

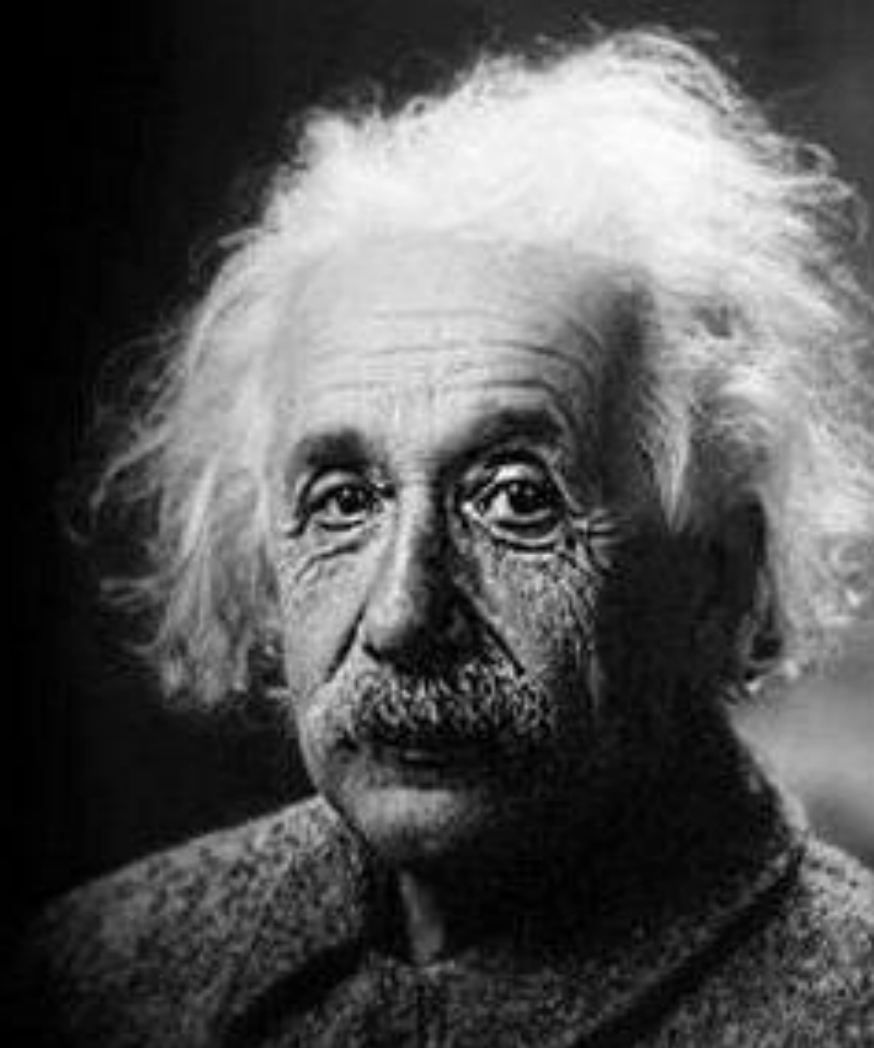
TEMA: Python Sentencias Repetitivas

Matemáticas Aplicadas Desarrollo y Soporte de Sistemas

DOCENTE:DANIEL HANS HUAMAN CABRERA

**"Si quieres vivir
una vida feliz,
atala a una meta
no a una persona
o un objeto"**

Albert Einstein





Objetivo de la sesión

- Aprenderás a utilizar las sentencias repetitivas como : for , do while en Python
- Acumuladores son variables en Python.
- Ejercicios Prácticos y propuestos.

