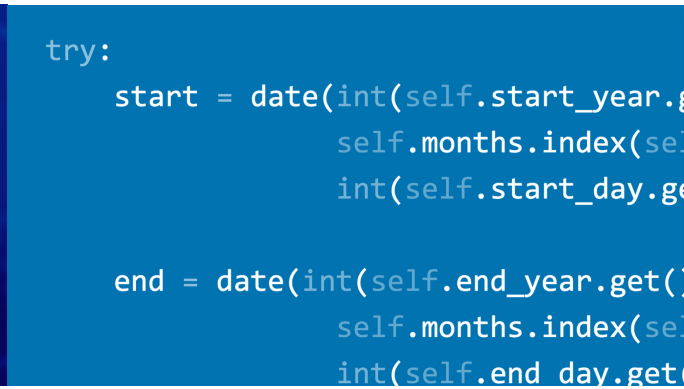
UNIVERSIDAD DIEGO PORTALES

Introducción a Python

Introducción

2026

- Estructuras de control
  - Condicionales IF-ELSE
  - Ciclo While



Ciclo Mientras

```
while <condición/valor lógico>:

    //bloque del while
```

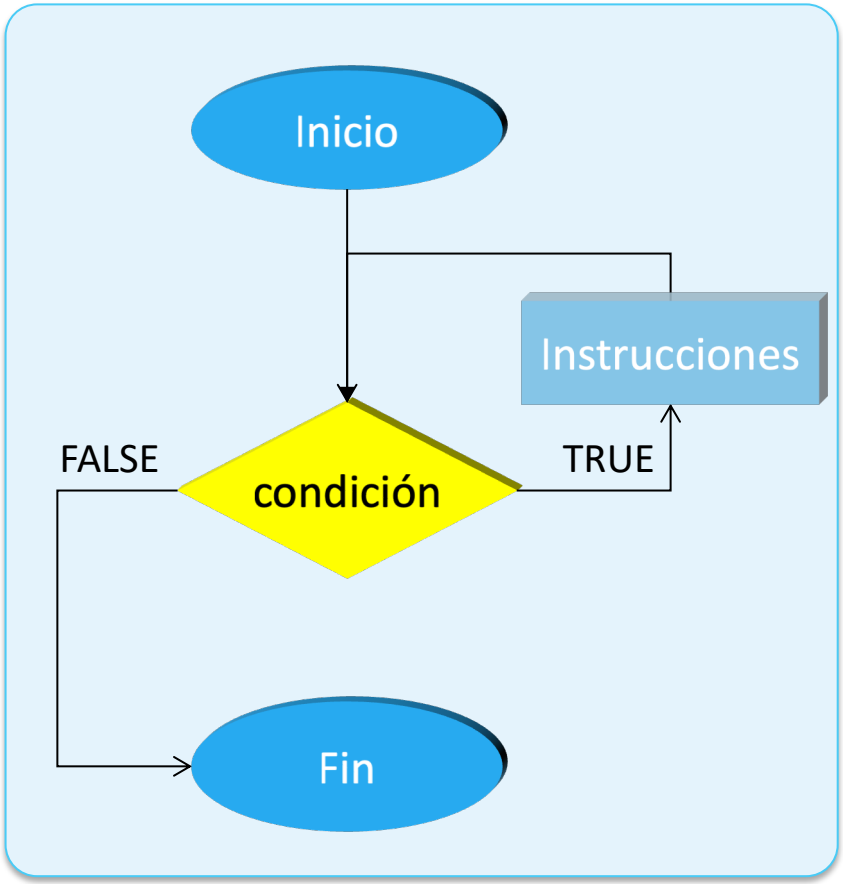
El ciclo while se repite mientras la condición sea verdadera. Termina cuando la condición es falsa.

