Bucles. Ejercicios (cont.)

Divisores de un número

Diseña una función que calcule la lista de los divisores, y si es posible hazlo de forma que recorra los candidatos sin rebasar la raíz del dato.

In [1]:
```
# Tres soluciones:
import math

def imprimir_divisores(n):
    for d in range(1, n+1):
        if n % d == 0:
            print(d, " ", end="")
    print()

def divisores(n):
    return [d for d in range(1, n+1) if n % d == 0]

def divisores_2(n):
    divs = []
    for d in range(1, int(math.sqrt(n))+1):
        if n % d == 0:
            divs.append(d)
            if d*d != n:
                divs.append(n//d)
    return divs

imprimir_divisores(36)

print(divisores(36))

print(divisores_2(36))
```

```
1  2  3  4  6  9  12  18  36
[1, 2, 3, 4, 6, 9, 12, 18, 36]
[1, 36, 2, 18, 3, 12, 4, 9, 6]
```

Mínimo y máximo

Diseña una función que toma una lista de enteros como parámetro y devuelve el mínimo y el máximo valor de dicha lista..

In [2]:
```
def minimo_maximo(lista):
    a, b = lista[0], lista[0]
    for i in lista[1:]:
        if i < a:
            a = i
        if i > b:
            b = i
    return a, b

# Pensándolo mejor, existen funciones ya predefinidas para esto:

def min_max(lista):
    return min(lista), max(lista)

print(minimo_maximo([1, 4, 2, 876, 12345, 647, 123, 531245, 3145]))
print(min_max([1, 4, 2, 876, 12345, 647, 123, 531245, 3145]))
```

```
(1, 531245)
(1, 531245)
```

Minúsculas

Diseña una función que filtra las palabras de una lista, quedándose con las que están escritas en minúscula.

In [3]:
```
def minusculas(lista):
    return [p for p in lista if p.lower() == p]

minusculas(["peluche", "Peluchito", "fer", "Fer"])
```

Out[3]:
```
['peluche', 'fer']
```

Simplificar una fracción

Una fracción se puede simplificar calculando el máximo común divisor de sus términos… Diseña una función que da la fracción simplificada y otro que simplifica “in place” una fracción dada. Este asunto puede ilustrar el concepto de objetos mutables e inmutables.

In [4]:
```
from math import gcd as mcm

def simplificar(fraccion):
    m = mcm(fraccion[0], fraccion[1])
    fraccion[0] = fraccion[0] // m
    fraccion[1] = fraccion[1] // m

f = [36, 24]
simplificar(f)
print(f)
```

```
[3, 2]
```