

IWI 131-
PROGRAMACIÓN

Ayudantía 4

Ayudante: Anastasiia Fedorova

Paralelo: 212

Fecha: 29.04.2020



¿QUÉ SON LOS CICLOS? REPETICIONES EN PYTHON



ciclo

nombre masculino

1. Serie de fases o estados por las que pasa un acontecimiento o fenómeno y que se suceden en el mismo orden hasta llegar a una fase o estado a partir de los cuales vuelven a repetirse en el mismo orden.

En Python, por ahora utilizaremos solamente el bucle while, "ejecutar mientras que...".

El ciclo for tiene el comportamiento parecido a while (o while a for), pero a veces conviene usar while en vez de for y en otras ocasiones – for, en vez de while.



BUCLE WHILE

Un bucle while permite repetir la ejecución de instrucciones mientras se cumpla una condición (es decir, mientras la condición tenga el valor True).

La sintaxis del while es:

```
while condicion:  
    #codigo a ejecutar
```

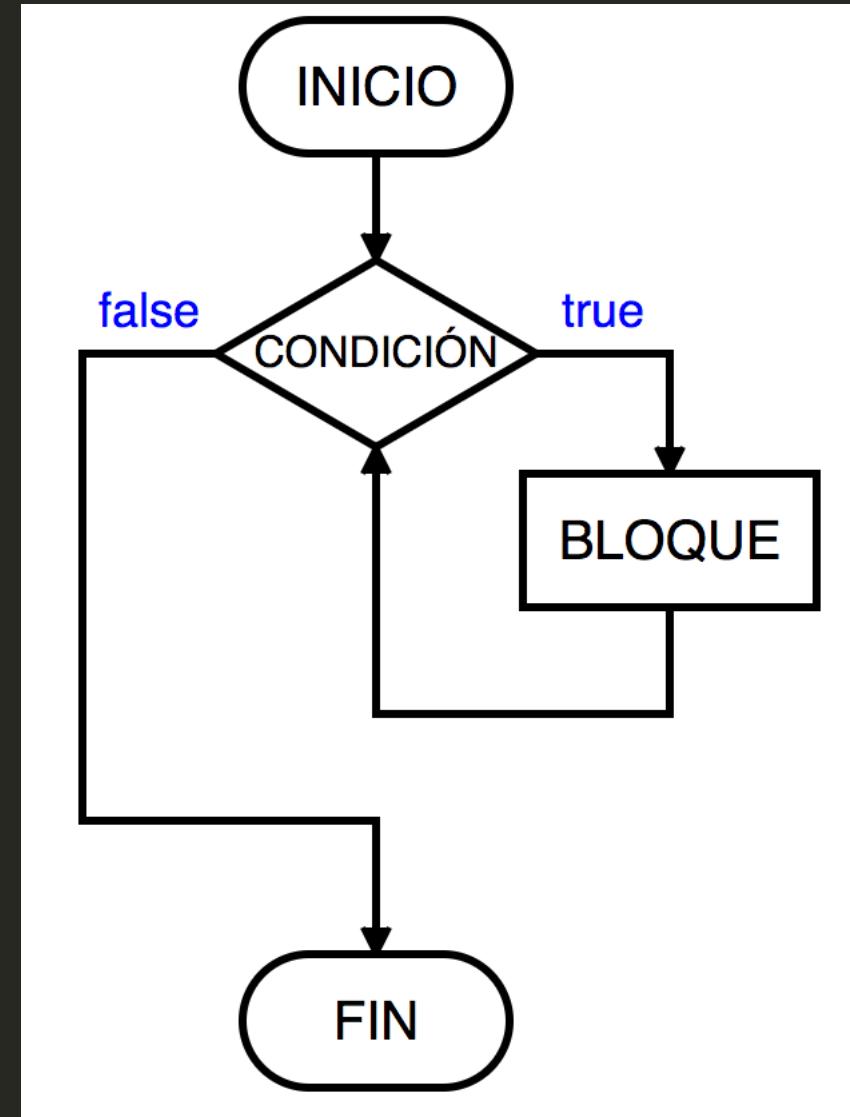


Diagrama de flujo

BUCLE WHILE

La variable o las variables que aparezcan en la condición se llaman variables de control. Las variables de control se definen fuera del bucle y se cambian dentro de este.

Por ejemplo:

```
n = 10
while(n > 0):
    #hacer algo mientras n es mayor que 10
    #una vez que n == 0, el ciclo termina
    n -= 1
```

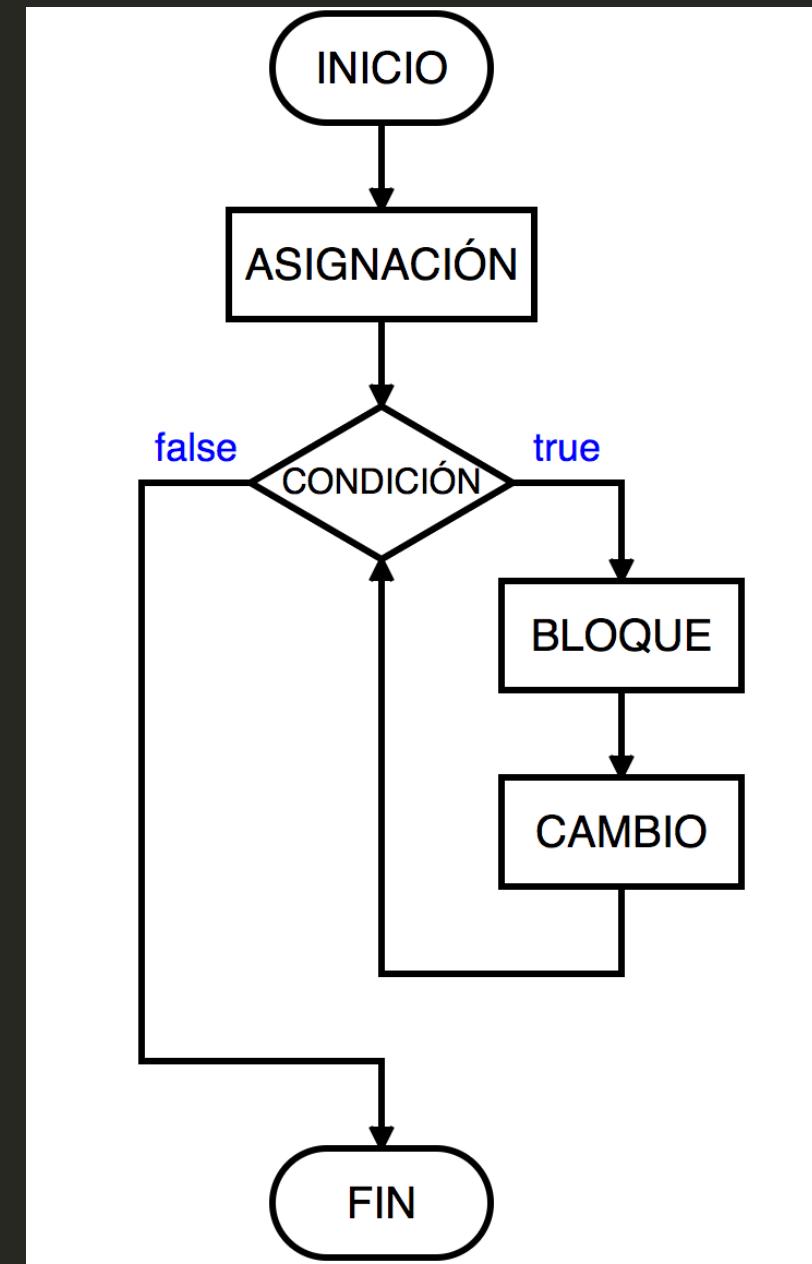


Diagrama de flujo

BUCLE INFINITO

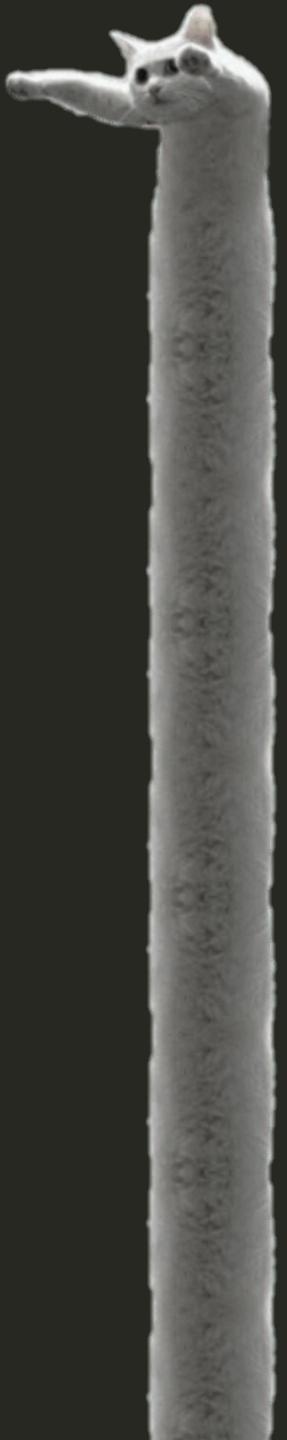
Bucle infinito es un error que consiste en realizar un ciclo que se repite de forma indefinida ya que su condición de control siempre se evalua como True.

Esto puede pasar si, por ejemplo, nunca cambio la variable de control:

```
n = 10  
while n > 0:  
    #codigo
```

O si corro un while, cuya condicion es siempre cierta:

```
while 1:  
    #codigo
```



EJEMPLOS

Podemos utilizar el loop controlado para conseguir, por ejemplo, recepción continua de input, hasta que la condición impuesta a while se vuelva False.

Por ejemplo, podemos pedir dígitos a un usuario, pero terminar el programa una vez que este ingrese un carácter cualquiera o algo de largo 1, que no sea un entero entre 0 y 9.

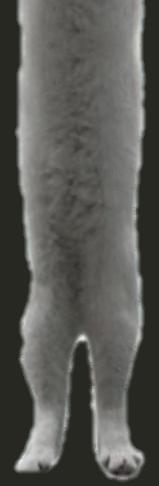
```
flag = True
while flag:
    i = input("Ingrese un digito: ")
    if ord(i) >= 48 and ord(i) <= 57:
        i = int(i)
        #hacer algun calculo
    else:
        flag = False
        #sale del ciclo, recibimos algo, que no es un entero de 0 a 9
```

EJEMPLOS

```
n = 10
print("/\\____/\\")
print("//      \\\")
print("|\ 0 0 |")
print("\\"  \u25bd |")
print("\\" _|_ /")
print(" / \\"____")
print(" / ----- \\")
print("|\____ \\" \\"/")
print("|\ \\"/")
while n > 0:
    print("|\ |")
```