Realice las siguientes instrucciones:

1. Inicio
2. Retire una hoja de su cuaderno
3. Sitúe la hoja sobre su escritorio
4. mientras (¿puedo doblarla?==verdadero)
    1. Doble la hoja por la mitad
    2. Regrese al paso 4
5. fin

condición

Apuesto a que no podrá doblarla más de 7 veces



Ejemplo

Mostrar los números
del 1 al 10 por pantalla

```
i = 1
while i < 10:
    print(i)
    i = i + 1
```

Acumulador

¿Qué se muestra por pantalla en este caso?

¿Cómo lo arreglamos?

Python nos entrega una manera de poder terminar un ciclo de manera anticipada

```
while True:

  print('Escribe 0 para salir')
  respuesta = int( input() )

  if respuesta == 0:
      break
```

¡La condición siempre se cumple!

Break nos permite “escapar” de un ciclo while antes que se deje de cumplir la condición

Ciclo Mientras

Tiempo : 8 minutos

Escribe un programa en Python que solicite números al usuario hasta que este ingrese el número 0.
Una vez que el usuario ingresa el 0, el programa debe desplegar (mostrar) el mayor número ingresado (sin contar el cero).

Tiempo : 8 minutos

Escribe un programa en Python que solicite números al usuario hasta que este ingrese el número 0.
Una vez que el usuario ingresa el 0, el programa debe desplegar (mostrar) el mayor número ingresado (sin contar el cero).

Solución propuesta

```
fin = 0
max = 1

while fin==0:
    num = int(input('Ingresa un numero. Para terminar, ingresa 0:'))

    if num==0:
        fin = 1
    else:
        if num>max:
            max = num
print('El mayor valor ingresado fue: ', max)
```

Tiempo : 8 minutos

Escribe un programa en Python que solicite números al usuario hasta que este ingrese el número 0.
Una vez que el usuario ingresa el 0, el programa debe desplegar (mostrar) el mayor número ingresado (sin contar el cero).

Solución propuesta #2

```
num = int(input('Ingresa un numero. Para terminar, ingresa 0:'))
max = num

while num!=0:
      num = int(input('Ingresa un numero. Para terminar, ingresa 0:'))

      if num>max and num!=0:
            max = num

print('El mayor valor ingresado fue:', max)
```

Ahora tú

Tiempo : 10 minutos

Escribe un programa en Python que se mantenga solicitando números al usuario hasta que éste ingrese un 0. En ese caso, el programa le indica cuántos números ingresó (sin contar el 0) y cuánto suman. Ejemplos de ejecución del programa:

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
Ingresa un número 1
Ingresa un número 2
Ingresa un número 3
Ingresa un número 4
Ingresa un número 5
Ingresa un número 6
Ingresa un número 7
Ingresa un número 0
Ingresaste  7  números que suman  28
>
```

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
Ingresa un número 0
Ingresaste  0  números que suman  0
>
```

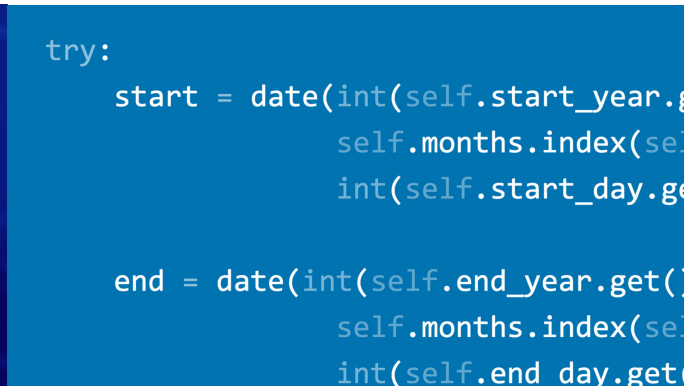
Tiempo : 10 minutos

Escribe un programa en Python que se mantenga solicitando números al usuario hasta que éste ingrese un 0. En ese caso, el programa le indica cuántos números ingresó (sin contar el 0) y cuánto suman. Ejemplos de ejecución del programa:

Solución propuesta

```
N = int(input("Ingresa un número"))
contador = 0
suma = N

while N!=0:
    N = int(input("Ingresa un número"))
    contador = contador+1
    suma = suma+N

print("Ingresaste ",contador," números que suman ", suma)
```

- Estructuras de control
  - Condicionales IF-ELSE
  - Ciclo While
  - Lista de Números



¿ciclos while?

udp UNIVERSIDAD DIEGO PORTALES

▼ Algoritmo

```
Imprimir los números enteros del 1 al 10:
1. Comenzar con número igual a 1
2. Decir el número
3. Sumar 1 al número
4. Si número es menor a 11 repetir desde el paso 2
```

▼ Python

```
numero = 1
while numero < 11:
    print(numero)
    numero = numero + 1
```

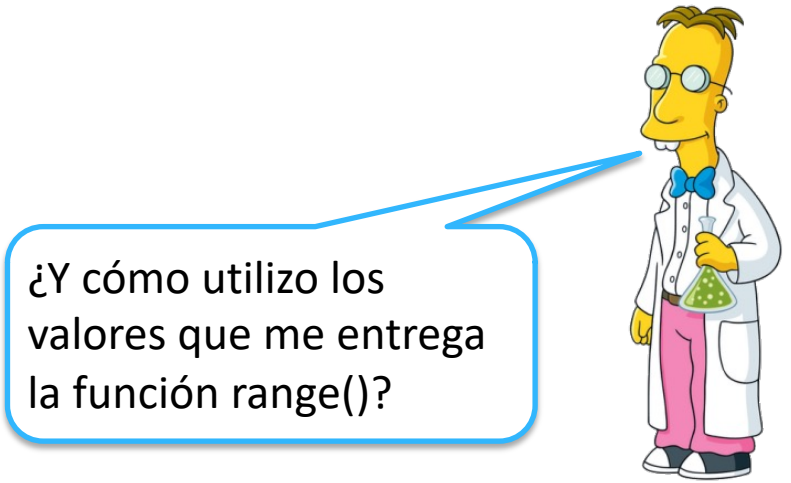
Función range()

`range([start], stop, [step])`

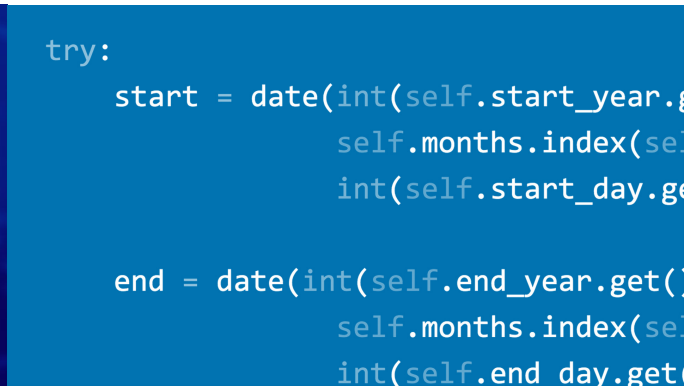
La función range nos entrega una colección de elementos que van desde el valor inicial start (o 0 si no está definido), hasta el valor stop (no incluido), avanzando con paso step (o 1 si no está definido)

en Python el rango no incluye el valor final
→ recorro de 1 a 10

```
numeros = range(1,11)
print(list(numeros))
# imprime [1, 2, 3, 4, 5, 6, 7, 8, 9 ,10]

impares = range(11, 0, -2)
print(list(impares))
# imprime [11, 9, 7, 5, 3, 1]

multiplos_3 = range(0, 10, 3)
print(list(multiplos_3))
# imprime [0, 3, 6, 9]
```



- Estructuras de control
  - Condicionales IF-ELSE
  - Ciclo While
  - Lista de Números
  - Ciclo FOR



Ciclo For

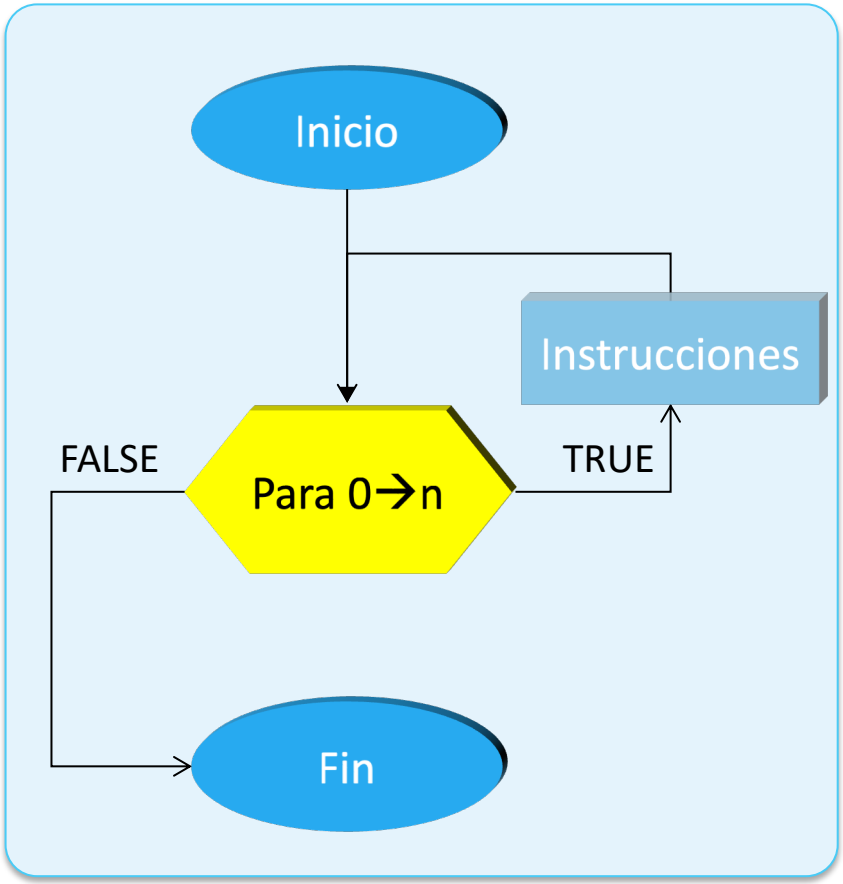
```
for i in <colección>:

    //bloque del for
```

El ciclo for se repite mientras hayan elementos en una colección. Termina cuando no hay más elementos

Realice las siguientes instrucciones:

1. Inicio
2. Retire una hoja de su cuaderno
3. Sitúe la hoja sobre su escritorio
4. Contador = 5
5. Para n=0 hasta n
   {
      1. Doble la hoja por la mitad
   }
6. Fin

condición



Ciclo For

Algoritmo

Imprimir los números enteros del 1 al 10:

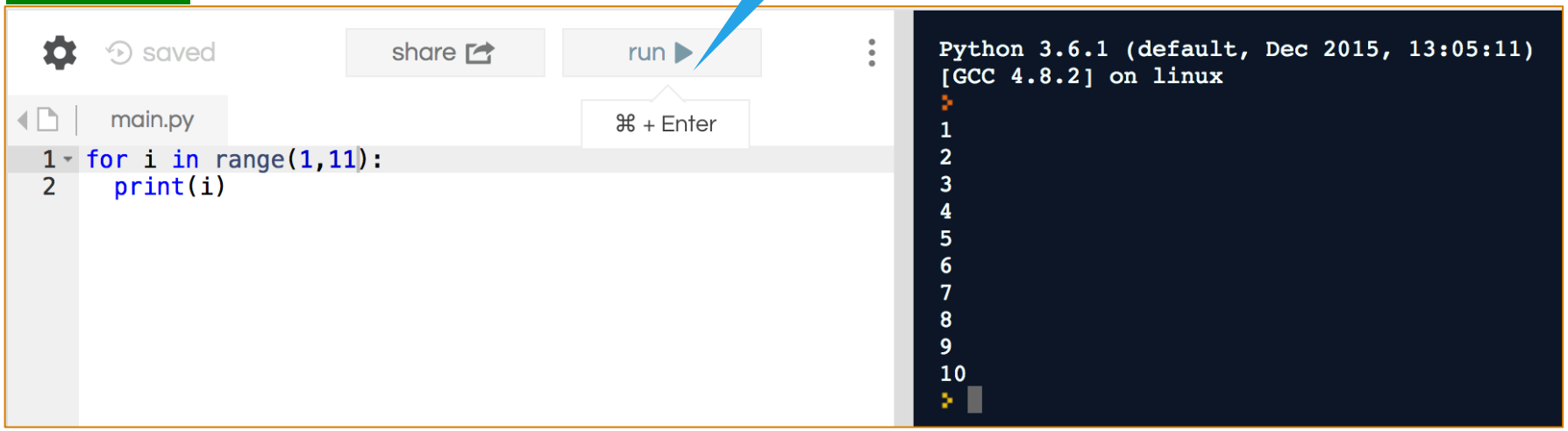
1. Comenzar con número igual a 1
2. Decir el número
3. Sumar 1 al número
4. Si número es menor a 11 repetir desde el paso 2

Python

```
for i in range(1,11):
    print(i)
```

En la instrucción: `for i in range(a,b)`, el valor `i` toma los valores: a , a+1, a+2, ..., b-1.

repl.it



- Enunciado

  - Realice un programa que solicite al usuario la base y el exponente de un número. Determine la potencia de dicho número.
  - Haga una versión con ciclo for y otra con ciclo while.
  - ¿Cuál cree usted que es mejor?

$$r = a^n$$

Cosas que debo observar.

Observaciones	Respuestas
Solicito datos al usuario	Sí, la base y el exponente
Qué debo hacer?	Multiplicar la base *n* veces
Cómo obtengo los números del 1 al *n*	Ocupo un ciclo for ya que conozco el número del exponente (es n)

- El ciclo for se repite mientras hayan elementos en la colección. Termina cuando no hay más elementos que recorrer.

```
print("ingrese la base")
a = int(input())

print("ingrese el exponente")
n = int(input())

base = 1
```

Leemos los datos y la base del exponente

- El ciclo for se repite mientras hayan elementos en la colección. Termina cuando no hay más elementos que recorrer.

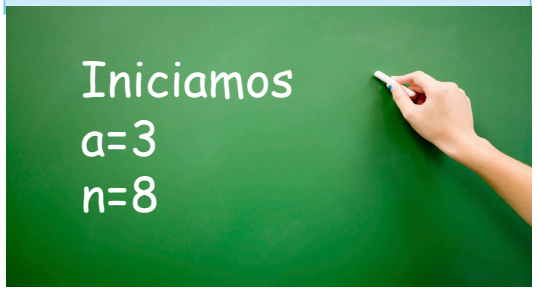
```
print("ingrese la base")
a = int(input())

print("ingrese el exponente")
n = int(input())

base = 1

for i in range(0,n):
        base = a*base
        print("valor:",base)

print("valor final", base)
```

- El ciclo for se repite mientras hayan elementos en la colección. Termina cuando no hay más elementos que recorrer.

```
print("ingrese la base")
a = int(input())

print("ingrese el exponente")
n = int(input())

0  base = 1

for i in range(0,n):
    1  base = a*base
       print("valor:",base)

print("valor final", base)
```

