

# Taller de Física Computacional

## Bucles

Cristián G. Sánchez y Carlos J. Ruestes

2020

## Bucle

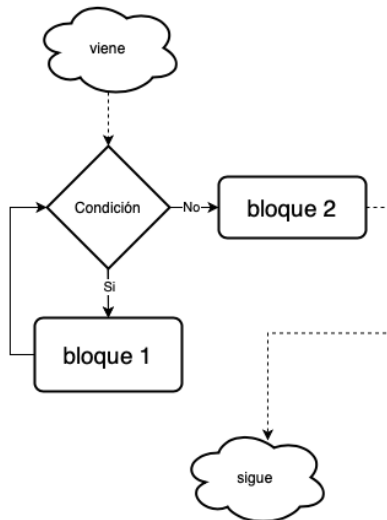
El bucle es una estructura que nos permite ejecutar en forma repetida un conjunto de declaraciones.

- La repetición puede ser infinita o finita.
- La terminación de la repetición se hace en base a una expresión lógica.
- Se puede **escapar** de una repetición.

Con los bucles y condicionales (sobre lo que ya vimos) en principio podemos programar lo que querramos.

# La declaración de bucle más completa

```
while_(condición):  
    #_Bloque_1  
    #_esto_se_repite_mientras  
    #_la_condición_sea_True  
else:  
    #_Bloque_2  
    #_esto_se_ejecuta  
    #_por_única_vez  
    #_si_la_condición_es_False  
    #_y_es_opcional
```



## Bucle

Dentro del bloque que sigue al `while` pueden utilizarse las siguientes palabras clave para alterar el flujo:

- La invocación de la palabra clave `break` sale incondicionalmente del bucle sin ejecutar el bloque que sigue al `else` (si existiera).
- La invocación de la palabra clave `continue` saltea el resto del bloque y vuelve a la evaluación de la condición presente en el `while`.

Las palabras clave `break` y `continue` suelen ser parte de declaraciones condicionales que permiten terminar o proseguir el bucle en base a otras condiciones.

# Bucle infinito

## Bucle infinito

```
while True:  
    #_esto_se_repita_hasta  
    #_que_el_programa_se_interrumpa
```

La combinación con una (o más) invocaciones a `break` dentro de una declaración condicional permite hacer cualquier bucle de esta manera.

# Iteración sobre una serie de naturales consecutivos

## Bucle sobre $n \in \mathbb{N}$

Podemos construir un bulce sobre una serie de enteros consecutivos de la siguiente forma:

```
i = 0 # se parte de i = 0
while (i <= maxn):
    # esto se repite hasta que i == nmax
    #
    i += 1 # aquí se incrementa i
```

## Bucles anidados

Podemos anidar bucles dentro de bucles para llevar a cabo iteraciones sobre conjuntos más complejos:

```
i=_j=_k=_0#_se_parte_de_i,j,k=_0
while_(i_<=_maxi):
    while_(j_<=_maxj):
        while_(k_<=_maxk):
            #_esto_se_repite_hasta_que
            #_(i_==_maxi)_and_(j_==_maxj)_and_(k_==_maxk)
            k_+=_1#_aquí_se_incrementa_k
        j_+=_1#_aquí_se_incrementa_j
    i_+=_1#_aquí_se_incrementa_i
```



Los bucles anidados son ingredientes fundamentales de cualquier programa. Muchas operaciones comunes en física computacional, desde las multiplicaciones matriciales, contracciones tensoriales, transformadas de Fourier o la simple evaluación de una función de varias variables en una grilla requieren de bucles anidados para ejecutarse. Los bucles en Python son *lentos*. Frustrantemente lentos de hecho, en algunos casos. Hasta el punto de volver las operaciones antes mencionadas prácticamente imposibles salvo para casos triviales. Esta es una de las razones de la existencia del paquete NumPy que permite llevar a cabo **algunos** bucles utilizando lo que se denomina *vectorización* llamando a rutinas específicas en C para cada caso.



# Síntesis y recursos:

- Bucle while en Wikipedia en español
- Bucle while en Wikipedia en inglés