Estructura de Control Repetitiva `while`

Computer Science

CS1100 - Introducción a Ciencia de la Computación

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Mark Zuckerberg - Chief Executive Officer of Facebook

https://www.youtube.com/watch?v=hYvcoRkAkOU

Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Diseñar e implementar algoritmos en **Python** utilizando `while` como estructura de control repetitiva.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Logro de la Sesión

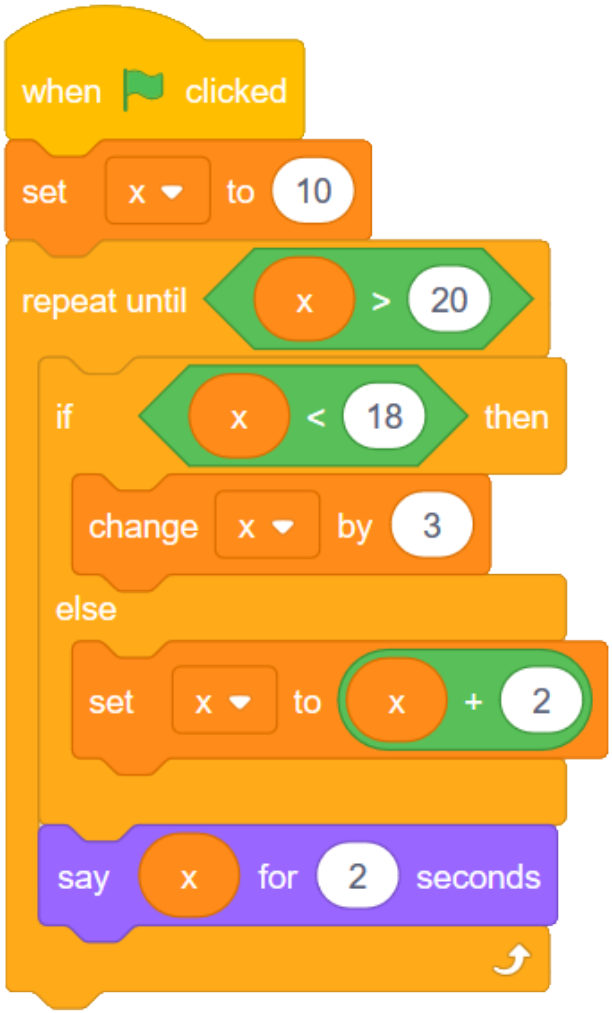
Al finalizar esta sesión, estarás en la capacidad de:

- Diseñar e implementar algoritmos en **Python** utilizando **`while`** como estructura de control repetitiva.
- Diseñar e implementar algoritmos utilizando **`break`** en la estructura de control repetitiva **`while`**.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

`repeat until` en Python es `while`

repeat until



Ejemplo de repeat until

Definicíon

¿Qué es una iteración?

- Una iteración es ejecutar el mismo bloque de código una y otra vez, muchas veces.

¿Qué es while?

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Definicíon

¿Qué es una iteración?

- Una iteración es ejecutar el mismo bloque de código una y otra vez, muchas veces.
- Una estructura de control repetitiva que implementa la iteración se llama **bucle** (Español) o **loop** (Ingles).

¿Qué es while?

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Definicíon

¿Qué es una iteración?

- Una iteración es ejecutar el mismo bloque de código una y otra vez, muchas veces.
- Una estructura de control repetitiva que implementa la iteración se llama **bucle** (Español) o **loop** (Ingles).

¿Qué es while?

- La sentencia `while` es una Estructura de control repetitiva.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Definicíon

¿Qué es una iteración?

- Una iteración es ejecutar el mismo bloque de código una y otra vez, muchas veces.
- Una estructura de control repetitiva que implementa la iteración se llama **bucle** (Español) o **loop** (Ingles).

¿Qué es while?

- La sentencia **`while`** es una Estructura de control repetitiva.
- Al igual que una sentencia **`if`** , la sentencia **`while`** utiliza una **condicional**.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Definicíon

¿Qué es una iteración?

- Una iteración es ejecutar el mismo bloque de código una y otra vez, muchas veces.
- Una estructura de control repetitiva que implementa la iteración se llama **bucle** (Español) o **loop** (Ingles).

¿Qué es while?

- La sentencia **`while`** es una Estructura de control repetitiva.
- Al igual que una sentencia **`if`** , la sentencia **`while`** utiliza una **condicional**.
- El bloque de instrucciones de **`while`** se ejecutará mientras la **condicional** se evalúe como **True**.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Definicíon

¿Qué es while?