Contenido de la clase

1 Forma iniciar Python

```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [Versión 10.0.19042.867]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>python
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr  4 2021, 13:27:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

2 Forma iniciar Python

```
C:\Users\Usuario>py
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr  4 2021, 13:27:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Contenido de la clase

C:\WINDOWS\system32\cmd.exe - py

Microsoft Windows [Versión 10.0.19042.928]

(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>py

Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 4 2021, 13:27:16) [MSC v.1928 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> range(30)

range(0, 30)

>>> range(5)

range(0, 5)

>>> range(0,30)

range(0, 30)

>>> list(range(30))

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]

>>> list(range(5))

[0, 1, 2, 3, 4]

>>> list(range(50))

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49]

>>> list(range(3))

[0, 1, 2]

>>>

Contenido de la clase

```
C:\WINDOWS\system32\cmd.exe - py
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>py
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr  4 2021, 13:27:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> for i in list(range(5)):
...     print(i)
...     File "<stdin>", line 2
...         print(i)
...         ^
IndentationError: expected an indented block
>>> print(i)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'i' is not defined
>>> for i in list(range(5)): print(i)
...
0
1
2
3
4
>>> for i in list(range(5)): print('hello , world')
...
hello , world
hello , world
hello , world
hello , world
hello , world
>>>
```

PYTHON VERSION 3.2

List(range(30))

PYTHON VERSION 2.1

range(30)

for loop

- Se puede utilizar para recorrer strings (una string es una secuencia)
- Se necesita el keyword **in**

Fuente:

<https://www.youtube.com/watch?v=GQGhU1526Oo>

Contenido de la clase

```
>>> for i in list(range(11)): print(i)
...
0
1
2
3
4
5
6
7
8
9
10
>>>
```


Bucles While

El concepto detrás de un ciclo while es simple: mientras una condición es verdadera -> Ejecuta mis comandos.

El bucle while comprueba la condición cada vez, y si devuelve "true", ejecutará las instrucciones dentro del bucle.

Antes de comenzar a escribir código, veamos el diagrama de flujo para ver cómo funciona.

Fuente:

<https://www.youtube.com/watch?v=XkDOJC4hpw0>

Contenido de la clase

Ahora escribamos algo de código. Así es como se escribe un bucle while simple para imprimir números del 1 al 10.

```
#!/usr/bin/python

x = 1

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

```
> python script.py
1
2
3
4
5
6
7
8
9
10
```

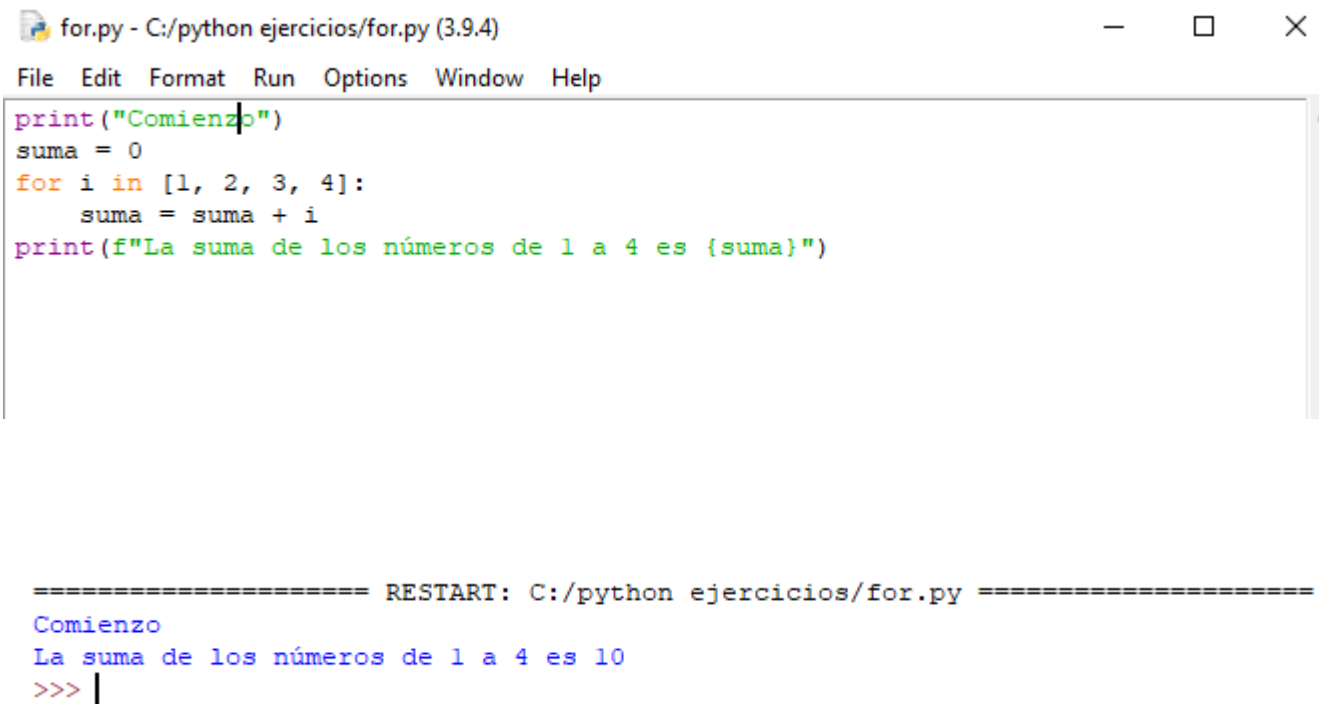
Acumulador

Se entiende por acumulador una variable que acumula el resultado de una operación. El ejemplo siguiente es un ejemplo de programa con acumulador (en este caso, la variable que hace de acumulador es la variable *suma*)

Fuente:

<https://www.youtube.com/watch?v=exwxcLobw-w>

Contenido de la clase



The screenshot shows a Python IDE window titled "for.py - C:/python ejercicios/for.py (3.9.4)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The editor contains the following Python code:

```
print("Comienzo")
suma = 0
for i in [1, 2, 3, 4]:
    suma = suma + i
print(f"La suma de los números de 1 a 4 es {suma}")
```

Below the editor, the output of the script is displayed, preceded by a restart separator:

```
===== RESTART: C:/python ejercicios/for.py =====
Comienzo
La suma de los números de 1 a 4 es 10
>>> |
```

EJERCICIOS EN CLASE

Contenido de la clase

```
for.py - C:/python ejercicios/for.py (3.9.4)
File Edit Format Run Options Window Help
coleccion = [2, 4, 5, 7, 8, 9, 3, 4]
for e in coleccion:
    if e == 7:
        break
    print(e)
```

```
IDLE Shell 3.9.4
File Edit Shell Debug Options Window Help
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr  4 2021, 13:27:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/python ejercicios/for.py =====
>>>
>>>
===== RESTART: C:/python ejercicios/for.py =====
>>>
>>>
===== RESTART: C:/python ejercicios/for.py =====
>>>
>>>
===== RESTART: C:/python ejercicios/for.py =====
2
4
5
>>>
```

Contenido de la clase

```
for.py - C:/python ejercicios/for.py (3.9.4)
File Edit Format Run Options Window Help
valores = {'A': 4, 'E': 3, 'I': 1, 'O': 0}
for k in valores:
    print(k)

===== RESTART: Shell =====
>>>
===== RESTART: C:/python ejercicios/for.py =====
A
E
I
O
>>> |
```

Contenido de la clase

Ejercicio Practico

File Edit Format Run Options Window Help

```
print("Comienzo")
encontrado = False
for i in range(1, 6):
    if i % 2 == 0:
        encontrado = True
if encontrado:
    print(f"Entre 1 y 5 hay al menos un múltiplo de 2.")
else:
    print(f"Entre 1 y 5 no hay ningún múltiplo de 2.")
```

```
===== RESTART: C:/python ejercicios/for.py =====
Comienzo
Entre 1 y 5 hay al menos un múltiplo de 2.
>>> |
```


Ejercicio Practico

Ejercicio

Escribir un programa que pida al usuario una palabra y la muestre 10 veces por pantalla.

