Estructuras de control iterativo

Los ciclos o bucles en Python. Uso declaración y sintaxis de ciclos en Python

Los ciclos, también conocidos como bucles o estructuras de control repetitivas, son de total importancia para el proceso de creación de un programa. Un ciclo en Python o bucle en Python (como prefieras llamarlos) te permite repetir una o varias instrucciones cuantas veces lo necesitemos, por ejemplo, si quisiéramos escribir los números del uno al cien no tendría sentido escribir cien líneas de código mostrando un número en cada una, para eso y para varias cosas más (que veremos enseguida), es útil un ciclo. Un ciclo nos ayuda a llevar a cabo una tarea repetitiva en una cantidad de líneas muy pequeña y de forma prácticamente automática (y muy rápida).

Existen diferentes tipos de ciclos o bucles en Python, cada uno tiene una utilidad para casos específicos y depende de nuestra habilidad y conocimientos poder determinar en qué momento es bueno usar alguno de ellos. Tenemos entonces a nuestra disposición los siguientes tipos de ciclos en Python:

- Ciclo `_while_`.
- Ciclo `_for_`.

Ciclo while (Mientras)

Los ciclos while son una estructura cíclica, que nos permite ejecutar una o varias líneas de código de manera repetitiva sin necesidad de tener un valor inicial e incluso a veces sin siquiera conocer cuando se va a dar el valor final que esperamos.

Con el ciclo while, no conoces el cuándo sino el cómo. Es decir, conocer la condición bajo la cual se va a detener el ciclo, pero no sabes cuántas iteraciones tomará eso, ni cuánto tiempo. Por ello se los llama ciclos indeterminados.

Un 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).

Ejemplo 1

`Imaginemos que, por algún motivo, queremos pedirle a un usuario una serie de números cualquiera y que solo dejaremos de hacerlo cuando el usuario ingrese el número 0.`

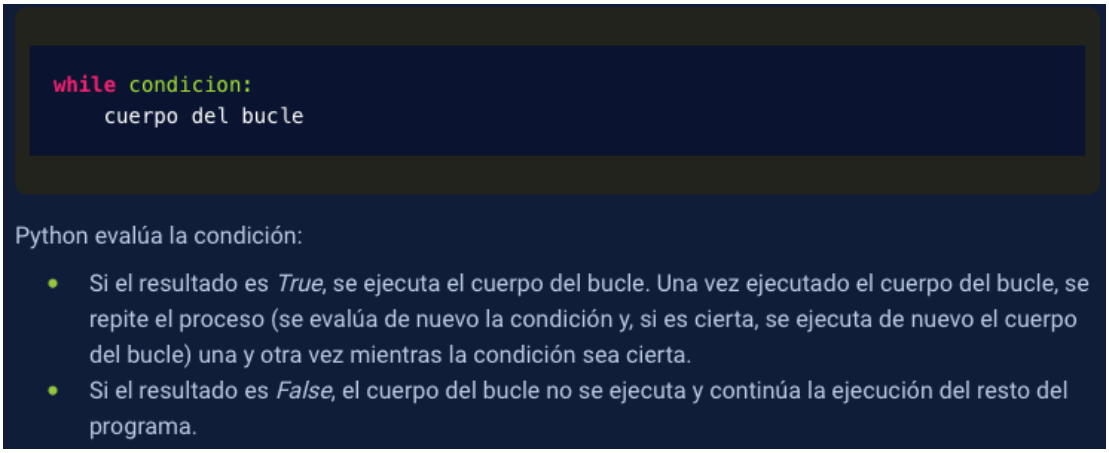
Ejemplo 2

`Si estamos haciendo en un juego de un bingo, el juego no va a parar hasta que halla un ganador.`

Ejemplo 3

`Imaginemos que estamos rifando un obsequio entre varios amigos, si en la primera ronda no hay un ganador, se seguira rifando y haciendo varias rondas, hasta que haya un ganador`

La sintaxis del bucle while es la siguiente:



Algoritmo para hacer amigos

In [ ]:
```
# El presente bloque es un ciclo while infinito. CUIDADO: puede crashear el sist
# Para usarlo, borra las comillas.
"""
suma = 0

# While - Mientras
while True:
  suma += 1 # suma = suma + 1
  print('Ciclo número:', suma)
"""
```

In [3]:
```
# Ejercicio 2
estudiantes = 4
preguntas = 0

while (preguntas < estudiantes): # preguntas = 5 es menor e igual a 5 = TRUE
    preguntas += 1 # contador
    # preguntas = 6
    print("El profe realiza la pregunta: ", preguntas)  # Pregunta N 6


print("¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨")
print("ooooooooooooooooooo")
print("El programa Finalizo")
```

```
El profe realiza la pregunta:  1
El profe realiza la pregunta:  2
El profe realiza la pregunta:  3
El profe realiza la pregunta:  4
……………………
oooooooooooooooooooo
El programa Finalizo
```

In [9]:
```python
# Ejercicio 3
contador = 0
tecla_pulsada = input("Oprime una letra :\n")

while tecla_pulsada != "p":
    print('Oprimiste la tecla ..."',tecla_pulsada,'"... Por favor intenta nuevam
    contador += 1
    tecla_pulsada = input("Oprime de nuevo una letra: \n")

print("Has orpimido la tecla perfecta..., Gracias")
print(contador)
```

```
Oprimiste la tecla ..." u "... Por favor intenta nuevamente
Oprimiste la tecla ..." j "... Por favor intenta nuevamente
Oprimiste la tecla ..." t "... Por favor intenta nuevamente
Oprimiste la tecla ..." r "... Por favor intenta nuevamente
Has orpimido la tecla perfecta..., Gracias
4
```

In [ ]:
```python
# Ejercicio 3.1
# Imprime la cantidad de notas y el promedio e estas

cantidad_notas = int(input("Ingrese la cantidad de notas para saber su promedio:

contador = 1
Suma = 0

while (contador <= cantidad_notas):
    print("Ingrese su nota # ", contador)
    nota = float(input())
    Suma += nota
    contador += 1

promedio = Suma/cantidad_notas
print("Su promedio es: ", promedio)
```

In [ ]:
```python
# Ejercicio 4
print("""
**********************************************************
==========================================================
        HOLA, FASE ESPECIAL DE VACUNACIÓN

Hola, estás ingresando a la aplicación de vacunación.
Estamos vacunando de acuerdo a la prioridad que se
encuentra en las fases.

==========================================================
**********************************************************
""")
edad_maxima = 60
edad_minima = 30
```

```
print("Primera persona")
edad = int(input("Ingresa tu edad :\n")) #

#                  (  False          and     TRUE) FALSE) TRUE
while not(( edad >= edad_minima) and ( edad <= edad_maxima )):
    print('Tu edad no está dentro de la fase de vacunación \nMantente pendiente
    print("Siguiente persona")
    edad = int(input("Ingresa tu edad :\n"))

print("FELICITACIONES, estás dentro de la fase de vacunación...")
```

In [ ]:
```
# Ejercicio 5
print("""
**********************************************************
==========================================================
        HOLA, FASE ESPECIAL DE VACUNACIÓN

Hola, estás ingresando a la aplicación de vacunación.
Estamos vacunando de acuerdo a la prioridad que se
encuentra en las fases.

==========================================================
**********************************************************
""")
edad_maxima = 60
edad_minima = 30

edad = int(input("Ingresa tu edad :\n")) #

#               (FALSE         OR       FALSE) : FALSE
while ( edad < edad_minima) or ( edad > edad_maxima ):
    print('Tu edad no está dentro de la fase de vacunación \nMantente pendiente
    edad = int(input("Ingresa tu edad :\n"))

print("""
----------------------------------------------------------
----------------------------------------------------------
FELICITACIONES, estás dentro de la fase de vacunación...
----------------------------------------------------------
----------------------------------------------------------
""")
```

Ejercicio 5 - un loop while evitando un bucle infinito

Imaginemos que estamos pidiendo a algún amigo que acierte el número. Después de 3 intentos, el juego finaliza asi no hayas acertado

