



Universidad Metropolitana
para la Educación y el Trabajo

Expresiones y operadores en Python

Lic. Diego Krauthamer

Agenda

- