

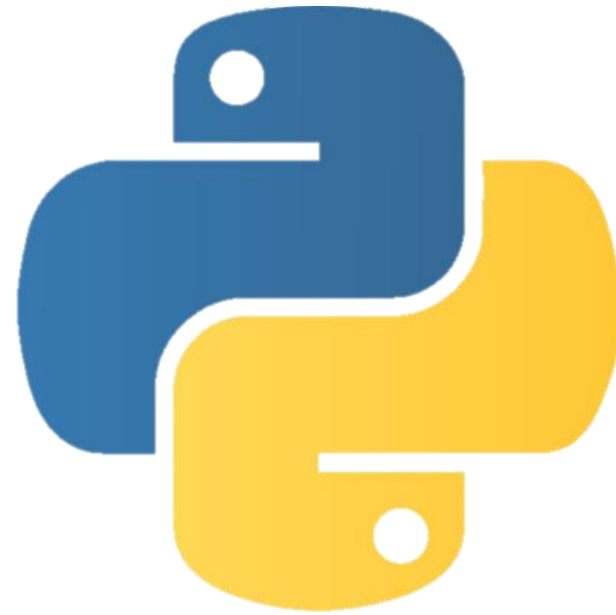
Curso Python 2024

Teoría-2: Slicing, Unpacking, NamedTuples



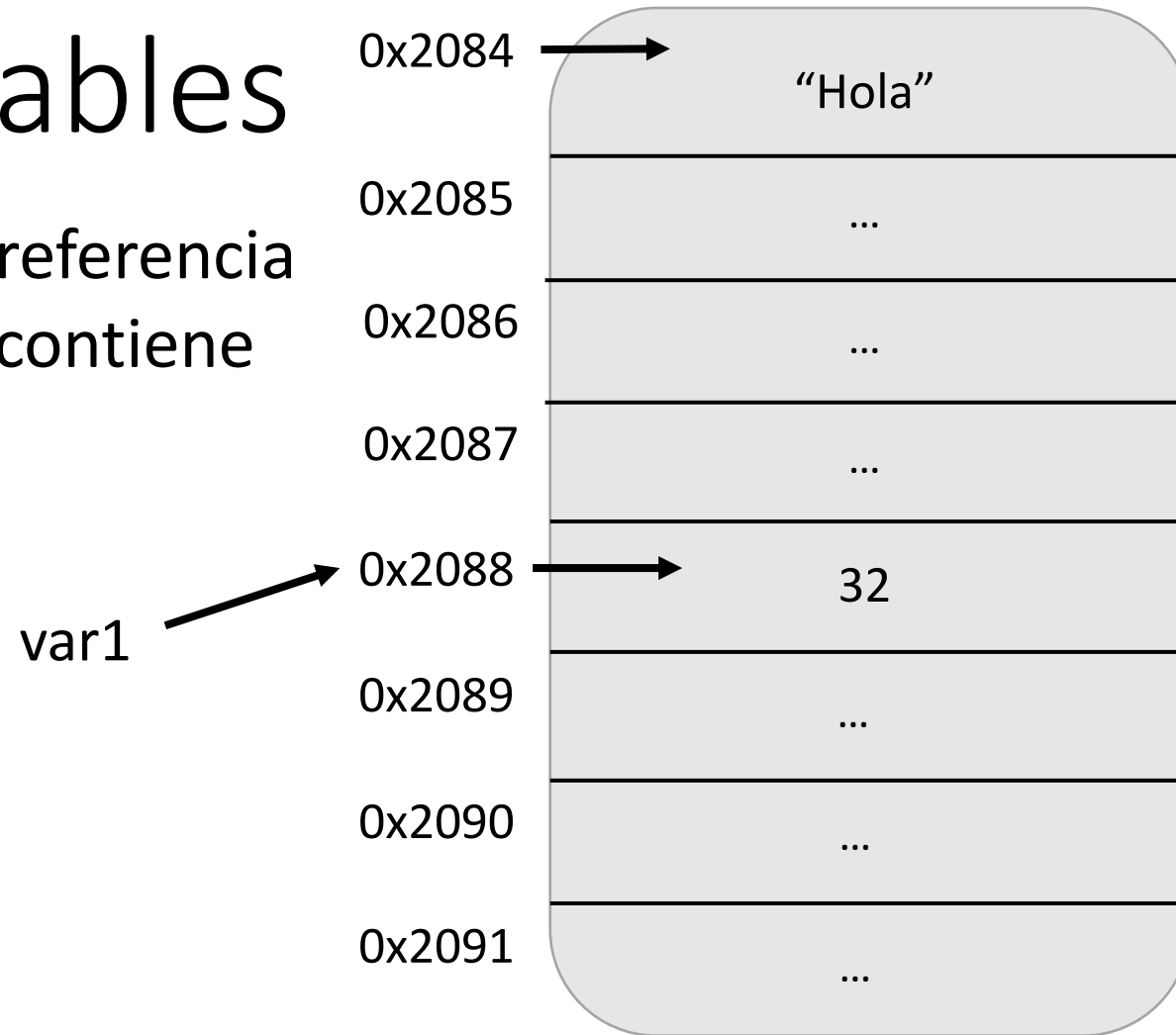
Hoy:

