Python

Tipo	Hace	Fórmula	Ejemplo

Métodos Cadenas				Ejemplo para todo: **cadena = "Hola Mundo"**
Slice	Cortar		cad[:0] --> H	cad[-1:] --> o
Count	Número de veces	.count()	cadena.count("a") --> 1	
Find e Index	Retorna ubicación	.find() o .index()	cadena.find("Mundo") --> 5	cadena.index("Hola") --> 0
Rfind o Rindex	Ubicación de Izq.	.rfind() o .rindex()		

Carácter Booleano				
Startswith	Indica si empieza	startswith()	cadena.startswith("Hola")	True
Endswith	Indica si termina	endswith()	cadena.endswith("world")	False (Minúsculas)
Isdigit	Si solo tiene números	.isdigit()	cadena.isdigit()	False
Isnumeric	Nº y caract. Numéricos	.isnumeric()	cadena.isnumeric()	False
Isdecimal	Si es decimal	.isdecimal()	cadena.isdecimal()	False
Isalnum	Alfanúmericos	.isalnum()	cadena.isalnuml()	False
Isalpha	Alfabéticos	.isalpha()	cadena.isalpha()	True
Islower	Si son todas minúsculas	.islower()	cadena.islower()	False
Isupper	Si son todas mayúsculas	.isupper()	cadena.isupper()	False
Isprintable	Si son imprimibles	.isprintable()	cadena.isprintable()	True
Isspace	Si son solo espacios	.isspace()	cadena.isspace()	False

Métodos de Transformación Cadenas				
Capitalize	Convertir la 1ª mayúscula	.capitalize()	cadena.capitalize()	
Center	Alinean en el centro	.center()	cadena.center()	cadena.center(10, "*")
Ljust	Alinean en la izquierda	.ljust()	cadena.ljust()	
Rjust	Alinean en la derecha	.rjust()	cadena.rjust()	
Lower	Transf. en minúsculas	.lower()	cadena.lower() >>>> hola mundo	
Upper	Transf. en mayúsculas	.upper()	cadena.upper() >>>> HOLA MUNDO	
Swapcase	Invierten el tipo	.swapcase()	cadena.swapcase() >>>> hOLA mUNDO	
Strip	Elimina espacios en blanco	.strip()		
Rstrip	Elim. Espacios blanco Dcha.	.rstrip()		
Lstrip	Elim. Los de la Izq.	.lstrip()		
Replace	Remplaza	.replace()	cadena.replace("mundo", "world") >>>>> Hola world	

Métodos de Separación y Unión			
Split	Separa en saltos o espacios	.split()	cadena.split(" ") >>> Hola, Mundo
Join	Unir en cadena elem. Listas	.join()	" ".join(["Hola", "mundo"])] >>>> Hola mundo

Entrada por teclado			
Input	Entrada por teclado	iput()	Cadena inpu(" ") ; Entero int(input()) ; Decimal float(input())

Listas		lista = ["Andrés", "Miguel", 1, 3]
Unir dos listas	lista1 = [1, 2, 4] // lista2 = [3, 6, 7] // lista3 = lista1 + lista 2 >>>> [1, 2, 4, 3, 6, 7]	
Concatenar listas	vocales = ["E", "I", "O"] // vocales = vocales + ["U"] >>>> ["E", "I", "O", "U"]	
Elementos individuales	lista[0] >>>> Andrés	lista[-1] >>>> 3
Modificar elemento por posición	lista[2] = "Ricardo" (antes era Miguel) // lista[2] >>>>> Ricardo	
Manipular sublistas	lista[1:3] >>>> ["Andrés", "Miguel", 1]	# El número 3 no se cogería
	lista[2:3] >>>> 1	# Se cogería solo el valor 2
	lista[:3] >>>> ["Andrés", "Miguel", 1]	#Se cogería hasta valor 3 (no incluido)
	lista[2:] >>>> [1, 3]	# Se extrae una lsita desde valor 2 (incluido)
	lista[:] >>>> # Se extrae una lsita con todos los valores	
	lista[1:3] = ["Nop"] >>>> lista ["Andrés", "Nop"]	# Sustituye los elementos que elijamos
del	del lista[1] >>>> ["Andrés", 1, 3]	# Elimina la posición que digamos
	del lista >>>> lista []	# Elimina completamente la lista

