



SECRETARÍA DE INNOVACIÓN

Agenda

Sesión 8/18

1. Punto 1: ciclo while
2. Punto 2: ciclo for
3. Punto 3: función range
4. Punto 4: función enumerate
5. Punto 5: función break
6. Punto 6: función continue
7. Punto 7: asignación de valores if

PROGRAMACIÓN CON PYTHON

CICLO - WHILE

Ciclo - While

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

La sintaxis del bucle while es la siguiente:

```
while condicion:  
    cuerpo del bucle
```

Ciclo - While

La ejecución de esta estructura de control while es la siguiente:

- **Python evalúa la condición:**
 - ✓ **Si el resultado es True se ejecuta el cuerpo del bucle. Una vez ejecutado el cuerpo del bucle, se repite el proceso (se evalúa de nuevo la condición y, si es cierta, se ejecuta de nuevo el cuerpo del bucle) una y otra vez mientras la condición sea cierta.**
 - ✓ **Si el resultado es False, el cuerpo del bucle no se ejecuta y continúa la ejecución del resto del programa.**

La variable o las variables que aparezcan en la condición se suelen llamar variables de control. Las variables de control deben definirse antes del bucle while y modificarse en el bucle while.

Ciclo - While

¿Qué hace este ciclo?

Se puede ejecutar X cantidad de veces un bloque de código hasta que la condición se deje de cumplir.

```
i = 0
```

```
while i <= 3:
```

```
    print(i)
```

```
    i = i + 1 #contador utilizado para que el bucle
```

```
            #no sea infinito
```

```
print("Fin del bucle")
```

```
#####
```

```
i = 0
```

```
while i <= 3:
```

```
    print(i)
```

```
    i += 1
```

```
print("Fin del bucle")
```

Ciclo - While

else

El código se ejecutará cuando la condición ya no se cumpla.

```
i = 0
```

```
while i <= 50:
```

```
    print(i)
```

```
    i = 3 * i + 1
```

```
else:
```

```
    print('Ya no es menor o igual a 50')
```

```
#####
```

```
repetir = int(input("¿Cuántos números deseas imprimir?:"))
```

```
i = 1
```

```
while i <= repetir:
```

```
    print(i)
```

```
    i += 1
```

```
else:
```

```
    print('Fin del ciclo')
```

PROGRAMACIÓN CON PYTHON

CICLO - FOR

Ciclo - For

Un bucle for es un bucle que repite el bloque de instrucciones un número predeterminado de veces. El bloque de instrucciones que se repite se suele llamar cuerpo del bucle y cada repetición se suele llamar iteración.

La sintaxis del bucle for es la siguiente:

for variable **in** elemento iterable (lista, cadena, range, etc)
cuerpo del bucle

Ciclo - For

No es necesario definir la variable de control antes del bucle, aunque se puede utilizar como variable de control una variable ya definida en el programa.

El cuerpo del bucle se ejecuta tantas veces como elementos tenga el elemento recorrible.

```
numeros = [1,2,3,4,5,6,7,8,9,10]
```

```
for numero in numeros:
```

```
    print(numero) #imprime iteración por iteración
```

```
#####
```

```
print("Inicio del ciclo")
```

```
repetir = [1,2,3,4]
```

```
for x in repetir:
```

```
    print(x) #imprime el número de iteración
```

```
print("Fin del ciclo")
```

Ciclo - For

No es necesario definir la variable de control antes del bucle, aunque se puede utilizar como variable de control una variable ya definida en el programa.

El cuerpo del bucle se ejecuta tantas veces como elementos tenga el elemento recorrible.

```
print("Inicio del ciclo")
```

```
repetir = [1,2,3,4]
```

```
for x in repetir:
```

```
    print("Python 3") #imprime el string y no el número  
                        #de la iteración
```

```
print("Fin del ciclo")
```

```
#####
```

```
print("Inicio del ciclo")
```

```
repetir = (1,2,3,4) #tupla
```

```
for i in repetir:
```

```
    print("Python 3")
```

```
print("Fin del ciclo")
```

Ciclo - For

```
print("Inicio del ciclo")
```

```
dias = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"]
```

```
for dia in dias:
```

```
    print(dia)
```

```
print("Fin del ciclo")
```

```
#####
```

```
print("Inicio del ciclo")
```

```
for caracter in "Python 3": #Recorre un string
```

```
    print(caracter)
```

```
print("Fin del ciclo")
```

Ciclo - For

```
nombre = input("Digita tu nombre:")
```

```
for caracter in nombre:  
    print(caracter)
```

PROGRAMACIÓN CON PYTHON

Función Range y Enumerate

Función range

Podremos crear una secuencia de números la cual podremos iterar.

El tipo range es una lista inmutable de números enteros en sucesión aritmética.

En Python 2, range() se consideraba una función, pero en Python 3 no se considera una función, sino un tipo de datos , aunque se utiliza como si fuera una función.

```
for valor in range(10):
```

```
    print(valor)
```

```
#####
```

```
for valor in range(1,15):
```

```
    print(valor)
```

```
#####
```

```
for valor in range(-20,15):
```

```
    print(valor)
```

```
#####
```

```
for valor in range(1,101,2):
```

```
    print(valor)
```

Función range

```
lista=[1,2,3,4,5,6,7,8]
```

```
for valor in range( len(lista) ):  
    print(valor)
```

```
#####
```

```
lista=[1,2,3,4,5,6,7,8]
```

```
for valor in range( len(lista) ):  
    print("El indice:",valor, "el valor:",lista[valor])
```


Función range

```
repetir=int(input("¿Cuántas veces deseas repetir?"))  
for valor in range(repetir):  
    print(valor)
```

#####

```
caracter = input("Digita un caracter:")  
repetir=int(input("¿Cuántas veces deseas repetirlo?:"))  
for repeticion in range(repetir):  
    print(caracter)
```

Función enumerate

Toma como argumento un objeto iterable y retorna otro cuyos elementos son tuplas de dos objetos, el primero de los cuales indica la posición de un elemento y el segundo, el elemento mismo.

```
lenguajes = ["Python 2", "Python 3", "Java", "C", "C++"]  
valores = list(enumerate(lenguajes))  
print(valores)
```

```
#####
```

```
lista=[1,2,3,4,5,6,7,8]  
for elindice, elvalor in enumerate(lista):  
    print("El indice:",elindice, "el valor:",elvalor)
```

```
#####
```

```
lenguajes = ["Python 2", "Python 3", "Java", "C", "C++"]  
for i, lenguaje in enumerate(lenguajes):  
    print(i, lenguaje)
```

PROGRAMACIÓN CON PYTHON

Función Break y Continue

Función break

El control de bucle break se utiliza en bucles **for** y **while** y simplemente finaliza el bucle actual de forma abrupta y continua con la ejecución de la siguiente instrucción.

```
lenguajes = ["Python 2", "Python 3", "Java", "C", "C++"]
```

```
for lenguaje in lenguajes:
```

```
    if lenguaje == "Java":
```

```
        break
```

```
    print(lenguaje)
```

```
#####
```

```
numeros = [1,2,3,4,5,6,7,8,9,10]
```

```
for numero in numeros:
```

```
    if numero == 8:
```

```
        break
```

```
    print(numero)
```

Función break

```
print("Inicio del ciclo")
dias = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"]
for x in dias:
    print(x)
    if x == "Jueves":
        break
print("Fin del ciclo")

#####
print("Inicio del ciclo")
dias = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"]
for x in dias:
    if x == "Jueves":
        break
    print(x)
print("Fin del ciclo")
```

Función continue

El control de bucle **continue** hace que el ciclo “salte” a la siguiente iteración.

Al aparecer un **continue** en Python, este regresa al comienzo del bucle, ignorando Todas las instrucciones que quedan en la iteración actual del bucle e inicia la siguiente iteración.

```
lenguajes = ["Python 2", "Python 3", "Java", "C", "C++"]
```

```
for lenguaje in lenguajes:
```

```
    if lenguaje == "Java":
```

```
        continue
```

```
    print(lenguaje)
```

```
#####
```

```
numeros = [1,2,3,4,5,6,7,8,9,10]
```

```
for numero in numeros:
```

```
    if numero == 8:
```

```
        continue
```

```
    print(numero)
```

PROGRAMACIÓN CON PYTHON

Asignación de valores con if

Asignación de valores con if

Consiste en asignar un valor a un objeto, para este caso una variable, en base al cumplimiento o no de una condición.

Asignación de valores con if

Ejemplo

El usuario deberá digitar 3 calificaciones de una materia X.

Calcule el promedio de las 3 calificaciones.

Solo si el promedio supera el 5.99 entonces la materia es aprobada.

Imprima el promedio de las 3 calificaciones

Si la materia es aprobada imprima la palabra “Aprobada”

Si la materia es reprobada imprima la palabra “Reprobada”

Asignación de valores con if

Desarrollo

```
print("*****REGISTRO ACADÉMICO*****")  
cali1=float(input("Digite la primera calificación:"))  
cali2=float(input("Digite la segunda calificación:"))  
cali3=float(input("Digite la tercera calificación:"))
```

```
promedio = (cali1 + cali2 + cali3)/3
```

```
resultado = None
```

```
if promedio >= 6:
```

```
    resultado = "Aprobado"
```

```
else:
```

```
    resultado = "Reprobado"
```

```
print(promedio, resultado)
```

Asignación de valores con if

Reduciendo el código

```
print("*****REGISTRO ACADÉMICO*****")  
cali1=float(input("Digite la primera calificación:"))  
cali2=float(input("Digite la segunda calificación:"))  
cali3=float(input("Digite la tercera calificación:"))
```

```
promedio = (cali1 + cali2 + cali3)/3  
resultado = "Reprobado"
```

```
if promedio >= 6:  
    resultado = "Aprobado"
```

```
print(promedio, resultado)
```

Asignación de valores con if

Reduciendo el código aún más

```
print("*****REGISTRO ACADÉMICO*****")  
cali1=float(input("Digite la primera calificación:"))  
cali2=float(input("Digite la segunda calificación:"))  
cali3=float(input("Digite la tercera calificación:"))  
  
promedio = (cali1 + cali2 + cali3)/3  
resultado = "Aprobado" if promedio >= 6 else "Reprobado"  
  
print(promedio, resultado)
```

Importante

- **Ciclos en Python: For y While**
- **El for esta diseñado para recorrer elementos (for each en JAVA)**
- **Los ciclos do while y switch en Python no existen.**
- **Un switch en Python se puede sustituir mediante condiciones (elif)**
- **Un do while en Python se reemplaza agregando un else al while**

RESUMEN DE SESIÓN



SECRETARÍA DE
INNOVACIÓN