

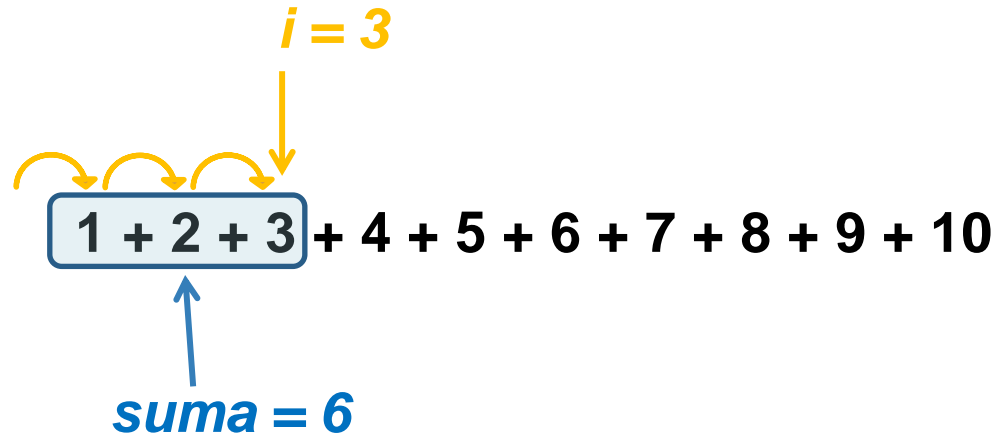


# Programación. Python

## Bucles

# Bucles. Un problema inicial

$$\text{suma\_hasta}(10) = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$$



$$\text{suma\_hasta}(n) = 1 + 2 + 3 + \dots + n$$

# Bucles

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$$

## Condicionales

## Acotados

```
while <<condición>>:  
    <<instrucciones>>
```

```
n = 10 # límite superior  
suma = 0  
i = 1  
while i <= n:  
    suma = suma + i  
    i = i + 1  
  
print(suma)
```

# Bucles

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$$

## Condicionales

```
while <<condición>>:  
    <<instrucciones>>
```

```
n = 10 # límite superior  
suma = 0  
i = 1  
while i <= n:  
    suma = suma + i  
    i = i + 1  
  
print(suma)
```

## Acotados

```
for <<variable>> in <<secuencia>>:  
    <<cuerpo del bucle>>
```

```
n = 10 # límite superior  
suma = 0  
for i in range(1, 10+1):  
    suma = suma + i  
  
print(suma)
```

# Bucles

1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10

## Condicionales

```
while <<condición>>:  
    <<instrucciones>>
```

```
► def suma_hasta(n):  
    suma = 0  
    i = 1  
    while i <= n:  
        suma = suma + i  
        i = i + 1  
    return suma
```

```
► print(suma_hasta(10))
```

55

## Acotados

```
for <<variable>> in <<secuencia>>:  
    <<cuerpo del bucle>>
```

```
def suma_acot_hasta(n):  
    suma = 0  
    for i in range(1, n+1):  
        suma = suma + i  
    return suma
```

```
print(suma_acot_hasta(10))
```

55

# Bucles while

Extraer como factor  
la mayor potencia de 2:

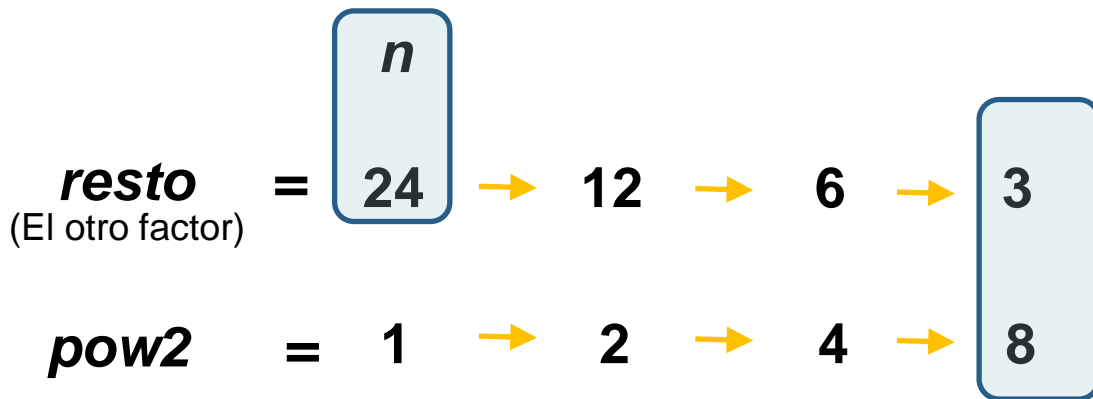
```
mayor_pot_2(12), mayor_pot_2(8), mayor_pot_2(7)
```

```
Out[28]: ((4, 3), (8, 1), (1, 7))
```

```
Out[28]: ((4, 3), (8, 1), (1, 7))
```

