



Guía rápida de Python

LO BÁSICO

APLICACIONES EN GENERAL Y ESPECIALIZADAS

Los espacios en blanco importan. Tu código no se ejecutará correctamente si utilizas una indentación errónea. Por ejemplo, si usas **SUBLIME TEXT** cómo editor, te sugiero empezar grabando el archivo con la extensión **.py**, así el editor se encargará de los espaciados.

```
# Esto es un comentario
```

Lógica básica

Condicional "if"

```
if test:           # realizar acción si test es True
elif test2:       # realizar acción si test2 es True
else:             # realizar acción si las anteriores
                 # son falsas (test y test2)
```

Bucle "while"

```
while test:       # realizar acción mientras test
                 # sea True
```

Bucle "for"

```
for item in secuencia: # realizar acción para cada miembro
                     # (item) de secuencia.
for i in range(10):    # realizar acción 10 veces (0 a 9)
for i in range(5, 10): # realizar acción 5 veces (5 a 9)
```

Cadenas

Una cadena es una secuencia de caracteres, generalmente utilizada para almacenar texto.

Creación:	cadena = "Hola mundo."		
Acceso:	cadena[4]	retorna:	'a'
División:	cadena.split(' ')	retorna:	['Hola', 'mundo.']
División:	cadena.split('n')	retorna:	['Hola mu', 'do.']

Para unir una lista de cadenas, utiliza la función `join()` como método de la cadena que utilizarás para separar cada uno de los elementos de la lista (o bien ").

```
palabras = ['Esto', 'es', 'una', 'lista', 'de', 'cadenas']
```

" ".join(palabras)	retorna "Esto es una lista de cadenas"
'ZOOl'.join(palabras)	retorna "EstoZOOllesZOOlLunaZOOlListaZOOldeZOOlcadenas"
".join(palabras)	retorna "Estoesunalistadecadenas"

Formateo de cadenas: similar a la función de C `printf()`, utiliza en su lugar el operador `%` para añadir los elementos de una tupla a una cadena.

```
cadena = "Python"           print("Hola %s." % cadena)   retorna: "Hola Python."
```



Tuplas

Una tupla consiste en un conjunto de valores separados por comas. Son útiles para pares ordenados y retornar varios valores desde una función.

Creación:

```
tupla = ()

otra_tupla = ("spam",) # Nótese la coma

esta_tupla = 12, 89, 'a', True

esta_tupla = (12, 89, 'a', True)
```

```
Acceso:      esta_tupla[0]      retorna:      12
            esta_tupla[3]      retorna:      True
```

Diccionarios

Un diccionario es un conjunto de pares clave:valor (o nombre:valor). Todos los nombres deben ser únicos.

```
Creación:      diccionario = {}
              diccionario = {'a':1, 'b':23, 'c':"huevos"}
```

```
print(diccionario['a'])      retorna:      1
print(diccionario.items())  retorna:      dict_items([('a', 1), ('b', 23), ('c', 'huevos')])
print(diccionario.values()) retorna:      dict_values([1, 23, 'huevos'])
print(diccionario.keys())   retorna:      dict_keys(['a', 'b', 'c'])
print("a" in diccionario)   retorna:      True
```

Listas

Una de las estructuras de datos más importantes en Python son las listas. Además de ser muy flexibles, cuentan con varias funciones de control.

	<u>Código</u>	<u>Valor de retorno</u>	<u>Contenido actual de la lista</u>
Creación	<code>lista = [5, 3, 'p', 9, 'e']</code>		<code>[5, 3, 'p', 9, 'e']</code>
Acceso	<code>lista[0]</code>	<code>5</code>	<code>[5, 3, 'p', 9, 'e']</code>
Slicing	<code>lista[1:3]</code>	<code>[3, 'p']</code>	<code>[5, 3, 'p', 9, 'e']</code>
	<code>lista[2:]</code>	<code>['p', 9, 'e']</code>	<code>[5, 3, 'p', 9, 'e']</code>
	<code>lista[:2]</code>	<code>[5, 3]</code>	<code>[5, 3, 'p', 9, 'e']</code>
Tamaño	<code>lista[2:-1]</code>	<code>['p', 9]</code>	<code>[5, 3, 'p', 9, 'e']</code>
Ordenar	<code>len(lista)</code>	<code>5</code>	<code>[5, 3, 'p', 9, 'e']</code>
Añadir elemento	<code>lista.sort()</code>		<code>[3, 5, 9, 'e', 'p']</code>
Retornar y remover	<code>lista.append(37)</code>		<code>[3, 5, 9, 'e', 'p', 37]</code>
	<code>lista.pop()</code>	<code>37</code>	<code>[3, 5, 9, 'e', 'p']</code>
Insertar	<code>lista.pop(1)</code>	<code>5</code>	<code>[3, 9, 'e', 'p']</code>
	<code>lista.insert(2, 'z')</code>		<code>[3, 'z', 9, 'e', 'p']</code>
Remover / eliminar	<code>lista.remove('e')</code>		<code>[3, 'z', 9, 'p']</code>
	<code>del lista[0]</code>		<code>['z', 9, 'p']</code>
Concatenación	<code>lista + [0]</code>	<code>['z', 9, 'p', 0]</code>	<code>['z', 9, 'p']</code>
Búsqueda	<code>9 in lista</code>	<code>True</code>	<code>['z', 9, 'p']</code>

