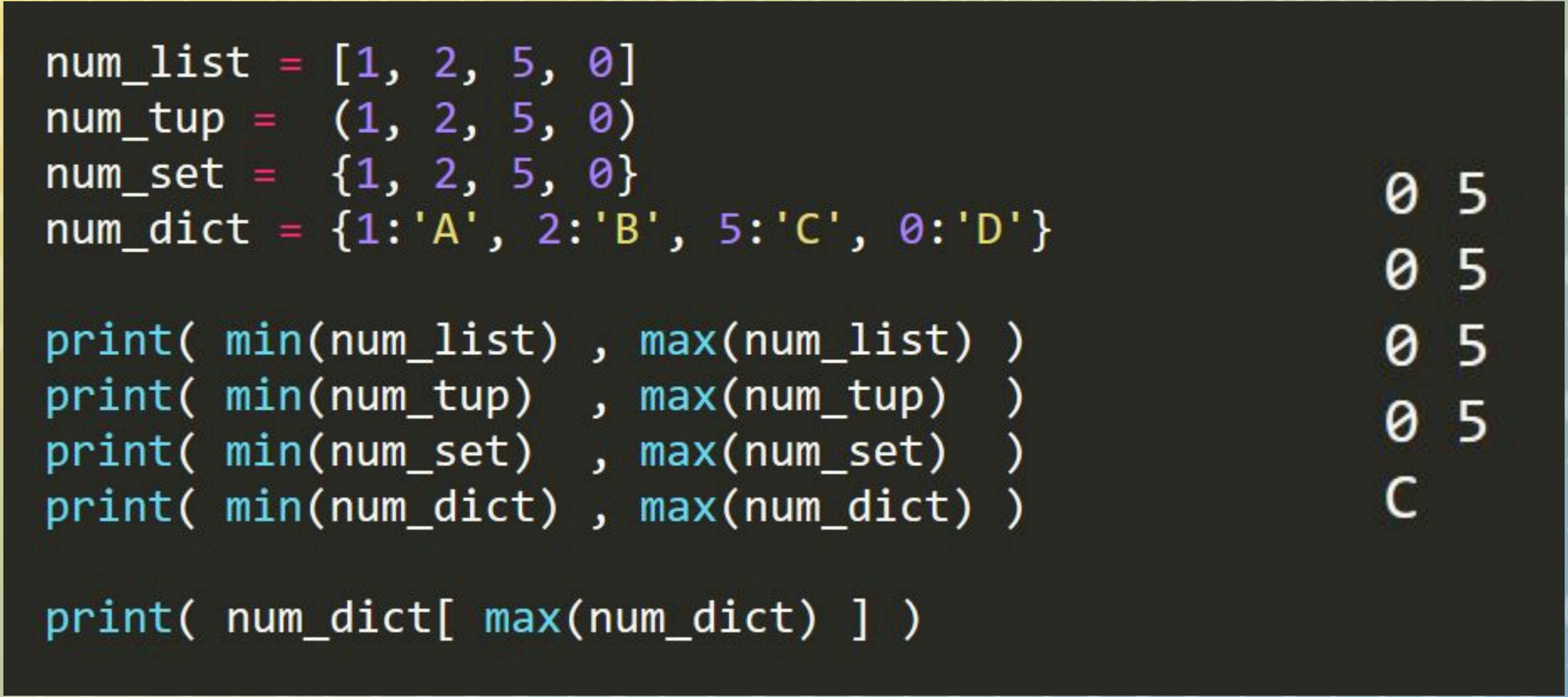
Clase 2

Algunos Tips extra para Python min - max



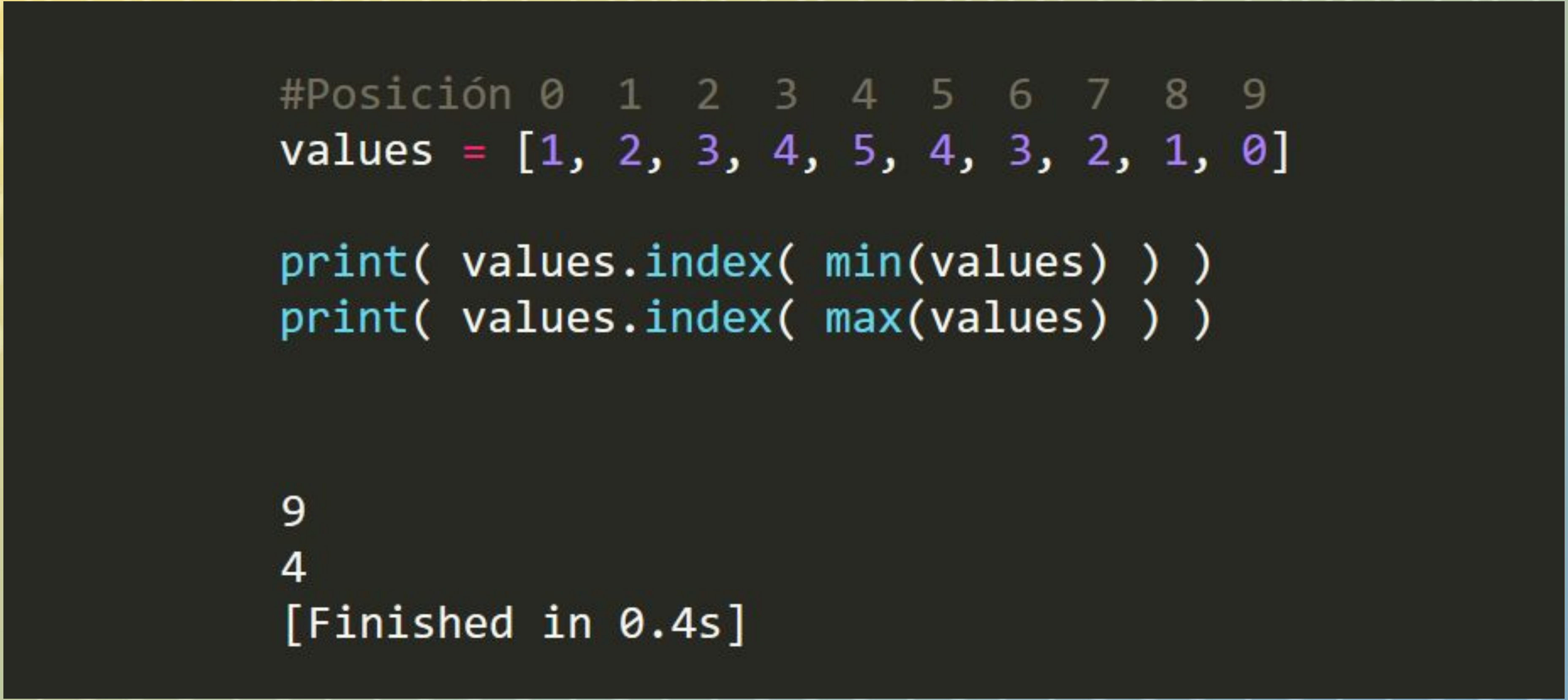
```python
num_list = [1, 2, 5, 0]
num_tup =  (1, 2, 5, 0)
num_set =  {1, 2, 5, 0}
num_dict = {1:'A', 2:'B', 5:'C', 0:'D'}

print( min(num_list) , max(num_list) )
print( min(num_tup)  , max(num_tup)  )
print( min(num_set)  , max(num_set)  )
print( min(num_dict) , max(num_dict) )

print( num_dict[ max(num_dict) ] )
```

```
0 5
0 5
0 5
0 5
C
```

min - max

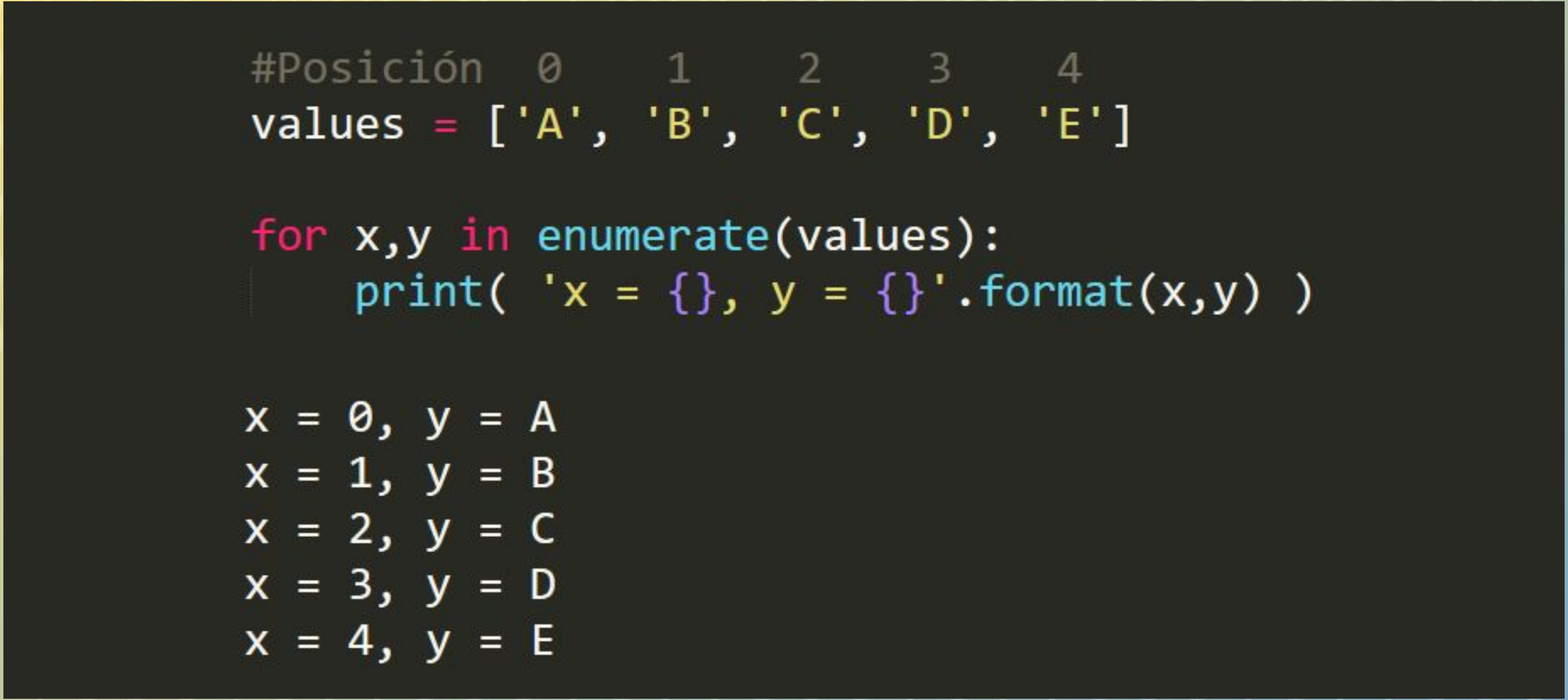


```
#Posición 0  1  2  3  4  5  6  7  8  9
values = [1, 2, 3, 4, 5, 4, 3, 2, 1, 0]

print( values.index( min(values) ) )
print( values.index( max(values) ) )
```

```
9
4
[Finished in 0.4s]
```

enumerate

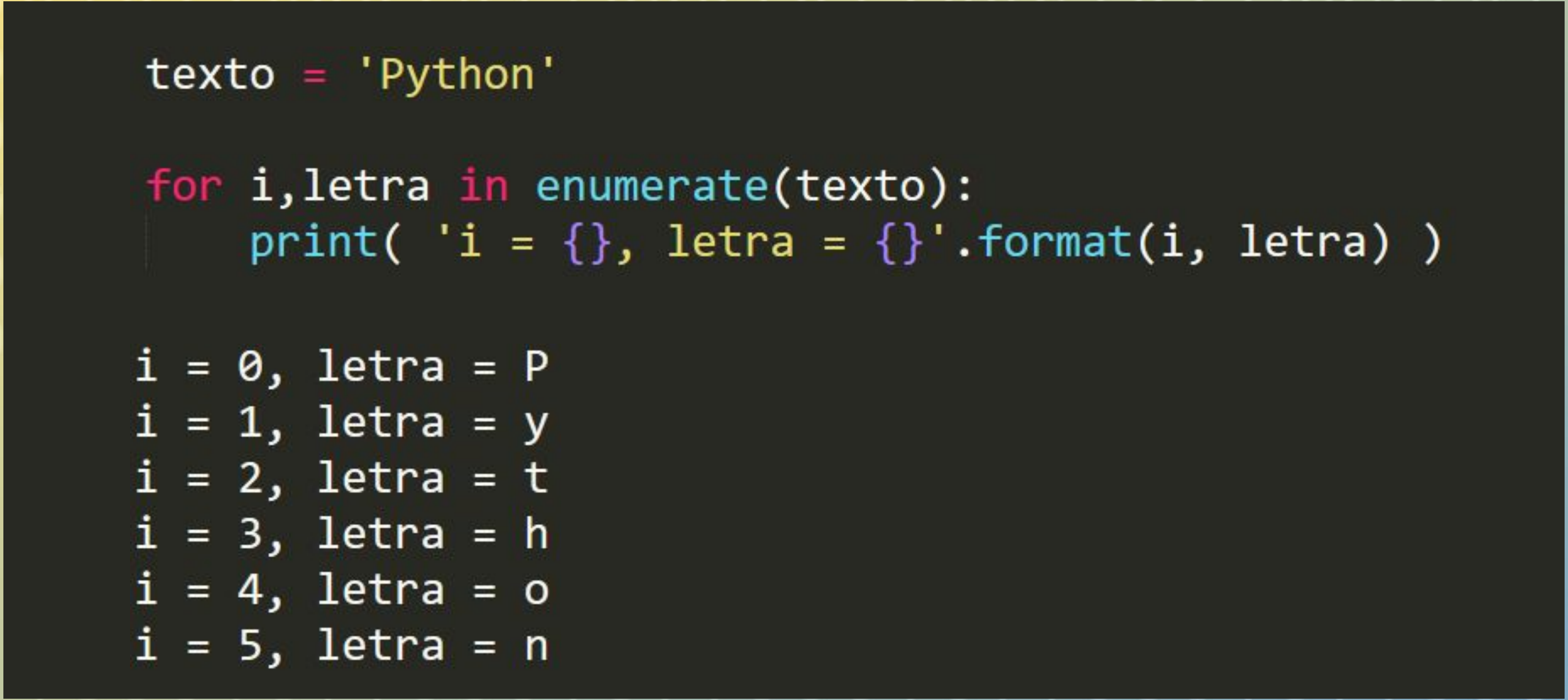


```
#Posición  0    1    2    3    4
values = ['A', 'B', 'C', 'D', 'E']

for x,y in enumerate(values):
    print( 'x = {}, y = {}'.format(x,y) )
```

```
x = 0, y = A
x = 1, y = B
x = 2, y = C
x = 3, y = D
x = 4, y = E
```

enumerate

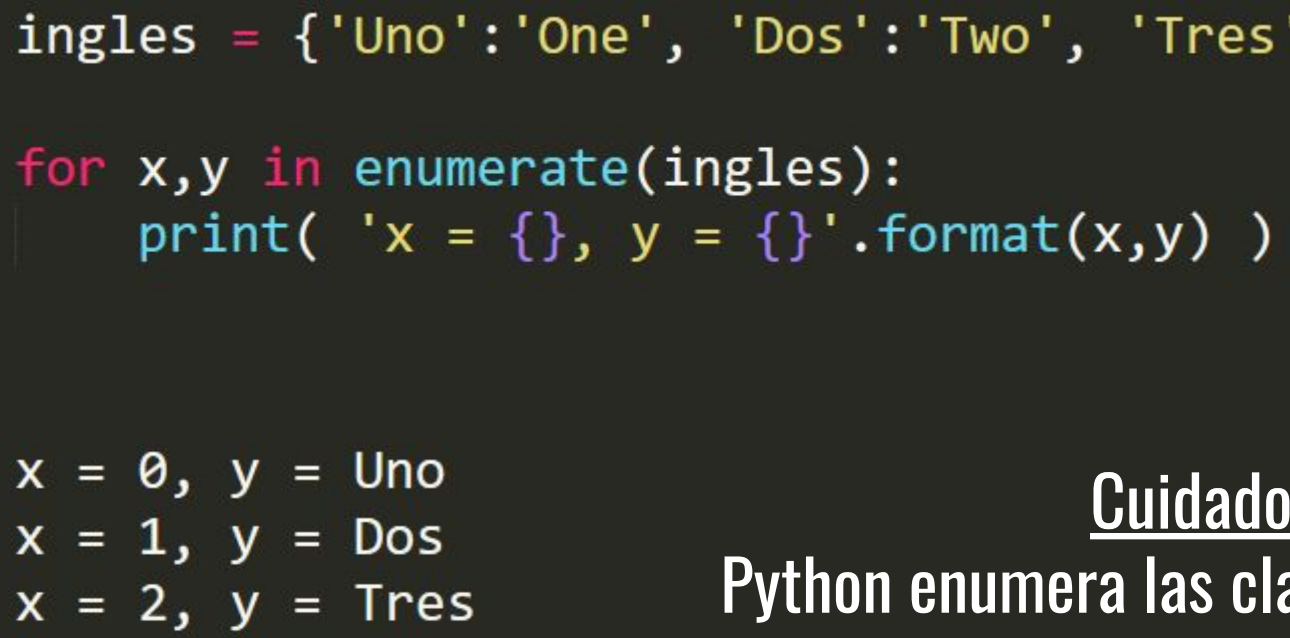


```
texto = 'Python'

for i,letra in enumerate(texto):
    print( 'i = {}, letra = {}'.format(i, letra) )
```

```
i = 0, letra = P
i = 1, letra = y
i = 2, letra = t
i = 3, letra = h
i = 4, letra = o
i = 5, letra = n
```

enumerate



```python
ingles = {'Uno':'One', 'Dos':'Two', 'Tres':'Three'}

for x,y in enumerate(ingles):
    print( 'x = {}, y = {}'.format(x,y) )
```

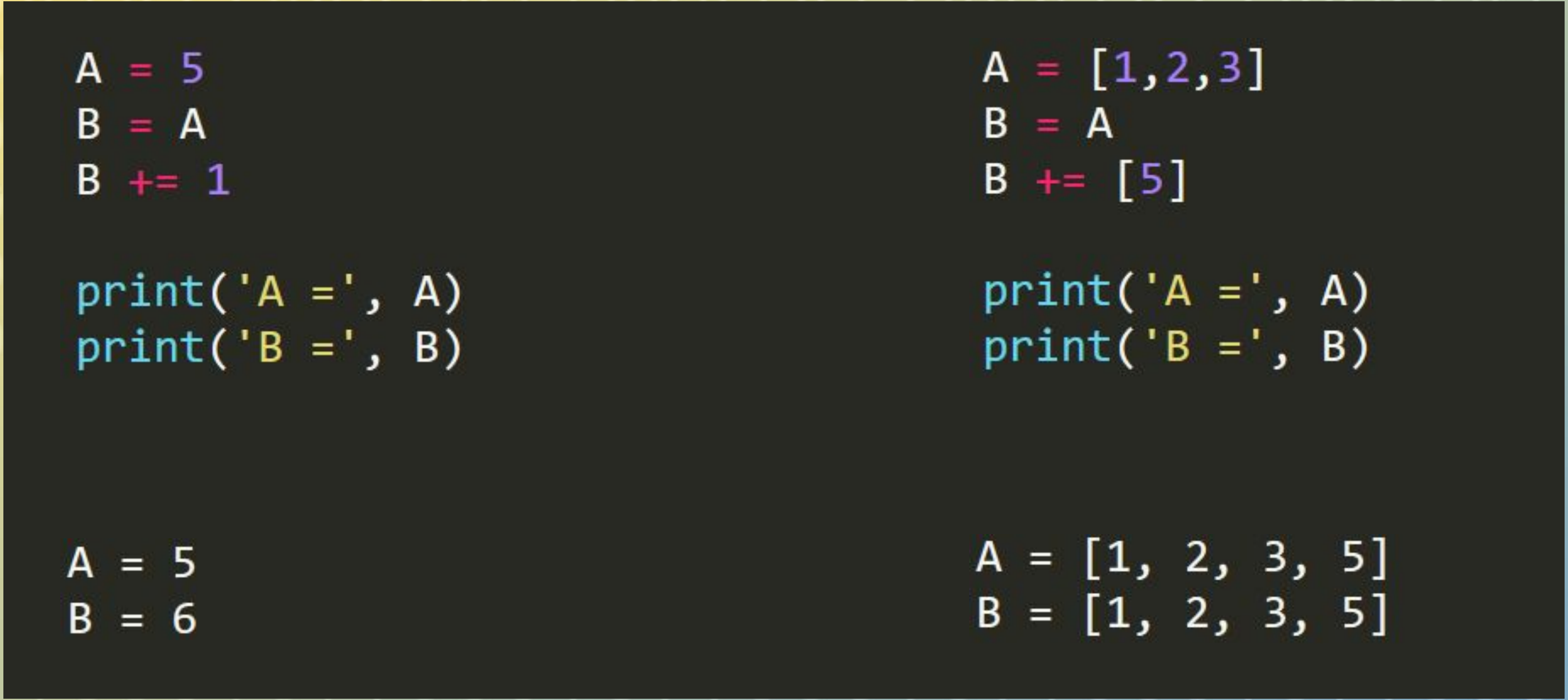
```
x = 0, y = Uno
x = 1, y = Dos
x = 2, y = Tres
```