- ✓ Repaso clase anterior: Mutabilidad
- ✓ Slicing
- ✓ Unpacking
- ✓ NamedTuples* (Útil)

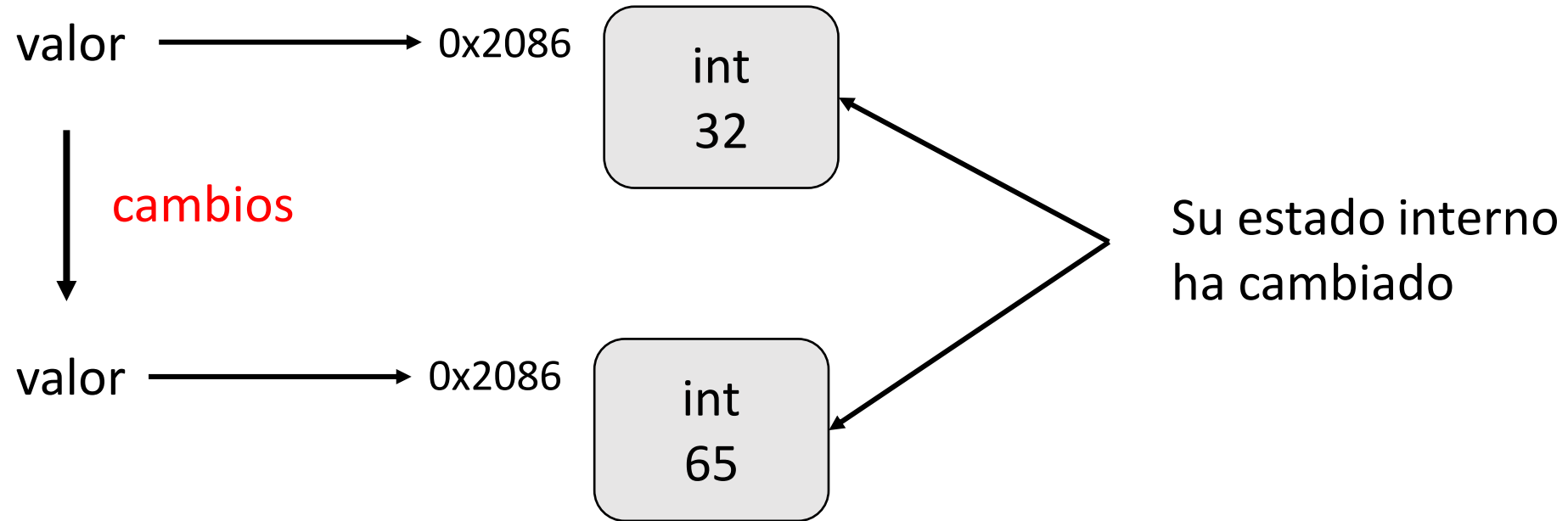


Repaso: Variables

- Una variable es una referencia a una dirección que contiene un objeto



Repaso: Mutabilidad



Secuencias: Concepto

- Conjunto de elementos ordenado
- Orden posicional
- Una lista es una secuencia
- Un set() no es una secuencia

$$l1 = [1, 2, 3, 4]$$

$$x_1, x_2, x_3, x_4$$

Secuencias: Heterogeneidad

Homogéneas

- Cada elemento es del mismo tipo
- Strings: “Python”

Heterogéneas

- Los elementos pueden ser de tipos distintos
- Tuplas: ('a', 200, [1, 2])



Secuencias: ¿Iterables?