```
[ ] 1 word = input("Introduce una palabra: ")
     2 for i in range(10):
     3     print(word)
```

Introduce una palabra: Alf

Alf

Alf

Alf

Alf

Alf

Alf

Alf

Alf

Alf

Alf

Ejercicio Practico

Ejercicio

Escribir un programa que pregunte al usuario su edad y muestre por pantalla todos los años que ha cumplido (desde 1 hasta su edad).

```
[ ] 1 age = int(input("¿Cuántos años tienes? "))  
    2 for i in range(age):  
    3     print("Has cumplido " + str(i+1) + " años")
```

```
¿Cuántos años tienes? 8  
Has cumplido 1 años  
Has cumplido 2 años  
Has cumplido 3 años  
Has cumplido 4 años  
Has cumplido 5 años  
Has cumplido 6 años  
Has cumplido 7 años  
Has cumplido 8 años
```

Ejercicio Practico

Ejercicio

Escribir un programa que pida al usuario un número entero positivo y muestre por pantalla todos los números impares desde 1 hasta ese número separados por comas.

```
1 n = int(input("Introduce un número entero positivo: "))
2 for i in range(1, n+1, 2):
3     print(i, end=", ")
```

Introduce un número entero positivo: 9
1, 3, 5, 7, 9,

Ejercicio Practico

Ejercicio

Escribir un programa que muestre el eco de todo lo que el usuario introduzca hasta que el usuario escriba "salir" que terminará.

```
[ ] 1 while True:
    2     frase = input("Introduce algo: ")
    3     if frase == "salir":
    4         break
    5     print(frase)
```

```
Hola
Adios
Termina
Hola otra vez
```

```
[ ] 1
```

Contenido de la clase

Ejercicio Practico

Ejercicio

Escribir un programa que pida al usuario un número entero positivo mayor que 2 y muestre por pantalla si es un número primo o no.

```
1 n = int(input("Introduce un número entero positivo mayor que 2: "))
2 i = 2
3 while n % i != 0:
4     i += 1
5 if i == n:
6     print(str(n) + " es primo")
7 else:
8     print(str(n) + " no es primo")
```

Introduce un número entero positivo mayor que 2: 17
17 es primo

Los **números primos** son aquellos que solo, muy importante, solo son divisibles (al dividirse entre otro da un número entero) entre **ellos mismos y el 1**.

Por ejemplo: el 7.

Es un número primo porque solo es divisible por 7 y por 1.

Fuente:

<https://youtu.be/c11n-QDzTGk>

CERTUS

Ejercicio Propuesto 1

Escribí un programa para solicitar al usuario tres números y mostrar en pantalla al menor de los tres.

Primer número: *20*

Segundo número: *30*

Tercer número: *10*

Menor: 10

Ejercicio Propuesto 2

Escribí un programa que pregunte al usuario si desea analizar calificaciones de alumnos y, sólo si responde "S" comenzará el procesamiento de los datos, hasta que el usuario ingrese algo diferente de "S". Por cada alumno, permitir ingresar su calificación. Si es mayor a 4 el alumno está aprobado. Finalmente, mostrar "Porcentaje de alumnos aprobados: x %" (donde x es el porcentaje de aprobados sobre el total de calificaciones procesadas). También se debe imprimir "Promedio de los aprobados: y" (donde y es la calificación promedio, sólo de los alumnos aprobados).

¿Analizar calificaciones? 'S' para 'sí': *S*

Calificación de un alumno: *9*

¿Continuar? 'S' para 'sí': *S*

Calificación de un alumno: *4*

¿Continuar? 'S' para 'sí': *S*

Calificación de un alumno: *8*

¿Continuar? 'S' para 'sí': *N*

Porcentaje de alumnos aprobados: 66.66666666666667 %

Promedio de los aprobados: 8.5



Referencias bibliográficas

- <https://www.python.org/>



Gracias

