Python Curso Básico

Una guia mas completa se puede consultar en
https://www.w3schools.com/python/python_operators.asp

El estudiante que ya domina Python puede pasar a resolver los ejercicios propuestos

Este cuaderno solo repasará los temas básicos en el siguiente orden:

- Tipo de datos
  - Numeros
  - Cadenas
  - Impresión
  - Listas
  - Diccionarios
  - Booleans
  - Tuplas
  - Conjuntos
- Operadores de Comparación
- Estamentos if, elif, else
- Bucle for
- Bucle while
- range()
- comprension de listas
- funciones
- expresiones lambda
- map y filter
- metodos

Tipo de Datos

Numeros

In [6]:
```
1 + 1
```

Out[6]: 2

In [7]:
```
1 * 3
```

Out[7]:
```
3
```

In [8]:
```
1 / 2
```

Out[8]:
```
0.5
```

In [9]:
```
2 ** 4
```

Out[9]:
```
16
```

In [12]:
```
(2 + 3) * (5 + 5)
```

Out[12]:
```
50
```

Asignación de Variables

In [44]:
```
# No puede comenzar con números o caracteres especiales.
num = 2
```

In [2]:
```
x = 2
y = 3
```

In [15]:
```
z = x + y
```

In [16]:
```
z
```

Out[16]:
```
5
```

Cadenas

In [3]:
```
'comillas simples'
```

Out[3]:
```
'comillas simples'
```

In [4]:
```
"comillas dobles"
```

Out[4]:
```
'comillas dobles'
```

In [5]:
```
" envolver 'muchas' otras citas"
```

Out[5]:
```
" envolver 'muchas' otras citas"
```

Impresión

In [6]:
```
x = 'hola'
```

In [7]:
```
x
```

Out[7]:
```
'hola'
```

In [8]:
```
print(x)
```

```
hola
```

In [49]:
```
num = 20
name = 'Pedro'
x = "hola"
```

In [48]:
```
print('Mi numero: {one}, y mi nombre es: {two} y tres es: {three}'.format(one=num,tw
```

```
Mi numero: 2, y mi nombre es: Pedro y tres es: hola
```

In [12]:
```
print('Mi numero: {}, y mi nombre es: {}'.format(num,name))
```

```
Mi numero: 20, y mi nombre es: Pedro
```

Listas

In [14]:
```
[1,2,3]
```

Out[14]:
```
[1, 2, 3]
```

In [15]:
```
['hola',1,[1,2]]
```

Out[15]:
```
['hola', 1, [1, 2]]
```

In [16]:
```
my_list = ['a','b','c']
```

In [17]:
```
my_list.append('d')
```

In [18]:
```
my_list
```

Out[18]:
```
['a', 'b', 'c', 'd']
```

In [19]:
```
my_list[0]
```

Out[19]:
```
'a'
```

In [20]:
```
my_list[1]
```

Out[20]:
```
'b'
```

In [21]:
```
my_list[1:]
```

```
['b', 'c', 'd']
```

Out[21]:

In [22]:
```
my_list[:1]
```

Out[22]:
```
['a']
```

In [23]:
```
my_list[0] = 'Nuevo'
```

In [24]:
```
my_list
```

Out[24]:
```
['Nuevo', 'b', 'c', 'd']
```

In [25]:
```
nest = [1,2,3,[4,5,['destino']]]
```

In [26]:
```
nest[3]
```

Out[26]:
```
[4, 5, ['destino']]
```

In [27]:
```
nest[3][2]
```

Out[27]:
```
['destino']
```

In [28]:
```
nest[3][2][0]
```

Out[28]:
```
'destino'
```

Diccionarios

In [55]:
```
d = {'key1':['item1','item2','item3'],'key2':'item2'}
```

In [56]:
```
d
```

Out[56]:
```
{'key1': ['item1', 'item2', 'item3'], 'key2': 'item2'}
```

In [58]:
```
d['key1']
```

Out[58]:
```
['item1', 'item2', 'item3']
```

In [62]:
```
d['key1']
```

Out[62]:
```
['item1', 'item2', 'item3']
```

Booleans

In [32]:
```
True
```

Out[32]:
```
True
```

In [33]:
```
False
```

Out[33]:
```
False
```

Tuplas

In [34]:
```
t = (1,2,3)
```

In [35]:
```
t[0]
```

Out[35]:
```
1
```

In [37]:
```
t[0] = 'Nuevo'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-37-c554ffec4506> in <module>
----> 1 t[0] = 'Nuevo'

TypeError: 'tuple' object does not support item assignment
```

In [39]:
```
"""
The key difference between the tuples and lists is that while the tuples are immutab
This means that tuples cannot be changed while the lists can be modified. Tuples are
"""

t = list(t)
```

In [40]:
```
t
```

Out[40]:
```
[1, 2, 3]
```

In [41]:
```
t[0] = 'Nuevo'
```

In [42]:
```
t = tuple(t)
```

In [43]:
```
t
```

Out[43]:
```
('Nuevo', 2, 3)
```

Conjuntos (Sets)

