el valor 2
lista[1]              # Acceso al índice 1
lista[1:3]            # Sublista
```

Comprensión de listas:


python

 Copiar código

```
nueva_lista = [x**2 for x in range(5)]
print(nueva_lista) # [0, 1, 4, 9, 16]
```

6. Manejo de Diccionarios


python

 Copiar código

```
dic = {"clave1": "valor1", "clave2": "valor2"}
dic["clave1"]      # Acceso al valor
dic["clave3"] = "valor3" # Agregar clave-valor
del dic["clave2"]   # Eliminar clave
```

7. Entrada y Salida

python


 Copiar código

```
# Entrada
nombre = input("¿Cuál es tu nombre? ")

# Salida
print(f"Hola, {nombre}!") # f-strings
```

8. Módulos y Bibliotecas

python


 Copiar código

```
import math
print(math.sqrt(16)) # 4.0

from random import randint
print(randint(1, 10)) # Número aleatorio entre 1 y 10
```

9. Manejo de Archivos

python


 Copiar código

```
with open("archivo.txt", "r") as archivo:
    contenido = archivo.read()

with open("archivo.txt", "w") as archivo:
    archivo.write("Hola, mundo!")
```

10. Excepciones

python


 Copiar código

```
try:
    # Código que puede causar un error
    x = 1 / 0
except ZeroDivisionError:
    print(";No puedes dividir entre 0!")
finally:
    print("Bloque final siempre se ejecuta.")
```

11. Métodos Útiles

Cadenas (`str`):


python

 Copiar código

```
texto = "Hola Mundo"
texto.lower()      # "hola mundo"
texto.upper()      # "HOLA MUNDO"
texto.split()      # ["Hola", "Mundo"]
" ".join(["Hola", "Mundo"]) # "Hola Mundo"
texto.replace("Hola", "Adiós") # "Adiós Mundo"
```

Listas:


python

 Copiar código

```
lista = [3, 1, 4, 1, 5]
sorted(lista)      # [1, 1, 3, 4, 5]
len(lista)         # 5
sum(lista)         # 14
```

12. Trucos con `range()`

python

 Copiar código

```
range(5)           # [0, 1, 2, 3, 4]
range(1, 6)        # [1, 2, 3, 4, 5]
range(1, 10, 2)    # [1, 3, 5, 7, 9] (con saltos de 2)
```