

ACLARACIÓN

IMPORTANTE: **x, y** representan cualquier tipo de valores de datos, **s** para una cadena de texto, **n** para un número, **L** para una lista donde **i, j** son índices de lista, **D** representa un diccionario y **k** es una clave de diccionario.

TIPOS DE DATOS

integer = 1 / -5 / 10 / -500 / 9500 / 0

boolean = True / False

float = 10.01 / -5.2 / 100.63

string = 'Juan' / 'Pedro' / 'Monte Hermoso' / 'Lupa'

list = [valor1, valor2, valor3,, valor n]

dictionary = {key1: value1, key2: value2, key3: value3,, keyn: value n}

OPERADORES NUMÉRICOS y de COMPARACIÓN

+	Suma	==	Igual a
-	Resta	!=	Distinto de
*	Multiplicación	>	Mayor que
**	Potencia	<	Menor que
%	Módulo	>=	Mayor o igual que
//	División Entera	<=	Menor o igual que

OPERADORES BOOLEANOS y ESPECIALES

and	Operador lógico Y	#	Comentario
or	Operador lógico O	\n	Representa un salto de línea
not	Operador lógico NO	\t	Representa un tabulador
		\\	Representa una barra invertida
		\'	Representa comillas simples o dobles e o una cadena de texto
		\"	

OPERACIONES CON STRING (CADENAS DE TEXTO)

string[i] Obtiene el carácter en la posición *i*.

string[-1] Obtiene el último carácter de la cadena.

string-[i:j] Obtiene los caracteres desde la posición *i* hasta la posición *j*.

MÉTODOS PARA STRING (CADENA DE TEXTO)

string.upper() Convierte toda la cadena a mayúsculas.

string.lower() Convierte toda la cadena a minúsculas.

string.count(x) Devuelve cuantas veces aparece *x* en la cadena.

string.find(x) Devuelve la posición de la primera aparición de *x* en la cadena.

string.replace(x, y) Reemplaza *x* por *y* en toda la cadena.

string.strip(x) Divide la cadena usando el valor *x*.

BUCLES (CICLOS)

while <condición>:

<código>

for <variable> in <lista>:

<código>

for <variable> in range(inicio, fin, paso):

<código>

for key, value in dict.items() :

<código>

DECLARACIONES DE CONTROL DE BUCLES (CICLOS)

break Finaliza la ejecución del bucle.

continue Salta a la próxima iteración.

pass Continúa sin ejecutar código.

FUNCIONES

def <función> (parámetros):

<código>

return <valor>



OPERACIONES CON LISTAS

<code>list = []</code>	Define una lista vacía
<code>list [i] = x</code>	Guarda el valor <i>x</i> en el índice <i>i</i>
<code>list [i]</code>	Devuelve el valor en el índice <i>i</i>
<code>list [-1]</code>	Devuelve el valor del último índice de la lista
<code>list [i:j]</code>	Devuelve los valores en el rango de índices de <i>i</i> hasta <i>j</i>
<code>del list [i]</code>	Elimina el valor con el índice <i>i</i> . Ordena nuevamente los índices

MÉTODOS DE LISTAS

<code>list.append(x)</code>	Añade el valor <i>x</i> al final de la lista.
<code>list.extend(L)</code>	Añade la lista <i>L</i> al final de la lista.
<code>list.insert(i, x)</code>	Añade el valor <i>x</i> en el índice <i>i</i> .
<code>list.remove(x)</code>	Elimina el primer elemento de la lista cuyo valor es <i>x</i> .
<code>list.pop(i)</code>	Elimina el elemento en la posición <i>i</i> y devuelve su valor.
<code>list.clear()</code>	Elimina todos los valores de la lista.
<code>list.index(x)</code>	Devuelve el índice del valor <i>x</i> en la lista.
<code>list.count(x)</code>	Devuelve la cantidad de veces que aparece <i>x</i> en la lista.
<code>list.sort()</code>	Ordena la lista en forma <i>ascendente</i> .
<code>list.reverse()</code>	Ordena la lista en forma <i>descendente</i> .
<code>list.copy()</code>	Devuelve una copia de la lista.

FUNCIONES INCORPORADAS

<code>print(x, sep='y')</code>	Imprime <i>x</i> objetos separados por <i>y</i> .
<code>input(s)</code>	Imprime <i>s</i> y espera a que se introduzca un valor que será devuelto.
<code>len(x)</code>	Devuelve la longitud de <i>x</i> (s, L o D).
<code>min(L)</code>	Devuelve el menor valor en <i>L</i> .

FUNCIONES INCORPORADAS (cont)

<code>max(L)</code>	Devuelve el mayor valor en <i>L</i> .
<code>sum(L)</code>	Devuelve la suma de todos los valores en <i>L</i> .
<code>range(n1, n2, n)</code>	Devuelve la secuencia de valores entre <i>n1</i> y <i>n2</i> con saltos de <i>n</i> .
<code>abs(n)</code>	Devuelve el valor absoluto de <i>n</i> .
<code>round(n1, n)</code>	Devuelve el valor <i>n1</i> redondeado a <i>n</i> dígitos.
<code>type(x)</code>	Devuelve el tipo de valor de <i>x</i> (string, float, int, list, etc.)
<code>str(x)</code>	Convierte <i>x</i> en una cadena de texto.
<code>int(x)</code>	Convierte <i>x</i> a entero.
<code>list(x)</code>	Convierte <i>x</i> en una lista.
<code>float(x)</code>	Convierte <i>x</i> a float (decimal).

CONDICIONALES

```

if <condición> :
    <có dig o>
else if <condición> :
    <có dig o>
.....
else:
    <có dig o>

Otra Opción:
if <valor> in <lista> :

```

VALIDACIÓN DE DATOS

```

try:
    <có dig o>
except <error> :
    <có dig o>
else:
    <có dig o>

```



By [gbertonemh](#)

Published 6th October, 2024.

Last updated 6th October, 2024.

Page 3 of 2.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>