Un ejemplo de uso de “error” de loop infinito –
no es muy aplicable en la práctica.

```
(base) user-MacBook-Pro:desktop user$ python3 1.py
```



```
\____/
 |
 | 0 0
 | \u25bd
 | - -
 | \_____
 | \____ \u25bc
 | \____ \u25bc \u25bc /
```

EJERCICIO 1: ESTACA

Su ayudante actualmente está cursando INF280 – Estadística Computacional, y en esta se trabaja mucho con Python. Un día el profesor se le pidió calcular las medias de un conjunto de números, ingresados uno a uno por el usuario. Se desconoce la cantidad de números que ingresa el usuario, solo se sabe que este escribe un “x” cuando quiere terminar.

Se les pide escribir un programa, que calcula la media de números, ingresados hasta el momento (osea para cada nuevo numero se calcula una media con conjunto ingresado hasta el momento). Recordar que la media

viene dada por: $\bar{x} = \frac{\sum_{i=1}^N x_i}{N}$.



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

INF-280 Estadística Computacional I-2020

Evaluación Tema 1

Anastasiia Fedorova

ROL: 201873505-1

Paralelo: 200

Abriil, 2020-1

Input:	Output:
2	2.0
9	5.5
1	4.0
2	3.5
3	3.4
x	Media: 3.4

EJERCICIO 1: SOLUCIÓN

```
flag = True
suma = 0
cont = 0
while flag:
    i = input("Ingrese un numero: ")
    if i == "x": # alternativamente, if not(i.isnumeric()) (no es solo numeros)
        flag = False
    else:
        suma += int(i)
        cont += 1
        promedio = suma / cont
        print(promedio)
print("Promedio muestral:", promedio)
```

¡Ojo con la indentación!

