



## Otras funciones para recorrer contenedores: ZIP

En Python, además de la función `range(inicio, final, paso)` que permite generar una secuencia de valores desde el *inicio* (incluido), hasta el *final* (sin incluir), con un incremento de *paso*.

*Existen otras dos funciones importantes:*

- **zip** (contenedor 1, contenedor 2, contenedor 3, ...):  
Devuelve, en cada iteración *una tupla* con un elemento de cada contenedor, mientras haya elementos de todos los contenedores. En pocas palabras: “*va recorriendo simultáneamente un elemento de cada contenedor*”. Se suele usar para analizar/comparar los valores de contenedores de forma “sincronizada”

*Ejemplo:*

```
for tupla in zip ([1,2,3], "Joselito", ['a','b','c','d']):  
    print (tupla)
```

→ { (1, 'J', 'a')  
     (2, 'o', 'b')  
     (3, 's', 'c')

**¡ Termina el `for` cuando se haya recorrido el contenedor con menos elementos!**



## Otras funciones para recorrer contenedores: Enumerate

- *enumerate* (contenedor):

Devuelve, en cada iteración una tupla con dos elementos (*añade un nuevo primer elemento*):

- El 1º un contador o posición, que empieza en 0 (lo añade automáticamente la función *enumerate*)
- El 2º es el elemento correspondiente del contenedor.

### *Ejemplos:*

```
for elemento in "ejemplo":  
    print (elemento)
```

e  
j  
e  
m  
p  
l  
o

```
for i, elemento in enumerate ("ejemplo"):  
    print (i, elemento)
```

0 e  
1 j  
2 e  
3 m  
4 p  
5 l  
6 o

```
for tupla in enumerate ("ejemplo"):  
    print (tupla)
```

(0, 'e')  
(1, 'j')  
(2, 'e')  
(3, 'm')  
(4, 'p')  
(5, 'l')  
(6, 'o')



## Otras funciones para recorrer contenedores(III)

*Ejemplos que combina enumerate y zip:*

```
for posición, tupla in enumerate (zip([1,2,3],"Joselito",['a','b','c','d'])):  
    print ("La posición de",tupla,"es la",posición)
```

La posición de (1, 'J', 'a') es 0

La posición de (2, 'o', 'b') es 1

La posición de (3, 's', 'c') es 2

```
for elemento in enumerate (zip([1,2,3],"Joselito",['a','b','c','d'])):  
    print ("La posición de", elemento[1],"es la", elemento[0])
```

La posición de (1, 'J', 'a') es 0

La posición de (2, 'o', 'b') es 1

La posición de (3, 's', 'c') es 2

*Observar que el for termina cuando se acaba uno de los contenedores*



## Ejercicio:

Modifique el proyecto *T09\_Datos\_Personales*

- En el módulo *datos\_personales.py*:
  - Añada una función: *persona\_más\_alta* que, recibiendo una lista de tupla de tipo *Persona4*, devuelva los apellidos y nombre de la persona de mayor estatura.
  - Añada una función: *relación\_alfabética\_de\_personas* que, recibiendo una lista de tupla de tipo *Persona4*, una lista de tuplas con los apellidos, nombre y edad de las personas.
  - Añada una función: *n\_personas\_mayor\_edad* que, recibiendo una lista de tupla de tipo *Persona4* y un número entero *n*, devuelva una lista de tuplas con la edad, el dni, el peso y la estatura de las *n* personas menos jóvenes.
- En el módulo *test\_datos\_personales.py*
  - Añada un test para cada una de las funciones anteriores