<u>Cuidado!</u>

Python enumera las claves por default

Para obtener clave,valor usamos dict.items()

valor vs. referencia



```
A = 5
B = A
B += 1

print('A =', A)
print('B =', B)
```

```
A = 5
B = 6
```

```
A = [1,2,3]
B = A
B += [5]

print('A =', A)
print('B =', B)
```

```
A = [1, 2, 3, 5]
B = [1, 2, 3, 5]
```

Tipos inmutables: valor

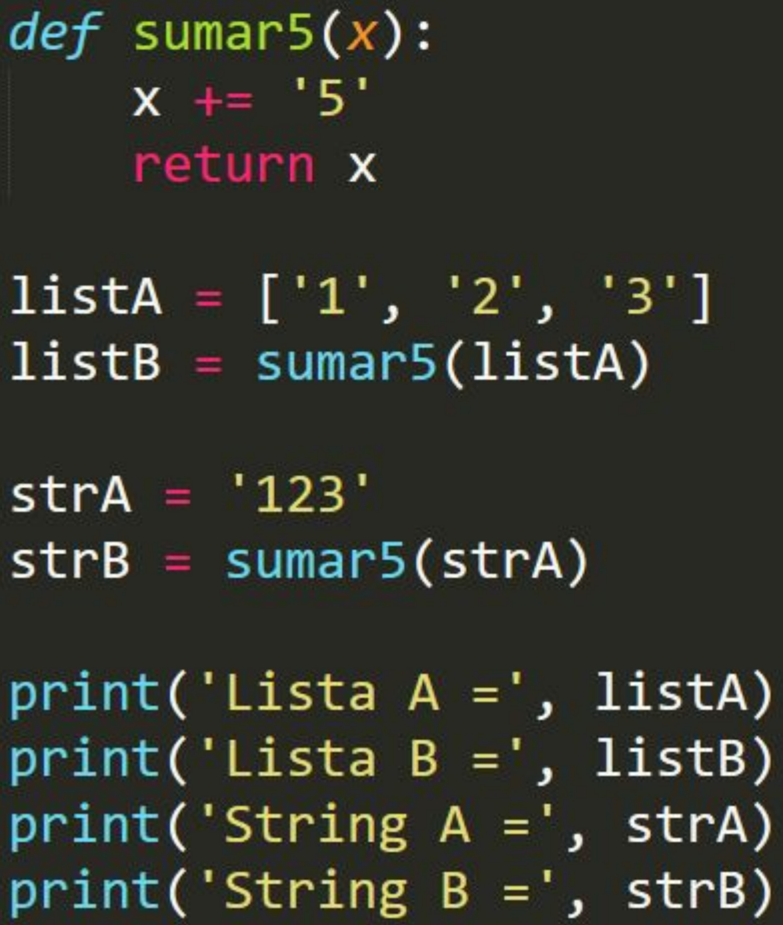
- int
- float
- string
- boolean
- tuple
- etc…

Tipos mutables: referencia

- list
- set
- dict
- etc…

Cuidado con los tipos mutables!

Las funciones crean un nuevo nombre de variable, pero referido a los mismos datos en memoria

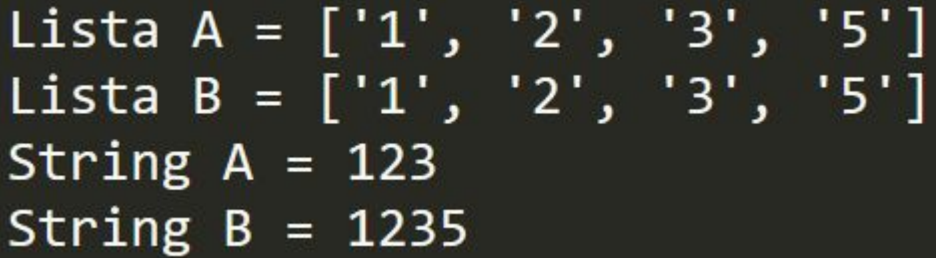


```python
def sumar5(x):
    x += '5'
    return x

listA = ['1', '2', '3']
listB = sumar5(listA)

strA = '123'
strB = sumar5(strA)

print('Lista A =', listA)
print('Lista B =', listB)
print('String A =', strA)
print('String B =', strB)
```

