



Universidad Metropolitana  
para la Educación y el Trabajo

# Control de flujo en Python

Lic. Diego Krauthamer



## Agenda

- Estructuras condicionales
- Estructuras de repetición
- Preguntas de revisión conceptual
- Bibliografía

## Control de Flujo – Sentencias Condicionales/1

- ✓ Las instrucciones condicionales nos permiten comprobar condiciones y que nuestro programa se comporte de una manera o de otra, y que ejecute un fragmento de código u otro, dependiendo de la condición.

```
#Sintaxis Sentencias condicionales  
Si (condición) Entonces  
    <Acciones si verdadero>  
<Sino>  
    <Acciones si falso>
```

**Figura – Sintaxis SI Pseudocódigo**

## Control de Flujo – Sentencias Condicionales/2

- ✓ A manera de ejemplo desarrollar un programa que determine si un número es negativo

```
#Programa positivo_negativo  
  
#Variables  
numero=5  
#Dertermino si el nro es positivo o negativo  
Si (numero >= 0) Entonces  
    Mostrar("Es positivo")  
Sino  
    Mostrar("Es negativo")
```

**Figura** – Pseudocódigo que evalúa si un número es positivo o negativo

## Control de Flujo – Sentencias Condicionales/3

- ✓ Las instrucciones condicionales nos permiten comprobar condiciones y que nuestro programa se comporte de una manera o de otra, y que ejecute un fragmento de código u otro, dependiendo de la condición.

```
#Sintaxis if Python  
if <condición>:  
    <Acciones si verdadero>  
<else:>  
    <Acciones si falso>
```

**Figura – Sintaxis if en python**

## Control de Flujo – Sentencias Condicionales/4

- ✓ A manera de ejemplo desarrollar un programa que determine si un número es negativo

```
#Programa positivo_negativo  
  
#Variables  
numero=5  
#Determino si el nro. Es positivo o  
negativo  
if numero >= 0:  
    print("Es positivo")  
else:  
    print("Es negativo")
```

**Figura** – código python que evalúa si un número es positivo o negativo

## Control de Flujo – Sentencias Condicionales/5

- ✓ Las instrucciones condicionales nos permiten comprobar condiciones y que nuestro programa se comporte de una manera o de otra, y que ejecute un fragmento de código u otro, dependiendo de la condición.

```
#Sintaxis Sentencias condicionales  
Si (condición) Entonces  
    <Acciones si verdadero>  
<sino si>  
    <Acciones>  
    ...  
<Sino>  
    <Acciones si falso>
```

## Control de Flujo – Sentencias Condicionales/6

- ✓ Versión 2.0 del programa para determinar si un numero es positivo o negativo

```
#Variables  
numero=0  
  
#Dertermino si el nro es positivo o negativo  
Si (numero < 0) Entonces  
    Mostrar("Es negativo")  
Sino si numero > 0 Entonces  
    Mostrar("Es positivo")  
Sino  
    Mostrar("Es cero")
```

**Figura** – Pseudocódigo que evalua si un número es positivo  
o negativo



## Control de Flujo – Sentencias Condicionales/7

```
#Sintaxis if Python
if <condición1>:
    <Acciones si condición 1 es verdadera>
<elif condicion2:>
    <Acciones si condición 2 es verdadera>
<elif condicion3:>
    <Acciones si condición 3 es verdadera>
<else:>
    <Acciones si condiciones 1,2 o 3 son falsas>
```

**Figura** – sintaxis Python del condicional

## Control de Flujo – Sentencias Condicionales/8

✓ Versión 2.0 del código Python

```
#Programa positivo_negativo  
#Variables  
numero=5  
#Determino si el nro. Es positivo o  
negativo  
if numero < 0:  
    print("Es negativo")  
elif numero >0:  
    print("Es positivo")  
else:  
    print("Es cero")
```

**Figura** – código python que evalúa si un número es positivo o negativo

## Estructuras de repetición - Bucle Mientras/1

- ✓ Las estructuras de repetición nos permiten ejecutar un mismo fragmento de código un cierto número de veces, mientras se cumpla una determinada condición.
- ✓ La estructura de repetición "Mientras" ejecuta un fragmento de código un cierto número de veces mientras se cumpla una determinada condición.

```
#Sintaxis Bucle Mientras  
Mientras (condición)  
    <Acciones si condición es verdadera>  
Fin Mientras
```

**Figura** – Sintaxis bucle Mientras

## Estructuras de repetición - Bucle Mientras/2

- ✓ Codificar un programa que valide que un número sea positivo

```
#Declaración de variables  
numero=-1  
  
#Ingreso de datos  
Mientras (numero < 0)  
    mostrar("Ingrese un numero positivo")  
    leer(numero)  
    Si (numero <0) Entonces  
        mostrar("Ingrese un número positivo")  
    Fin si  
Fin Mientras
```

**Figura** – Sintaxis bucle Mientras

## Estructuras de repetición - Sintaxis bucle While en Python

**#Sintaxis Bucle Mientras**

**while <condición>:**

    <Acciones si condición es verdadera>

**Figura** – Sintaxis bucle Mientras en Python

## Estructuras de repetición - Bucle Mientras/3

- ✓ Codificar un programa que valide que un número sea positivo

```
#Declaración de variables  
numero=-1  
  
#Ingreso de datos  
while numero < 0:  
    print("Ingresa un numero positivo")  
    numero=int(input())  
    #Mensaje de error al usuario  
    if numero <0:  
        print("Ingresa un número positivo")
```

**Figura** – Ejemplo estructura de repetición While

## Bibliografía obligatoria

- Python para todos. Raúl Gonzalez Duque. Capítulo Control de flujo – páginas 30 – 31.
- Guía de introducción a la programación Unidad 7