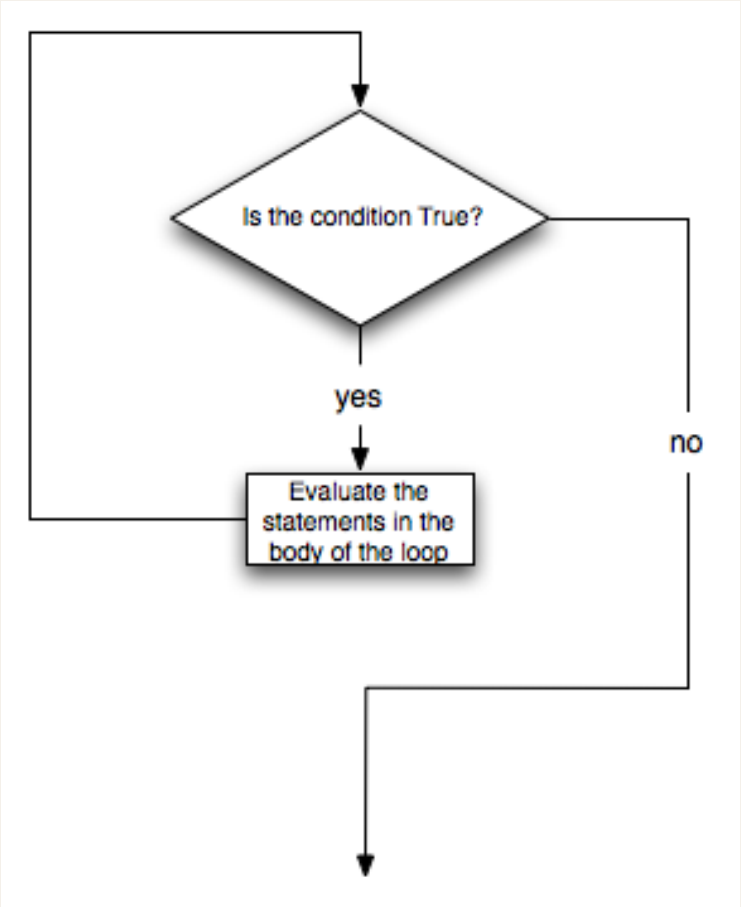
- El formato de un bucle `while` es:

```
1   while <condicion>:
2       <instrucción 1>
3       <instrucción 2>
4       ...
5       <instrucción n>
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Definicíon

Flujo de control de while



Ejemplo 1: `while`

Diseñe e implemente un algoritmo que imprima los números del 1 al 5

```
1   1
2   2
3   3
4   4
5   5
```

Ejemplo 1: `while`

Diseñe e implemente un algoritmo que imprima los números del 1 al 5

```
1
2
3
4
5
```

Algoritmo:

```
i = 1
while i <= 5:
    print(i)
    i = i + 1
```

Ejemplo 2: `while`

¿Qué hace y que imprime el siguiente algoritmo?

```
1   i = 5
2   while i <= 35:
3       print(i)
4       i = i + 5
```

Ejemplo 2: `while`

¿Qué hace y que imprime el siguiente algoritmo?

```
1  i = 5
2  while i <= 35:
3      print(i)
4      i = i + 5
```

El resultado de la ejecución es lo siguiente:

```
1  5
2  10
3  15
4  20
5  25
6  30
7  35
```

Definicíon

Interrupción de la iteración de un bucle

- En los dos ejemplos que hemos visto hasta ahora, todas las **intrucciones** del bucle `while` se ejecutan en cada iteración.
- Python proporciona la palabra reservada `break` que termina un bucle por completo, y salta a la siguiente instrucción que sigue al bucle.

```
1  while <condicion>:
2      <instrucción 1>
3      ...
4      break
5      ...
6      <instrucción n>
```

Definicíon

Interrupción de la iteración de un bucle

- En los dos ejemplos que hemos visto hasta ahora, todas las **intrucciones** del bucle **`while`** se ejecutan en cada iteración.
- Python proporciona la palabra reservada **`break`** que termina un bucle por completo, y salta a la siguiente instrucción que sigue al bucle.

```
1  while <condicion>:
2      <instrucción 1>
3      ...
4      break
5      ...
6      <instrucción n>
```

Ejemplo 3: `break`

Diseñe e implemente un algoritmo que imprima los números pares del 8 al 0, pero si se detecta el número 4 acabe el bucle y muestre el mensaje `while terminado`

```
1       8
2       6
3       while terminado
```

Ejemplo 3: `break`

Diseñe e implemente un algoritmo que imprima los números pares del 8 al 0, pero si se detecta el número 4 acabe el bucle y muestre el mensaje `while terminado`

```
8
6
while terminado
```

Algoritmo:

```
i = 10
while i >= 0:
    i = i - 2
    if i == 4:
        break
    print(i)
print("while terminado")
```

Ejercicio 1: Imprime números pares

Diseñe e implemente un algoritmo que permita al usuario ingresar un número, y el algoritmo debe imprimir los números pares, desde cero hasta el número ingresado.

input:

```
12
```

output:

```
0
2
4
6
8
10
12
```

Ejercicio 2: Promedio de un conjunto de números

Diseñe e implemente un algoritmo que obtenga el promedio de un conjunto de números ingresados por el usuario. El usuario ingresará cero para indicar que ya no ingresará más números. El cero no se considera en el promedio.

input:

```
1
6
2
14
0
```

output:

```
5.75
```

UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Ejercicio 3: Secuencia Simple

Dada la secuencia: 1, 4, 9, 16, 25, 36, ... Diseñe e implemente un algoritmo que permita al usuario ingresar la cantidad de números a mostrar, y el algoritmo debe imprimir la la secuencia: 1, 4, 9, 16, 25, 36, ..., hasta el cuadrado del número **n**.

input:

```
1   4
```

output:

```
1   1
2   4
3   9
4   16
```

Ejercicio 4: Tabla de multiplicar

Desarrolle un programa que imprima la tabla de multiplar de un número **n** ingresado por teclado.

input:

```
1   5
```

output:

```
1   5x1=5
2   5x2=10
3   5x3=15
4   5x4=20
5   5x5=25
6   5x6=30
7   5x7=35
8   5x8=40
9   5x9=45
```

Cierre

En esta sesión aprendiste:

- Desarrollar programas utilizando `while`
- Desarrollar programas utilizando `break` y `continue`

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Cierre

En esta sesión aprendiste:

- Desarrollar programas utilizando **`while`**
- Desarrollar programas utilizando **`break`** y **`continue`**

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA