

ACTIVIDADES TEMA3

```
>>>['a','b','c'].index("c")
```

La sentencia index muestra la posición que ocupa índice que introducimos entre paréntesis. En este caso la salida es generada es 2, posición que ocupa c.

```
>>>[3,2,1].pop(2)
```

Pop elimina y devuelve todos los elementos de la lista hasta la posición que le indiquemos. Por eso lo que nos devuelve es un 1.

```
>>>dic1={'prof1':'g_ingles','prof2':'g_frances'}
```

Se trata de un diccionario o matriz asociada, este está formado por pares, el primero es la clave y el segundo es el valor de la clave.

Para hacer una consulta en este diccionario podemos introducir:

```
dic1['prof1']
```

y nos devolverá: g_ingles

```
>>>dic1.update({'prof3':'g_aleman'})
```

Esta sentencia se utiliza para añadir una entrada nueva al diccionario anteriormente generado. Al añadir una entrada, se realiza, esta se introduce en la posición 0 del diccionario.

```
>>>print(dic1.keys())
```

Print nos permite visualizar las claves del diccionario que hemos creado.

```
>>>{1,2,3}&{2,4,6}
```

La sentencia representa dos conjuntos y con el operador & indicamos que nos muestre la intersección de los conjuntos. Por lo tanto muestra 2.

```
>>>{1,2,3}|{2,4,6}
```

| muestra la unión de los dos conjuntos, es decir la salida generada es 1,2,3,4,6, al tratarse de una unión los valores no se repiten.

```
>>> {1,2,3}&{1}
```

La salida que nos devuelve es 1, es decir la intersección de los dos conjuntos. En este caso 1 se trata de un subconjunto de (1,2,3).

```
>>> {1,2,3}-{2,4,6}
```

Aquí tenemos dos conjuntos y con la sentencia - se nos devuelve la exclusión del segundo conjunto respecto al primero, por lo tanto la salida generada es: {1, 3}

```
>>> {1,2,3}^ {2,4,6}
```

Con la sentencia ^ se nos devuelve los términos que están contenidos en el primer conjunto o los que están contenidos en el segundo, pero no los que hay en común. Salida generada: {1,2,3}^ {2,4,6}

```
>>> 3 in {1,2,3}^ {3,4,5}
```

La salida de esta sentencia genera un booleano, ya que lo que estamos evaluando es si 3 esta contenido en el primer conjunto o en el segundo, y como está en los dos la salida es un FALSE.

```
>>> 3 not in (1,2,3)
```

En esta sentencia analizamos si 3 no esta contenido en el conjunto, por ese motivo Python nos genera un FALSE, ya que sí está contenido.