

#1 Data Types

Int	-973,2103654
Str	"Hola, esto es un String"
Acceso a Secuencias	"Esto es un string [aquí accedes a un índice]"
Conversión	int("45") #Convierte a entero
	str(912) #Convierte a cadena

#2 Object References

No hay variables en Python como tal sino referencias de objetos

```
x = "blue"
```

```
y = "green"
```

```
z = x
```

El operador = enlaza una referencia a un objeto, no copia el objeto

#3 Collection Data Types

Tuplas Son inmutables y se crean con paréntesis o comas

```
("Denmark", "Finland")
```

Listas Son mutables y se crean con corchetes

```
[1,4,9]
```

Ejemplo de acceso y mutación de una lista

```
x = ['zebra', 49, -879]
```

```
x[1] = "forty nine"
```

Métodos comunes de listas

append Añade un elemento al final

insert Inserta un índice específico

remove Elimina un elemento

#4 Logical Operations

Operador de identidad

is Verifica si dos referencias apuntan al mismo objetivo

```
a is b
```

True si a y b son el mismo objeto

Operadores de comparación

< Menor que

```
3 < 5
```

!= Desigualdad

```
5 != 3
```

<= Menor o igual que

```
5 <= 5
```

== Igualdad

```
5 == 5
```

>= Mayor o igual que

```
5 >= 5
```

> Mayor que

```
5 > 3
```

Operador de pertenencia

in Verifica si un elemento está en una colección

```
'J' in 'Jung Chan'
```

```
#True
```

Operadores Lógicos

and Devuelve True si ambas condiciones son verdaderas

```
(5 > 3) and (8 > 6)
```

or Devuelve True si al menos una de las condiciones es verdadera

```
5 > 3 or 8 < 6
```

not Invierte el valor lógico

```
not 5 > 3
```

#5 Control Flow Statements

#5 Control Flow Statements (cont)

While

```
while True:
```

```
item = get_next_item()
```

```
if not item:
```

```
break
```

```
process_item(item)
```

for...in

```
for country in ["Denmark", "Norway"]:
```

```
print(country)
```

#6 Arithmetic Operators

Operadores básicos

+ 9 + 2 output 11

- 9 - 2 output 7

***** 9 * 2 output 18

/ 9 / 2 output 4.5

// 9 // 2 output 4

% 9 + 2 output 1

Operadores aumentados

Son una forma de realizar operaciones y asignarlas al mismo tiempo

+=, -=, *=

Ejemplo

```
x = 5
```

```
x += 3
```

Es equivalente a x = x + 3,

Resultado: 8

#7 Input / Output

input Captura datos del usuario

```
name = input("Enter name: ")
```

print Muestra salida en la consola

```
print("Total: ", total)
```

#8 Creating and calling functions

#8 Creating and calling functions (cont)

```
print(mensaje)
```

Función sin parámetros

```
def despedir():
```

```
return "Adiós"
```

```
print(despedir())
```

Función con varios parámetros

```
def sumar(a, b):
```

```
return a + b
```

```
resultado = sumar(3, 5)
```

Función sin return

```
def imprimir_saludo(nombre):
```

```
print("Hola, " + nombre)
```

```
imprimir_saludo("Ana")
```

if	Función
<pre>if lines < 1000:</pre>	Bloque reutilizable de código
<pre>print(" sma ll")</pre>	que realiza una tarea específica
<pre>elif lines < 10000:</pre>	<i>Ejemplo</i>
<pre>print(" med ium ")</pre>	<pre>def saluda r(n ombre):</pre>
<pre>else:</pre>	<pre>return " Hola, " + nombre</pre>
<pre>print(" lar ge")</pre>	<i>llamada a la función</i>
	<pre>mensaje = saluda r("C arl o s")</pre>



By **schmidtlady16**

Published 23rd October, 2024.

Last updated 23rd October, 2024.

Page 1 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

cheatography.com/schmidtlady16/