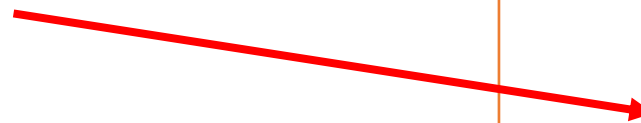
- Un **iterable** es un objeto del que podemos extraer elementos 1 por 1
- Cualquier tipo de secuencia es un iterable
- No todo iterable es una secuencia, por ejemplo $s = \{10, 20, 30\}$

```
l1 = [1, 2, 3, 4]
```

```
for elem in l1:  
    print(elem)
```

`l1[0]` ✓

`s[1]` ✗



Secuencias: Slicing

- Si la secuencia es mutable cuidado al modificar el estado interno
- Existen muchas operaciones con secuencias
- Vamos a recordar slicing

Operation	Description	Example
Indexing	Access individual elements in a sequence by their position.	<code>sequence[index]</code>
Slicing	Extract a portion of a sequence by specifying start, stop, and step values.	<code>sequence[start : stop : step]</code>
Concatenation	Combine two sequences together.	<code>sequence1 + sequence2</code>
Repetition	Create a sequence that repeats the original sequence a given number of times.	<code>sequence * n</code>
Length	Determine the number of elements in a sequence.	<code>len(sequence)</code>
Search	Find the first occurrence of an item in a sequence.	<code>sequence.index(item)</code>
Count	Count the number of times an item appears in a sequence.	<code>sequence.count(item)</code>

Secuencias: Slicing

seq = [1, 2, 3, 4, 5]
 0 1 2 3 4

- Desde **i** hasta **j** sin incluir **j** en pasos de tamaño **k**

seq[i : j : step] ← slice()

Secuencias: Slicing (step > 0)

seq = [1, 2, 3, 4, 5]
 0 1 2 3 4

$i, j > \text{len}(\text{seq}) \longrightarrow \text{len}(\text{seq})$

seq[i : j : step]

$i, j < 0 \longrightarrow \max(0, \text{len}(\text{seq}) + i/j)$

Ejemplos: seq[-4: 1] \longrightarrow [1]

seq[3: 20] \longrightarrow [4, 5]

Secuencias: Slicing (step > 0)

seq = [1, 2, 3, 4, 5]
 0 1 2 3 4

k = None → 1

i = None → 0

seq[i : j : step]

j = None → len(seq)

Ejemplos: seq[: 1] → [1]

seq[3: None] → [4, 5]

Secuencias: slice object

- Trabajar con números en ocasiones no es conveniente
- Disponemos de un objeto slice que recibe inicio, fin y paso
- `my_slice.indices(some_length)` devuelve el inicio, fin y paso según las transformaciones anteriores

`slice(start, stop, step)`

`my_slice = slice(1, 7, 2)`

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

`v[my_slice] → [2, 4, 6]`

Ejemplo: legibilidad

```
faces = face_detector.detectMultiScale(processed_frame, minNeighbors=5)
```

```
if np.any(faces):
```

```
# faces is a np.array of np.arrays with x, y, w, h of each face
```

```
x1, y1, w1, h1 = max(faces, key=lambda face: face[2] * face[3])
```

???

```
# Slices for readability
```

```
face_ypx, face_xpx = slice(y1, y1 + h1), slice(x1, x1 + w1)
```

