

## If elif else

1. Genera los posibles formatos para:
  - a) (A1lf-else1a.py) Si semáforo está en verde, cruzar la calle. Sino, esperar.
  - b) (A1lf-else1b.py) Si gasto hasta 100€, pago con dinero en efectivo. Sino, si gasto más de 100€ pero menos de 300€, pago con tarjeta de débito. Si no, pago con tarjeta de crédito.
  - c) (A1lf-else.py). Si la compra es mayor a 100€, obtengo un descuento del 10%.
2. (A1lf-else2.py). Genera la función **esUnNumero()**, a la que le llega un número (obtenido por teclado) y devuelve un mensaje con el tipo de número que es (entero: positivo, negativo, binario, octal, hex, otros: flotantes, complejo) al que pertenece.
3. (A1lf-else3.py). Genera varios ejemplos de if-elif-else con condiciones compuestas:
4. (A1lf-else4.py). Genera una función **opcionesMenu()** que le envíen la opción tecleada de un menú. Las posibles opciones del menú son 9 (1.9). Por ahora solo visualizará la opción enviada y un mensaje si la opción es incorrecta.
5. (A1lf-else5.py). A partir del ejemplo 6) Alternativa anidadas:

```
num1,num2,num3 = 4,3,5
if (num1 > num2):
    if (num1 > num3):
        print (" Es mayor el " , num1 )
        if (num2 > num3):
            print (" Es menor el " , num3 )
        else :
            print (" Es menor el " , num2 )
    else:
        if (num2 > num3):
            print ('El mayor el ' , num1)
            print ('El menor el ' , num3)
        else:
            print ('El mayor el ' , num3)
            print ('El menor el ' , num2)
else:
    if num3 > num1:
        print(" Es menor el " , num1 )
        if (num2 > num3):
            print ('El mayor el ' , num2)
        else:
            print ('El mayor el ' , num3)
    else:
        print(" Es mayor el " , num2 )
        print(" Es menor el " , num3 )
```

- Modificar para que se contemple que los números tengan igual valor.

## While

1. (A2While1.py). Dado el siguiente código:

```
while True:
    si_teclean = input("Teclee algun valor distinto a enter: ")
    print ("valor booleano ", bool(si_teclean))
    if si_teclean:
        pass #aqui hay que incluir las acciones cuando teclean
algo
```

```

        break
    else:
        print ("Por favor tecleen")

```

- Modificar para no utilizar las palabras claves **True** y **break**,

## For

1. (A3for1a.py). Dado el siguiente código:

```

lista = ['gato', 'perro', 'canario', 'peces']

```

- Genera la función **analizoLista()**.

Esta función recibe una **lista** (por ejemplo, la anterior) y a continuación nos visualizara:

- El número de elementos de la lista recibida.
- El valor de cada elemento
- Además, si el elemento es de tipo string, indicara igualmente el número de caracteres del mismo.

2. (A3for2.py). Dado el siguiente código:

```

print ("Listado de años del 2000 hasta ahora")
num1 = 2000 # valor inicial de entrada
num2 = 2019
while num1 <= num2:
    print ("Año " + str(num1))
    num1 += 1 #Modificando condicion

```

- Modificar para realizarlo con la sentencia **for**.

3. (A3for3.py). Dado el siguiente código:

```

lista = ['Mary', 'tenia', 'un', 'corderito']

```

- Visualizar, utilizando **range()**:

```

Visualizar los elementos de la lista anteponiendo su posicion
0 Mary
1 tenia
2 un
3 corderito

```

4. Utilizando **range()**:

- (A3for4a.py). Visualizar los “20” primeros números pares.
- (A3for4b.py). Visualizar los “N( valor tomado por teclado)” primeros números impares.
- (A3for4c.py). Realizar una función que recibiendo los parámetros necesarios realice lo anterior.

5. (A3for5.py). Utilizando **range()**:

- Crear la función **multiplosDe()**. Esta función recibirá la información necesaria para:
  - Visualizar los múltiplos de un número, en un intervalo dado.

6. (A3for6.py) Crear una lista con el intervalo de números (-20 a 20) pero solo los pares.

7. (A3for7.py). Modifica el ejercicio (Afor3.py). sin utilizar **range()**.

8. (A3for8.py). A partir de los ejemplos de formatos de **for**.

- Modificar para que al finalizar se visualice el número de elementos de la secuencia generada por las funciones.

9. (A3for9.py). A partir de los ejemplos de formatos de **for**.

- Crear el script que visualice:

```

Recorrido por zip(diccionario, tupla)
diccionario: {'Juan': 111, 'Antonio': 222, 'Pedro': 333}
tupla       : ('Luis', 5, 'sol', 'luna')
De la tupla es: Luis   y del Diccionario es: Juan con valor: 111
De la tupla es: 5     y del Diccionario es: Antonio con valor: 222
De la tupla es: sol   y del Diccionario es: Pedro con valor: 333

```

---

**10.** Tus propios enunciados y soluciones.

- El fichero aparte.