

#1 Data Types	#4 Logical Operations	#5 Control Flow Statements (cont)	#8 Creating and calling functions (cont)
<p>Int -973,2103654</p> <p>Str "Hola, esto es un String"</p> <p>Acceso a Secuencias "Esto es un string [aquí accedes a un índice]"</p> <p>Conversión int ("45") #Convierte a entero</p> <p> str(912) #Convierte a cadena</p>	<p>Operador de identidad</p> <p>is Verifica si dos referencias apuntan al mismo objetivo</p> <p> a is b</p> <p><i>True si a y b son el mismo objeto</i></p> <p>Operadores de comparación</p> <p>< Menor que</p> <p> 3 < 5</p> <p>!= Desigualdad</p> <p> 5 != 3</p> <p><= Menor o igual que</p> <p> 5 <= 5</p> <p>== Igualdad</p> <p> 5 == 5</p> <p>>= Mayor o igual que</p> <p> 5 >= 5</p> <p>> Mayor que</p> <p> 5 > 3</p> <p>Operador de pertenencia</p> <p>in Verifica si un elemento está en una colección</p> <p> 'J' in 'Jung Chan'</p> <p>#True</p> <p>Operadores Lógicos</p> <p>and Devuelve True si ambas condiciones son verdaderas</p> <p> (5 > 3) and (8 > 6)</p> <p>or Devuelve True si al menos una de las condiciones es verdadera</p> <p> 5 > 3 or 8 < 6</p> <p>not Invierte el valor lógico</p> <p> not 5 > 3</p>	<p>While</p> <p>while True:</p> <p>item = get_next_item()</p> <p>if not item:</p> <p>break</p> <p>process_item(item)</p> <p>for...in</p> <p>for country in ["Denmark", "Norway"]:</p> <p>print(country)</p>	<p><code>print(mensaje)</code></p> <p>Función sin parámetros</p> <p><code>def despedir():</code></p> <p> <code>return "Adiós"</code></p> <p> <code>print(despedir())</code></p> <p>Función con varios parámetros</p> <p><code>def sumar(a, b):</code></p> <p> <code>return a + b</code></p> <p> <code>resultado = sumar(3, 5)</code></p> <p>Función sin return</p> <p><code>def imprimir_saludo(nombre):</code></p> <p> <code>print("Hola, " + nombre)</code></p> <p> <code>imprimir_saludo("Ana")</code></p>
#2 Object References	#6 Arithmetic Operators	#7 Input / Output	
<p>No hay variables en Python como tal sino referencias de objetos</p> <p><code>x = "blue"</code></p> <p><code>y = "green"</code></p> <p><code>z = x</code></p> <p>El operador = enlaza una referencia a un objeto, no copia el objeto</p>	<p>Operadores básicos</p> <p>+ 9 + 2 output 11</p> <p>- 9 - 2 output 7</p> <p>* 9 * 2 output 18</p> <p>/ 9 / 2 output 4.5</p> <p>// 9 // 2 output 4</p> <p>% 9 + 2 output 1</p> <p>Operadores aumentados</p> <p>Son una forma de realizar operaciones y asignarlas al mismo tiempo</p> <p>+=, -=, *=</p> <p><i>Ejemplo</i></p> <p><code>x = 5</code></p> <p><code>x += 3</code></p> <p><i>Es equivalente a x = x + 3, Resultado: 8</i></p>	<p>input Captura datos del usuario</p> <p><code>name= input("Enter name: ")</code></p> <p>print Muestra salida en la consola</p> <p><code>print("Total: ", total)</code></p>	
#3 Collection Data Types	#5 Control Flow Statements		
<p>Tuplas Son inmutables y se crean con paréntesis o comas</p> <p> ("Denmark", "Finland")</p> <p>Listas Son mutables y se crean con corchetes</p> <p> [1,4,9]</p> <p><i>Ejemplo de acceso y mutación de una lista</i></p> <p><code>x = ['zebra', 49, -879]</code></p> <p><code>x[1] = "forty nine"</code></p> <p>Métodos comunes de listas</p> <p>append Añade un elemento al final</p> <p>insert Inserta un índice específico</p> <p>remove Elimina un elemento</p>	<p>#5 Control Flow Statements</p>		
		#8 Creating and calling functions	

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 **Readable.com**

Measure your website readability!

<https://readable.com>

cheatography.com/schmidtlady16/