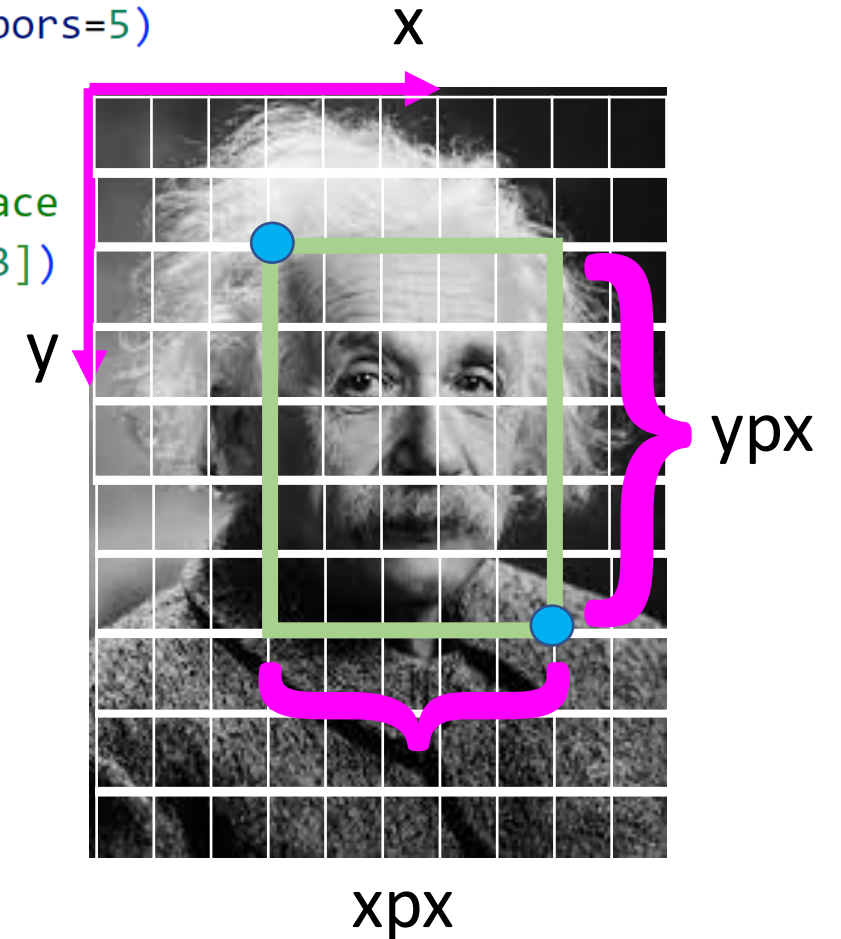
```
# Output frame
```

```
face_frame = frame[face_ypx, face_xpx]
```



```
# Output alternative
```

```
face_frame = frame[y1:y1+h1, x1:x1+w1]
```



Unpacking: Tuplas

- No necesitamos paréntesis para definir tuplas
- Excepto en tupla vacía ()

Recordatorio: Sintaxis tupla

`my_tuple = (1, 2, 3)`



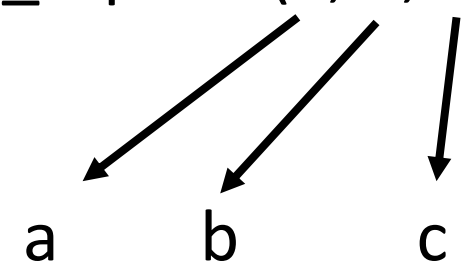
Es lo mismo

`my_tuple = 1, 2, 3`

Unpacking: Tuplas

- Unpacking es desempaquetar valores en variables individuales
- Se basa en posiciones
- Útil en secuencias

my_tuple = (1, 2, 3)



The diagram illustrates the unpacking of the tuple `my_tuple = (1, 2, 3)`. Three arrows originate from the elements of the tuple: the first arrow points from the value `1` to the variable `a`, the second arrow points from the value `2` to the variable `b`, and the third arrow points from the value `3` to the variable `c`.