In [ ]:
```
print("Adivina el número ")
num = int(input("Ingresa un número por favor: \n"))

intentos = 1
num_suerte = 7

while ( num != num_suerte ):
    print('Realiza de nuevo el intento, escoge otro numero')
```

```
    if (intentos == 3):
        print('Se te agotaron la cantidad de intentos. \nGracias por participar.
        break       #   Sale del bucle

    print("Intento # " + str(intentos))
    num = int(input("Ingresa un número por favor: \n"))
    if ( num != num_suerte ):
        intentos += 1

# --- FINALIZA EL LOOP

if (intentos <= 3) and (num == num_suerte):
    print('Has ingresado el numero correcto, en el intento #',intentos+1)
```

||| # Taller de ejercicios While

- `1.25 puntos -` Elaborar un programa que solicite al usuario que ingrese números enteros positivos y, por cada uno, imprimir la suma de los dígitos que lo componen. La condición para que salga del bucle es que se ingrese el número -1. Al finalizar, mostrar cuántos de los números ingresados por el usuario fueron números pares.

- `1.25 puntos -` Diseñar un programa que muestre un menú con tres opciones: 1- comenzar programa, 2- imprimir listado (compañeros de proyecto integrador), 3- finalizar programa. El usuario debe poder seleccionar una opción (1, 2 ó 3). Si elige una opción incorrecta, informarle del error. El menú se debe volver a mostrar luego de ejecutada cada opción, permitiendo volver a elegir. Si elige las opciones 1 ó 2 se imprimirá un texto. Si elige la opción 3, se interrumpirá la impresión del menú y el programa finalizará.

- `1.25 puntos -` Diseñe un programa que permita al usuario ingresar los montos de las compras de un cliente. El programa saldra del loop cuando el ingreso de datos sea un monto 0.

Si ingresa un monto negativo, no se debe procesar y se debe pedir que ingrese un nuevo monto. Al finalizar, informar el total a pagar teniendo que cuenta que, si las ventas superan el total de $1000, se le debe aplicar un 10% de descuento.

- `1.25 puntos -` Diseñar un programa que solicite el ingreso de números enteros positivos, hasta que el usuario ingrese el 0.

Al finalizar, el programa informará la cantidad de dígitos pares y de dígitos impares ingresados.

Bucle

Bucle “while” controlado por Conteo

In [ ]:
```
numero = 0
suma = 0
while (numero <= 2):
```

```
    suma = numero + suma  # suma += numero
    numero = numero + 1 # numero + = 1
    print("Numero = ",numero, "\nSuma = :", suma)
    print("-------------")
print ("La suma es " + str(suma))
```

Bucle “while” controlado por Evento

In [ ]:
```
print ("Introduzca la nota de un estudiante (-1 para salir): ")
grado = int(input())
total = 0
contar = 0
while grado != -1:
    total = total + grado
    contar = contar + 1
    print ("Introduzca la nota de un estudiante (-1 para salir): ")
    grado = int(input())
promedio = total / contar
print("La cantidad de notas ingresadas fueron: ", contar)
print ("Promedio de notas del grado escolar es: " + str(promedio))
```

Bucle “while” con “else”

In [ ]:
```
grado = int(input("Introduzca la nota de un estudiante (-1 para salir): "))
total = 0
contar = 0

while grado != -1:
    total = total + grado
    contar += 1
    grado = int(input("Introduzca la nota de un estudiante (-1 para salir): "))
else:
    promedio = total / contar
    print ("Promedio de notas del grado escolar: " + str(promedio))
```

Bucle While con sentencia break

In [ ]:
```
variable = int(input("Ingrese un número:\n"))

while variable > 0:
    print ("Actual valor de variable:", variable)
    variable = variable -1 #
    if variable == 5:
        break
```

Bucle While con sentencia continue

In [ ]:
```
variable = int(input("Ingrese un número:\n")) # 6

while variable > 0:
   variable = variable -1
   if variable == 5:
```

```
        continue
    print ("Actual valor de variable:", variable)
```

In [ ]: