Introducción a Python

Héctor Enríquez

Outline

Números Enteros

```
a = 1
print(a)
a += 1
print(a)
a = a + 1
print(a)
```

```
1
2
3
```

Conversión a enteros `int`

```
print(int(1), int("1"), int(1.2))
```

```
1 1 1
```

Cadenas de caracteres

```
a = "texto de ejemplo"
print(a)
a += " final"
print(a)
a = "#" * 10
print(a)
```

```
texto de ejemplo
texto de ejemplo final
##########
```

Conversión a cadenas str

```
print((str("3"), str(3), str(3.2)))
```

```
('3', '3', '3.2')
```

Números Reales

```
a = 6.4
print(a)
a /= 2
print(a)
```

```
6.4
3.2
```

Conversión a reales `float`

```
print(float(3.2), float("3"), float(3))
```

```
3.2 3.0 3.0
```

Listas

```
a = [1, 2, 3]
print(a)
print(a + [8, 3])
print(a * 3)
```

```
[1, 2, 3]
[1, 2, 3, 8, 3]
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

Convertir a lista `list`

```
print(list("123"), list((1, 2, 3)), list({1: "a", 2: "b", 3: "c"}))
```

```
['1', '2', '3'] [1, 2, 3] [1, 2, 3]
```

Diccionarios

```
a = {"clave1": "valor1", "clave2": "valor2", 3: "otro", "ultimo": 4}
print(a)
```

Listas sin abreviar y abreviando

```
lista = []
for i in range(10):
    lista.append(i**2)
print(lista)
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
lista = [i**2 for i in range(10)]
print(lista)
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

Diccionarios sin abreviar y abreviando

```
squares = {}
for i in range(10):
    squares[i] = i**2
print(squares)
```

```
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}
```

```
squares = {i: i**2 for i in range(10)}
print(squares)
```

```
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}
```

Tuplas sin abreviar y abreviando

```
tupla = tuple(i**2 for i in range(10))
print(tupla)
```

```
(0, 1, 4, 9, 16, 25, 36, 49, 64, 81)
```

Generadores

```
generador = (i**2 for i in range(10))
print(generador)
```

```
<generator object <genexpr> at 0x6ffffcab620>
```

```
generador = (i**2 for i in range(10))
for i in generador:
    print(i, end=", ")
for i in generador:
    print(i)
```

```
0, 1, 4, 9, 16, 25, 36, 49, 64, 81,
```

Generadores II

```
from sys import getsizeof
generador = (i**2 for i in range(10))
lista = [i**2 for i in range(10)]
print(getsizeof(generador), getsizeof(lista))
```

```
88 192
```

```
from sys import getsizeof
generador = (i**2 for i in range(100))
lista = [i**2 for i in range(100)]
print(getsizeof(generador), getsizeof(lista))
```

```
88 912
```

Generadores III

```
from timeit import timeit
print(timeit("sum(i**2 for i in range(100))", number=1000))
print(timeit("sum([i**2 for i in range(100)])", number=1000))
```

```
0.030426119999901857
0.029341689001739724
```

def

```
def fun(a, b, c, d=1, e=""):
    print(a, b, c, d, e)
```

```
def fun(*args, **kwargs):
    print(args, kwargs)
    return args, kwargs
```

```
def fun(arg, *args, kwarg="default", **kwargs):
    print(arg, args, kwarg, kwargs)
```

def II

Peligroso

```
def fun(arg=[]):
    arg.append(len(arg))
    print(arg)

fun()
fun()
```

```
[0]
[0, 1]
```

lambda

```
fun = lambda x: x**2
print(fun(10))
```

```
100
```

map

```
lista = [i**2 for i in range(10)]
cadenas = map(str, lista)
print(cadenas)
print(list(cadenas))
```

```
<map object at 0x6ffffcb0d68>
['0', '1', '4', '9', '16', '25', '36', '49', '64', '81']
```

filter

```
lista = [i**2 for i in range(10)]
positivos = filter(lambda x: x > 0, lista)
print(positivos)
print(list(positivos))
```

```
<filter object at 0x6ffffcb0d30>
[1, 4, 9, 16, 25, 36, 49, 64, 81]
```

reduce

```
from functools import reduce
lista = [i**2 for i in range(10)]
resta = reduce((lambda x, y: x - y), lista)
print(resta)
```

```
-285
```

Comprobando el tipo del parámetro arg

```
def plus1(arg):
    if isinstance(arg, int):
        return arg + 1
    elif isinstance(arg, str):
        return int(arg) + 1
    elif isinstance(arg, list):
        return list(map(lambda x: x + 1, arg))

print(plus1(2))
print(plus1("2"))
print(plus1([2, 3]))
```

```
3
3
[3, 4]
```

Comprobando el tipo dentro del parámetro

```
def plus1(arg):
    if isinstance(arg, int):
        return arg + 1
    elif isinstance(arg, str):
        return int(arg) + 1
    elif isinstance(arg, list):
        return list(map(plus1, arg))

print(plus1(2))
print(plus1("2"))
print(plus1([2, 3]))
```

```
3
3
[3, 4]
```

Definiendo una clase

```
class Clase:
    def __init__(self):
        self.variable1 = "cadena"
        self.variable2 = 20

    def metodo1(self, a, b):
        self.variable1 = a
        self.variable2 = b
        return self.variable1 + self.variable2
```