Swap:

Ejemplo C

```
int main()
{
    int x, y;
    printf("Enter Value of x ");
    scanf("%d", &x);
    printf("\nEnter Value of y ");
    scanf("%d", &y);

    // Using a temporary variable to swap the values
    // Store the value of x in a temporary variable
    int temp = x;
    // Assign the value of y to x
    x = y;
    // Assign the value stored in the temporary variable to
    // y
    y = temp;

    printf("\nAfter Swapping: x = %d, y = %d", x, y);
    return 0;
}
```



Swap: Ejemplo Python

`a = 10`

`b = 5`

`a, b = b, a`

1. Se evalúa **completamente** el lado derecho (RHS)

2. Se realiza la asignación (LHS)

Estamos creando una tupla con copias de las referencias a la memoria en el paso 1

Swap: Ejemplo Python

`a = 10`

`b = 5`



`a, b = b, a`

1. Se evalúa **completamente** el lado derecho (RHS)

2. Se realiza la asignación (LHS)

Las variables de la izquierda apuntan a esas referencias en el paso 2

Unpacking: Warnings

- Cuidado con no secuencias
- Es posible desempaquetar sin orden (sets)
- No es muy eficiente en iterables grandes

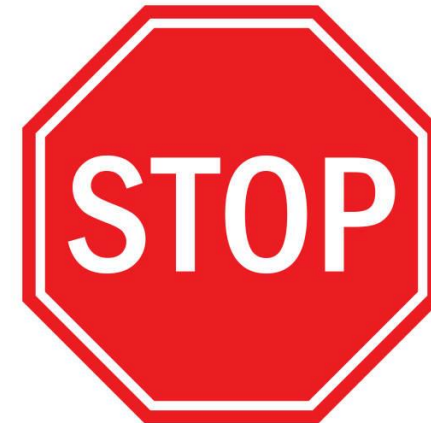


Unpacking: Operador *

- En ocasiones no queremos desempaquetar cada elemento
- El operador * nos permite desempaquetar el resto en una sola variable

```
len(my_iterable) = 1_000_000
```

```
a, b, c, d, ... = my_iterable
```



Unpacking: Operador *

- En ocasiones no queremos desempaquetar cada elemento
- El operador * nos permite desempaquetar el resto en una sola variable



$c = [1, 2, 3, 4]$
 $a, *b = c \longrightarrow a = 1$
 $b = [2, 3, 4]$

Operador *: Ejemplos

- También sirve para desempaquetar todo un iterable
- Útil en funciones
- (** para diccionarios)

```
def fun(a, b, c, d):  
    print(a, b, c, d)  
  
# Driver Code  
my_list = [1, 2, 3, 4]  
  
# Unpacking list into four arguments  
fun(*my_list)
```

Problema venv: Windows

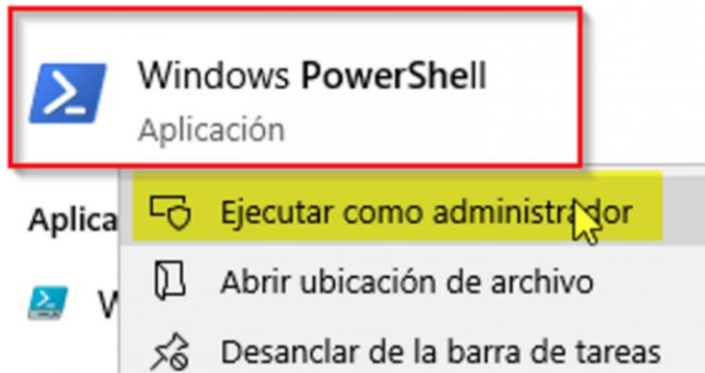
```
Windows PowerShell
PS C:\Users\Arturo\Documents> & '.\First Script.ps1'
& : No se puede cargar el archivo C:\Users\Arturo\Documents\First Script.ps1 porque la ejecución de scripts está
deshabilitada en este sistema. Para obtener más información, consulta el tema about_Execution_Policies en
https://go.microsoft.com/fwlink/?LinkID=135170.
En línea: 1 Carácter: 3
+ & '.\First Script.ps1'
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Users\Arturo\Documents>
```



Get-ExecutionPolicy -List

```
Windows PowerShell
PS C:\Users\Arturo\Documents> Get-ExecutionPolicy -List

Scope ExecutionPolicy
-----
MachinePolicy Undefined
UserPolicy Undefined
Process Undefined
CurrentUser Undefined
LocalMachine Undefined
```



Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

```
Windows PowerShell
PS C:\Users\Arturo\Documents> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```