# Ciclos Repetitivos

Jhon Jairo Padilla Aguilar, PhD.

# Necesidad

- Muchas veces necesitamos que nuestro programa haga una tarea de manera repetitiva muchas veces
- Si tuviésemos que repetir 3 instrucciones 100 veces, requeriríamos 300 líneas de código
- El código se puede optimizar usando ciclos repetitivos

# Ciclos Repetitivos

- Son estructuras de programación que permiten repetir muchas veces un mismo conjunto de instrucciones
- Requieren que se cumpla una condición para mantenerse dentro del ciclo.  Si se deja de cumplir, el computador se sale del ciclo y continúa la siguiente instrucción posterior a este.
- Hay varios tipos de ciclos: Mientras (While), Repetir hasta (Repeat-until), Para (For)

# Ciclo While

```
n = 5
while n > 0:
    print (n)
    n = n-1
print (“¡Despegue!”)
```

# Tips para usar ciclos

- Suelen usarse para ir llevando cuentas acumuladas como sumatorias, productos, etc.
- Variables típicas:
  - **Acumulador** (Lleva la suma o producto acumulado):
  - Ej: suma=suma+dato
- Muchas veces las operaciones del ciclo se hacen por un número limitado de veces
- Variables típicas:
  - Es útil llevar la cuenta de cuántas veces se ha realizado mediante una variable que lleva este conteo (denominada **contador**)
  - Ej: contador=contador+1

# La condición del ciclo

- Puede evaluar la variable contador, si necesitamos que se haga un número de veces fijo
- Ej:

```
contador=1
Mientras contador<5:
    imprimir (contador)
    contador=contador+1
```

- Puede evaluar una variable que sea una bandera que indique el momento de salida del ciclo:
- Ej:

```
leer (letra)
Mientras letra!='s':
    imprimir (letra)
    leer (letra)
```

# Ciclos infinitos y la instrucción break

```
n = 10
while True:
    print (n)
    n = n - 1
print (“¡Terminado!”)
```

- El bucle es un bucle infinito, porque la expresión lógica de la sentencia while es simplemente la constante lógica True (verdadero)
- **PRECAUCIÓN: No lo ejecute!!!...se le bloqueará el computador!!!**

# Ejemplo Break

```
while True:
    linea = str(input('> '))
    if linea == 'fin':
        break
    print (linea)
print (“¡Terminado!”)
```

- La condición del bucle es True, lo cual es verdadero siempre, así que el bucle se repetirá hasta que se ejecute la sentencia break.
- Cada vez que se entre en el bucle, se pedirá una entrada al usuario. Si el usuario escribe fin, la sentencia break hará que se salga del bucle. En cualquier otro caso, el programa repetirá cualquier cosa que el usuario escriba y volverá al principio del bucle.

# Ejemplo: Break

- Este  es un ejemplo de su funcionamiento:

  > hola a todos

  hola a todos

  > he terminado

  he terminado

  > fin

  ¡Terminado!

# Instrucción continue

- Algunas veces, estando dentro de un bucle se necesita terminar con la iteración actual y saltar a la siguiente de forma inmediata.
- En ese caso se puede utilizar la sentencia *continue* para pasar a la siguiente iteración sin terminar la ejecución del cuerpo del bucle para la actual.

# Ejemplo continue

```
while True:
    linea = str(input('> '))
    if linea[0] == '#' :
        continue
    if linea == 'fin':
        break
    print (linea)
print (“¡Terminado!”)
```

- Bucle que repite lo que recibe como entrada hasta que el usuario escribe “fin”
- Pero trata las líneas que empiezan por el carácter almohadilla como líneas que no deben mostrarse en pantalla (algo parecido a lo que hace Python con los comentarios).

# Ejemplo continue

> hola a todos

hola a todos

> # no imprimas esto

> ¡imprime esto!

¡imprime esto!

> fin

¡Terminado!