Métodos de Listas			
Append	Añadir al final de la lista	.append()	lista.append(4) >>>>> ["Andrés", "Miguel", 1, 3, 4]
Count	Nº veces que aparece	.count()	lista.count(3) >>>>>> 1 #Solo aparece una vez
Extend	Se extiende la lista	.extend()	lista.extend([4]) >>>> ["Andrés", "Miguel", 1, 3, 4] // lista.extend(range(5,7) >>> [".....",5,6,7]
Index	Devuelve la 1ª aparición	.index()	lista.index(1) >>>> 2
Insert	Insertar elemento en lista	.insert()	lista.insert(1, 5.6) >>> ["Andrés", 5,6, "Miguel", 1, 3] #El primero indica la posición
Pop	Borra el último elemento	.pop()	lista.pop() >>> 3 #Lo muestra y lo borra
Remove	Borra la primera aparición	.remove()	lista.remove("Andrés") >>> ["Miguel", 1, 3]
Reverse	Invierte la lista	.reverse()	lista.reverse() >>>>>>> [3, 1, "Miguel", "Andrés"]
Sort	Ord. De menor a mayor	.sort()	# Se puede ordenar de mayor a menor -> lista.sort(reverse=True)
List	Convertir a lista	list	list("Hola") >>>>> ["H", "O", "L", "A"]
Len	Recorrer el nº elementos	len(tupla)	len(tupla) >>>> 3
	Copiar	[:]	# Para hacer una copia nueva -> lista2 = lista [:]

Diccionarios			diccionario = {"Andrés": 16, "Manuel": 88, "Ana": 31}
In	Para saber si es´ta	in	"Andrés" in diccionario >>>> True
Clear	Remueve todos los elem.	.clear()	diccionario.clear() >>>> {}
Copy	Una copia superficial	.copy()	lista2 = lista.copy()
Fromkeys	Crea un diccionario claves	.fromkeys()	# Tienes una lista -> dicci1 = dict.fromkeys(lista, 3) >>>> {"Andrés": 3, "Miguel": 3, 1:3, 3:3}
Get	Dev. El valor de coincide	.get()	dicc1 = dict(Plomo=3, Oro=7, Plata=4) // dicc1.get("Plata") >>>> 4
Items	Dev. Clave-Valor como Tupla	.items()	dicc1 = dict(Plomo=3, Oro=7, Plata=4) // dicc1.items() >>>> [("Plomo", 3), ("Oro", 7)….()]
Iteritems		.iteritems()	Devuelve un iterador sobre elementos (clave, valor)
Iterkeys		.iterkeys()	Devuelve un iterador sobre las claves
Itervalues		.itervalues()	Devuelve un iterador sobre los valores
Keys	Dev. Lista Claves de Diccio.	.keys()	diccionario.keys() >>>> ["Andrés", "Manuel", "Ana"]
Values	Dev. Lista Valor de Diccio.	.values()	diccionario.values() >>>> [16, 88, 31]
Pop	Borra el último elemento	.pop()	# Igual que en las Listas
Popitem	Borra (Clave, Valor)	.popitem()	diccionario.popitem() # Borra aleatorio

Tipo Range			
Range	Sucesión aritmética	range()	list(range(7)) >>>> [0, 1, 2, 3, 4, 5, 6, 7]
			list(range(5, 10)) >>>> [5,6,7,8,9,10] # La lista empieza en 5 y acaba en 10
			list(range(5, 21, 3)) >>>> [5, 8, 11, 14, 17, 21] # Lista de 5 a 21 con paso 3
			Concatenar -> list(range(3)) + list(range(2)) >>>> [0, 1, 2, 3, 0, 1, 2]
			list(range(len("mensaje secreto"))) >>>> [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

Bucle for			
For	Estr. Repite un bloque	for _ in _ ():	for i in [0, 1, 2]: // print("Hola ", end=" ") >>>> Hola Hola Hola
			for i in range(3): // print("Hola ", end=" " >>>> Hola Hola Hola

Bucle While			
While	Se repite mientras cumpla	while _ = _ :	i=1 // while i <= 3 // print(i) >>> 1, 2, 3