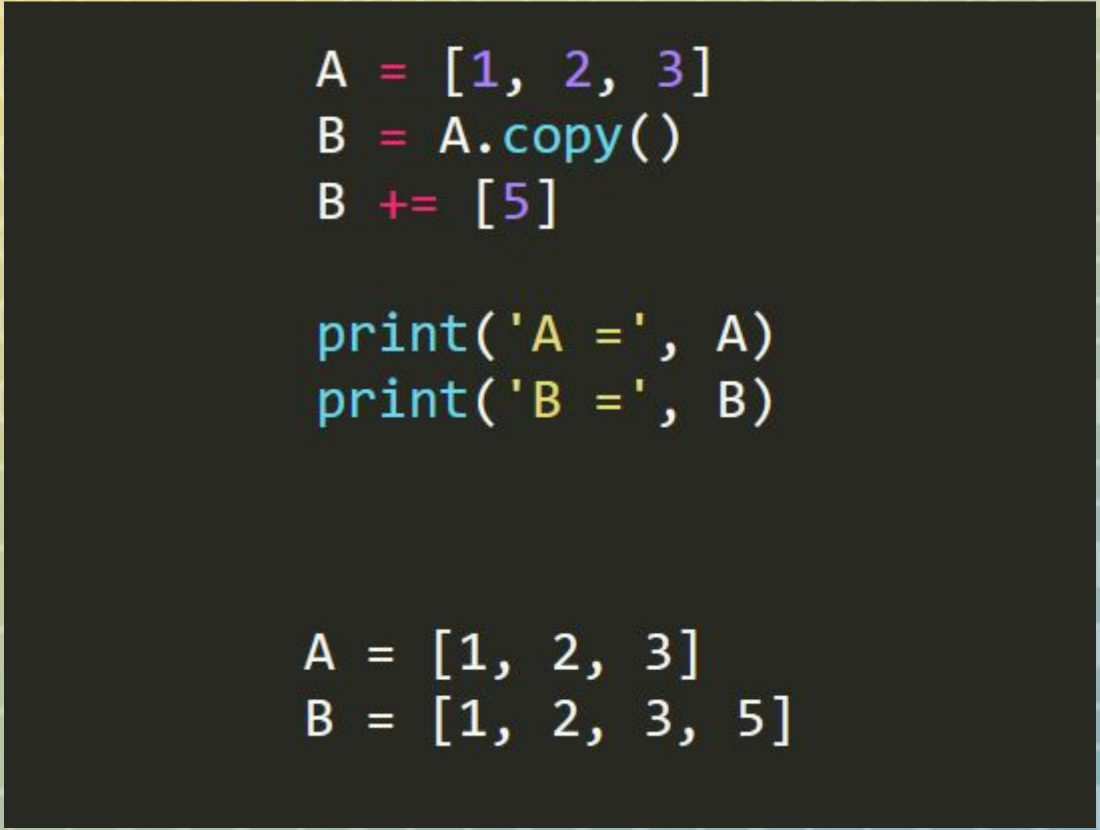


```
Lista A = ['1', '2', '3', '5']
Lista B = ['1', '2', '3', '5']
String A = 123
String B = 1235
```

¿Por qué?

Es más eficiente:
Velocidad y Memoria

.copy()