Ejecutamos un ciclo n veces

i	base
0	1*3
1	3*3=9
2	3*9=27
3	3*27=81
4	3*81=243
5	3*243=729
6	3*729=2187
7	3*2187=6561

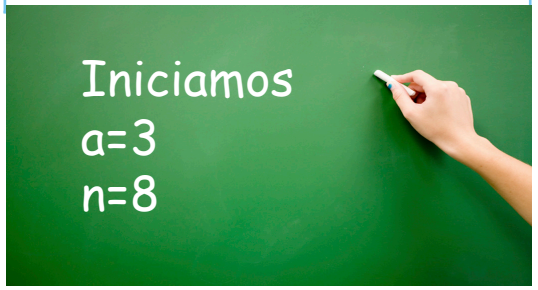


Problema

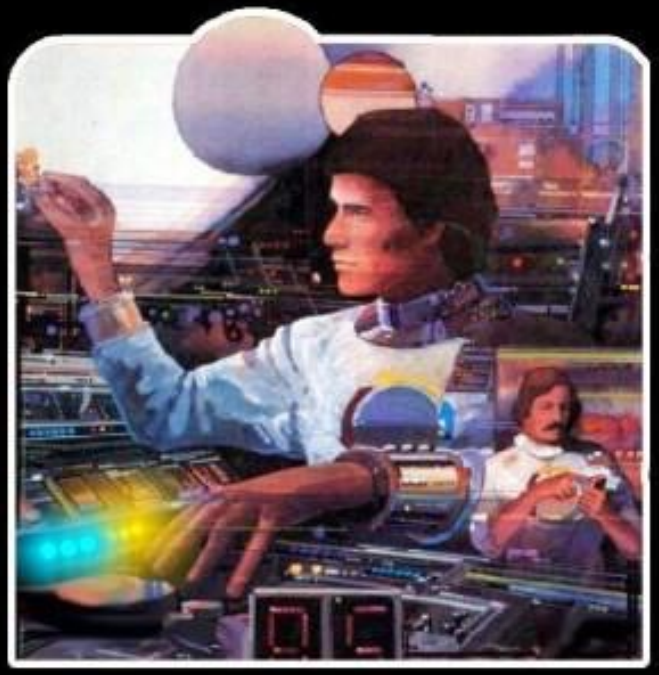
- El ciclo while se ejecutan mientras la condición sea verdadera. Termina cuando la condición es falsa

```
print("ingrese la base")
a = int(input())

print("ingrese el exponente")
n = int(input())

base = 1
i    = 0
while i<n:
        base = a*base
        print("valor:",base)
        i = i+1

print("valor final", base)
```

i	base
0	3
1	3*3=9
2	3*9=27
3	3*27=81
4	3*81=243
5	3*243=729
6	3*729=2187
7	3*2187=6561



THE TWO STATES OF EVERY PROGRAMMER



I AM A GOD.

I HAVE NO IDEA WHAT I'M DOING.

Tiempo : 5 minutos

Usando ciclo for, escribe un programa en Python que pida un número al usuario (N) e imprima un rectángulo relleno de símbolos #. El ancho del rectángulo es de 6 símbolos # y la altura es de N símbolos # (se imprime un total de 6*N símbolos). Ejemplo:

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
¿Cuántas filas? 4
######
######
######
######
>
```

Tiempo : 5 minutos

Usando ciclo for, escribe un programa en Python que muestre los divisores del número que el usuario ingresa por teclado. Ejemplo:

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
Ingresa un número 15
Los divisores del número 15 son:
1
3
5
15
>
```

Tiempo : 5 minutos

Realice un programa que lea un número entero por teclado y determine todas los valores fraccionarios hasta llegar a dicho número con 5 valores decimales

Ejemplo. Suponga que ingresa el número 12. El programa debe mostrar lo siguiente

Salida por pantalla

```
1/12  0.00694
2/12  0.16666
3/12  0.25000
      0.33333
      ..

12/12 1.00000
```