EJERCICIO 2: RUTEO

C1 2019-1, pregunta 1

```
a = 943
flag = True
n = 0
b = 0
c = 0
while flag :
    d = a % 10
    a = a // 10
    n = n * 10 + d
    if d % 2 == 0:
        b = b + 1
    else:
        c = c + 1
    if a == 0:
        flag = False
print (n,b,c)
```

EJERCICIO 2: SOLUCIÓN

```
a = 943
flag = True
n = 0
b = 0
c = 0
while flag :
    d = a % 10
    a = a // 10
    n = n * 10 + d
    if d % 2 == 0:
        b = b + 1
    else:
        c = c + 1
    if a == 0:
        flag = False
print (n,b,c)
```

Pantalla:
349 1 2

Explicación
Cambia de lugar
los dígitos
exteriores (de a
dos, hacia centro)

Imprime cantidad
de dígitos pares y
cantidad de dígitos
impares.

a	flag	n	b	c	d
943					
	True				
		0			0
			0		0
				3	3
94					
		3			
			1		1
				4	4
9					
		34			
			1		1
				9	9
0					
		349			
			2		2
				False	

EJERCICIO 3: GOTTA GO FAST



Cuando l@s sansan@s tenían que dar sus cursos de deporte offline, el ejercicio clásico consistía en correr alrededor de la U.

Escribir un programa que reciba el nivel de dificultad de ejercicio que el/la sansan@ tiene que hacer. Si la dificultad es “facil” – son 3 círculos, si es “media” – 5 y “extrema” – 7 círculos. Por cada repetición se les pida recibir la velocidad del estudiante y el tiempo (en segundos) empleado. Imprimir el tiempo total que gasto el estudiante y, además, su velocidad mínima y máxima en todo el trayecto.

Input:

facil

385

3.4

560

2.8

475

3.1

Output:

Tiempo: 1420 [s]

V max: 3.4 [m/s]

V min: 2.8 [m/s]

EJERCICIO 3: SOLUCIÓN

Para calcular el máximo y el mínimo (¡sin usar max() y min()!) se usa if-elif

```
dif = input("Ingrese la dificultad: ") while n:  
    t = int(input("Tiempo empleado: "))  
    v = float(input("Velocidad de  
estudiante: "))  
    t_total += t  
    if v < v_min:  
        v_min = v  
    elif v > v_max:  
        v_max = v  
    n -= 1  
  
print("Tiempo:", t_total, "[s]")  
print("V max:", v_max, "[m/s]")  
print("V min:", v_min, "[m/s]")
```

if dif == "facil":
 n = 3
elif dif == "media":
 n = 5
else:
 n = 7

t_total = 0
v_min = float('inf') # o un numero muy
grande, como 9999999
v_max = -float('inf') # o un numero
muy chico,
-9999999

EJERCICIO 4: !

Recibir un número N, ingresado por el usuario, que indica cuantos factoriales este quiere calcular. Uno a uno, recibir los N enteros (en rango de 0 a 10), e imprimir sus factoriales.

Utilice solo ciclo while (o varios).

Recordatorio:

$$0! = 1$$

$$1! = 1$$

$$2! = 1 * 2 = 2$$

$$3! = 1 * 2 * 3 = 6$$



Input:

2

3

4

Output:

6

24

EJERCICIO 4: SOLUCIÓN

```
factorial = 1
n = int(input("Cuantos factoriales quiere calcular? "))
while n:
    num = int(input("Ingrese el numero para calcular su factorial: "))
    if num < 0:
        print("Factorial no esta definido para los negativos.")
    elif num == 0 :
        print("Su factorial es:", factorial)
    else:
        while num > 0:
            factorial = factorial * num
            num = num - 1
        print("Su factorial es:", factorial)
        factorial = 1
    n -= 1
```