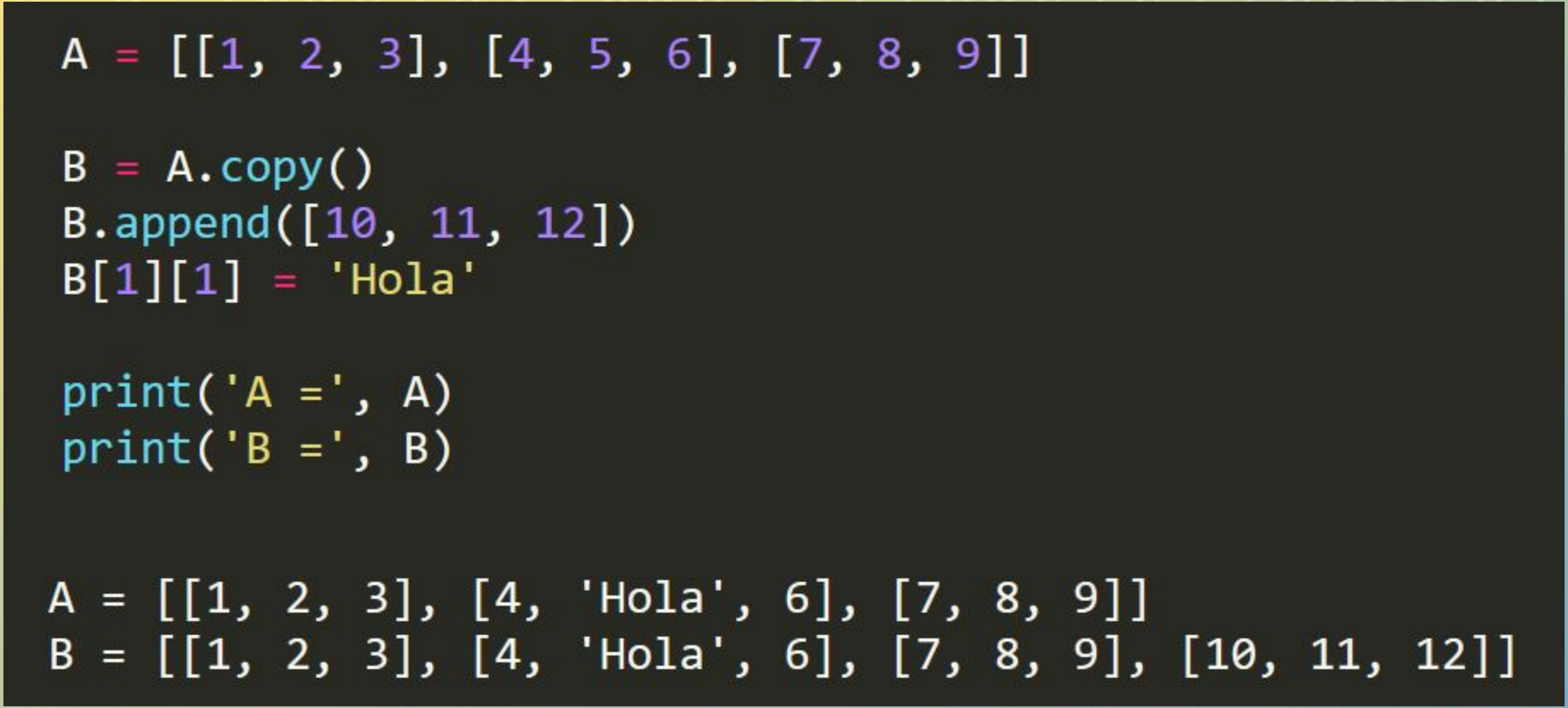


```python
A = [1, 2, 3]
B = A.copy()
B += [5]

print('A =', A)
print('B =', B)
```

```
A = [1, 2, 3]
B = [1, 2, 3, 5]
```

.copy() es *superficial*



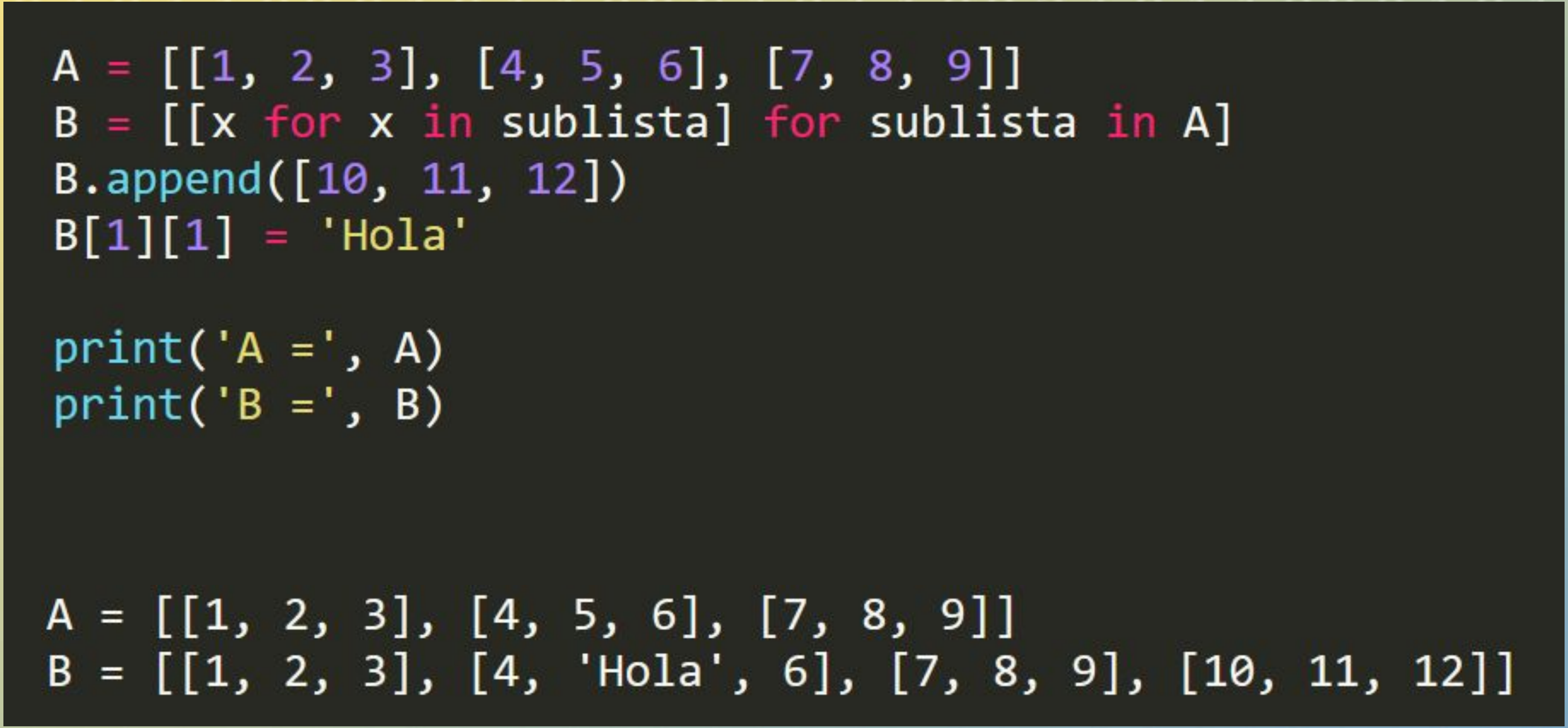
```
A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

B = A.copy()
B.append([10, 11, 12])
B[1][1] = 'Hola'

print('A =', A)
print('B =', B)
```

```
A = [[1, 2, 3], [4, 'Hola', 6], [7, 8, 9]]
B = [[1, 2, 3], [4, 'Hola', 6], [7, 8, 9], [10, 11, 12]]
```