# Bucles while

Extraer como factor  
la mayor potencia de 2:



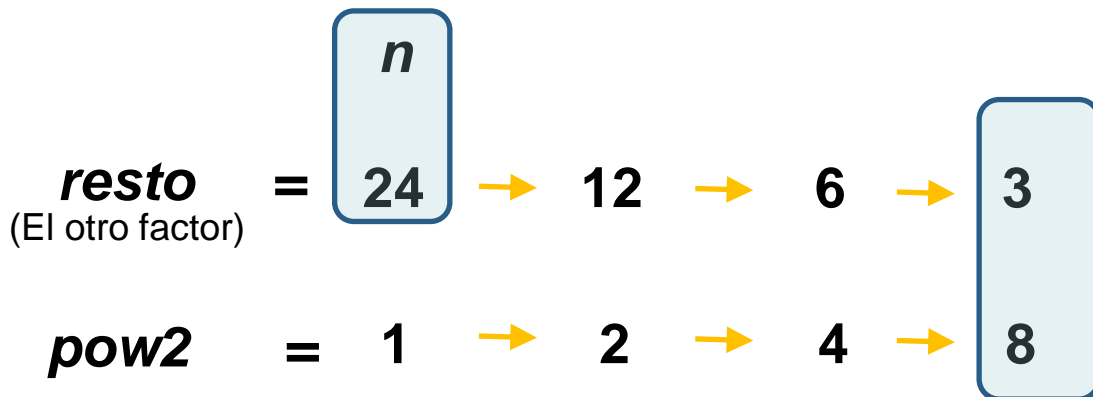
```
mayor_pot_2(12), mayor_pot_2(8), mayor_pot_2(7)
```

```
Out[28]: ((4, 3), (8, 1), (1, 7))
```

```
Out[28]: ((4, 3), (8, 1), (1, 7))
```

# Bucles while

Extraer como factor  
la mayor potencia de 2:



```
mayor_pot_2(12), mayor_pot_2(8), mayor_pot_2(7)
```

```
Out[28]: ((4, 3), (8, 1), (1, 7))
```

```
Out[28]: ((4, 3), (8, 1), (1, 7))
```

```
def mayor_pot_2(n):  
    pow2, resto = 1, n  
    while (resto % 2) == 0:  
        pow2 = pow2 * 2  
        resto = resto // 2  
    return pow2, resto
```



# Bucles for

Calcular la media  
de una lista de notas (números reales)

*lista\_de\_notas* = [9.5, 6.0, 4.5, 10.0]

```
nota media([9.5, 6.0, 4.5, 10.0])
```

```
7.5
```

# Bucles for

Calcular la media

de una *lista\_de\_notas* = [9.5, 6.0, 4.5, 10.0]

*nota* = 9.5, 6.0, 4.5, 10.0

```
def nota_media(lista_de_notas):  
    suma = 0.0  
    for nota in lista_de_notas:  
        suma = suma + nota  
    return suma / len(lista_de_notas)
```

```
nota_media([9.5, 6.0, 4.5, 10.0])  
7.5
```

# Bucles for

Calcular la media

de una *lista\_de\_notas* = [9.5, 6.0, 4.5, 10.0]

*nota* = 9.5, 6.0, 4.5, 10.0

*i* = 0, 1, 2, 3

*lista\_de\_notas*[*i*]

```
def nota_media(lista_de_notas):  
    suma = 0.0  
    for nota in lista_de_notas:  
        suma = suma + nota  
    return suma / len(lista_de_notas)
```

```
def media(lista_de_notas):  
    suma = 0.0  
    long_lista = len(lista_de_notas)  
    for i in range(long_lista):  
        suma = suma + lista_de_notas[i]  
    return suma / long_lista
```

```
nota_media([9.5, 6.0, 4.5, 10.0]), media([9.5, 6.0, 4.5, 10.0])  
(7.5, 7.5)
```