In [63]:
```
{1,2,3}
```

Out[63]:
```
{1, 2, 3}
```

In [64]:
```
{1,2,3,1,2,1,2,3,3,3,3,2,2,2,1,1,2}
```

Out[64]:
```
{1, 2, 3}
```

Comparison Operators

In [65]:
```
1 > 2
```

Out[65]:
```
False
```

In [66]:
```
1 < 2
```

Out[66]:
```
True
```

In [67]:
```
1 >= 1
```

Out[67]:
```
True
```

In [50]:
```
1 <= 4
```

Out[50]:
```
True
```

In [68]:
```
1 == 1
```

Out[68]:
```
True
```

In [69]:
```
'hola' == 'chau'
```

Out[69]:
```
False
```

Operadores Lógicos

In [53]:
```
(1 > 2) and (2 < 3)
```

Out[53]:
```
False
```

In [54]:
```
(1 > 2) or (2 < 3)
```

Out[54]:
```
True
```

In [55]:
```
(1 == 2) or (2 == 3) or (4 == 4)
```

Out[55]:
```
True
```

Estamentos if,elif, else

In [70]:
```
if 1 < 2:
    print('Siiii!')
```

```
Siiii!
```

In [75]:
```
if 1 < 3:
    print('Noo!')
```

```
Noo!
```

In [76]:
```
if 1 < 2:
    print('si')
else:
    print('no')
```

```
si
```

In [77]:
```
if 1 > 2:
    print('si')
else:
    print('no')
```

```
no
```

In [78]:
```
if 1 == 2:
    print('primera rpta')
elif 3 == 3:
    print('segunda rpta')
else:
    print('ultimas rpta')
```

```
segunda rpta
```

Bucle for

In [80]:
```
seq = [1,2,3,4,5]
```

In [81]:
```
for item in seq:
    print(item)
```

```
1
2
3
4
5
```

In [82]:
```
for item in seq:
    print('hola')
```

```
hola
hola
hola
hola
hola
```

In [84]:
```
for j in seq:
    print(j+j)
```

```
2
```

```
4
6
8
10
```

Bucle while

In [85]:
```
i = 1
while i < 5:
    print('i es: {}'.format(i))
    i = i+1
```

```
i es: 1
i es: 2
i es: 3
i es: 4
```

range()

In [88]:
```
range(5)
```

Out[88]:
```
range(0, 5)
```

In [67]:
```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

In [89]:
```
list(range(5))
```

Out[89]:
```
[0, 1, 2, 3, 4]
```

Comprensión de listas

In [91]:
```
x = [1,2,3,4]
```

In [92]:
```
out = []
for item in x:
    out.append(item**2)
print(out)
```

```
[1, 4, 9, 16]
```

In [93]:
```
[item**2 for item in x]
```

Out[93]:
```
[1, 4, 9, 16]
```

In [95]:
```
y = [item**2 for item in x]
```

In [96]:
```
y
```

Out[96]:
```
[1, 4, 9, 16]
```

Funciones

In [97]:
```
def my_func(param1='default'):
    """
    Documentacion va aqui
    """
    print(param1)
```

In [98]:
```
my_func
```

Out[98]:
```
<function __main__.my_func(param1='default')>
```

In [99]:
```
my_func()
```

```
default
```

In [100…
```
my_func('nuevo parametro')
```

```
nuevo parametro
```

In [101…
```
my_func(param1='nuevo parametro')
```

```
nuevo parametro
```

In [102…
```
def square(x):
    return x**2
```

In [103…
```
out = square(2)
```

In [104…
```
print(out)
```

```
4
```

Expresiones lambda

In [105…
```
def times2(var):
    return var*2
```

In [106…
```
times2(2)
```

Out[106…
```
4
```

In [109…
```
lambda var: var*2
```

```
<function __main__.<lambda>(var)>
```

Out[109…

map y filter

In [110…
```
seq = [1,2,3,4,5]
```

In [111…
```
map(times2,seq)
```

Out[111… <map at 0x14162624370>

In [85]:
```
list(map(times2,seq))
```

Out[85]: [2, 4, 6, 8, 10]

In [86]:
```
list(map(lambda var: var*2,seq))
```

Out[86]: [2, 4, 6, 8, 10]

In [87]:
```
filter(lambda item: item%2 == 0,seq)
```

Out[87]: <filter at 0x105316ac8>

In [88]:
```
list(filter(lambda item: item%2 == 0,seq))
```

Out[88]: [2, 4]

Metodos

In [118…
```
st = 'Hola mi nombre es Pat'
```

In [119…
```
st.lower()
```

Out[119… 'hola mi nombre es pat'

In [120…
```
st.upper()
```

Out[120… 'HOLA MI NOMBRE ES PAT'

In [121…
```
st.split()
```

Out[121… ['Hola', 'mi', 'nombre', 'es', 'Pat']

In [122…
```
tweet = 'Vamos Peru! #SiSePuede'
```

In [123…
```
tweet.split('#')
```

Out[123…
```
['Vamos Peru! ', 'SiSePuede']
```

In [124…
```
tweet.split('#')[1]
```

Out[124…
```
'SiSePuede'
```

In [125…
```
d
```

Out[125…
```
{'key1': ['item1', 'item2', 'item3'], 'key2': 'item2'}
```

In [126…
```
d.keys()
```

Out[126…
```
dict_keys(['key1', 'key2'])
```

In [127…
```
d.items()
```

Out[127…
```
dict_items([('key1', ['item1', 'item2', 'item3']), ('key2', 'item2')])
```

In [128…
```
lst = [1,2,3]
```

In [129…
```
"""
Remueve un item de una lista.  Si no se especifica, elimina el ultimo (index = -1)
"""
lst.pop()
```

Out[129…
```
3
```

In [131…
```
lst
```

Out[131…
```
[1, 2]
```

In [132…
```
'x' in [1,2,3]
```

Out[132…
```
False
```

In [133…
```
'x' in ['x','y','z']
```

Out[133…
```
True
```

Si llegaste hasta aqui, buen trabajo!