Solución particular para lista de listas

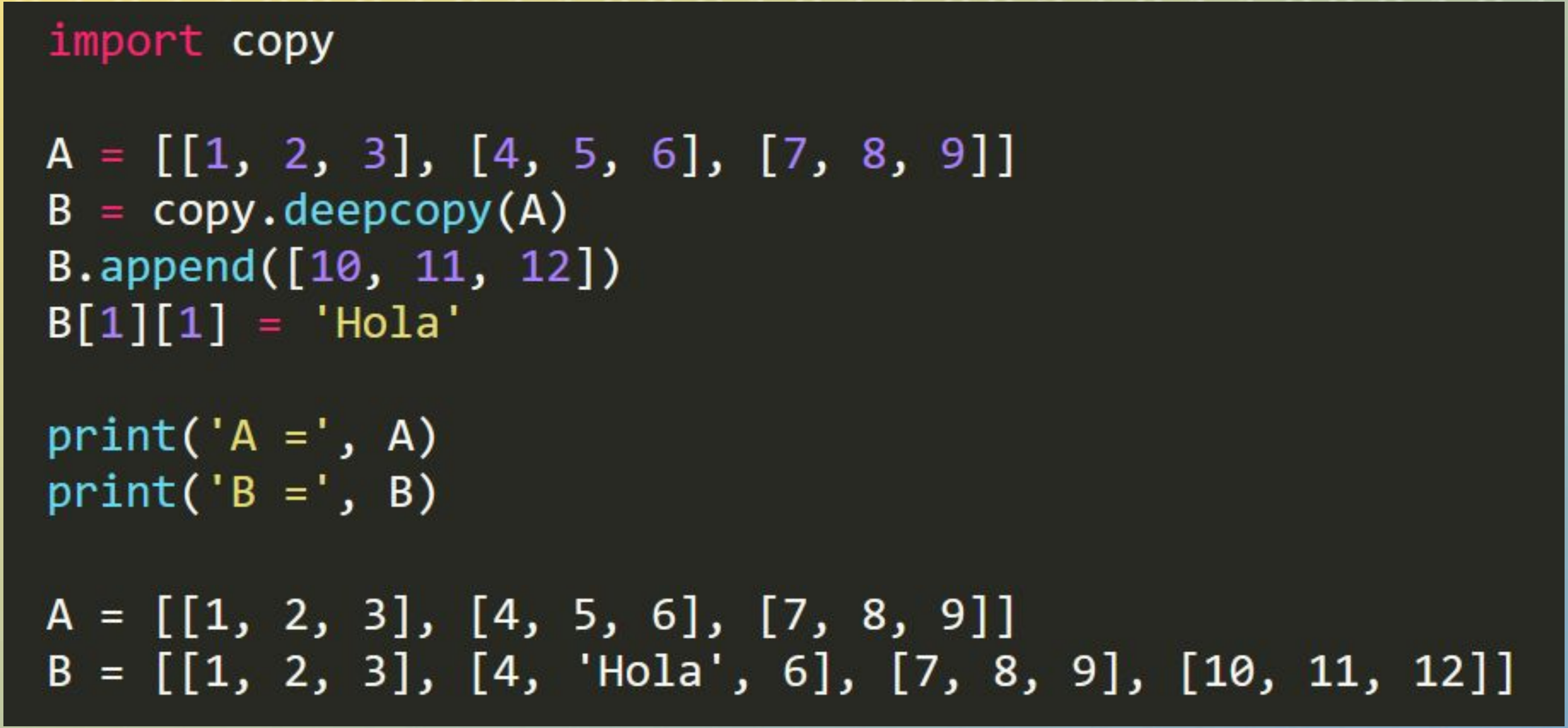


```
A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
B = [[x for x in sublista] for sublista in A]
B.append([10, 11, 12])
B[1][1] = 'Hola'

print('A =', A)
print('B =', B)
```

```
A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
B = [[1, 2, 3], [4, 'Hola', 6], [7, 8, 9], [10, 11, 12]]
```

Para hacer una copia *profunda* podemos usar librerías (Clase 3)



```python
import copy

A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
B = copy.deepcopy(A)
B.append([10, 11, 12])
B[1][1] = 'Hola'

print('A =', A)
print('B =', B)
```

```
A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
B = [[1, 2, 3], [4, 'Hola', 6], [7, 8, 9], [10, 11, 12]]
```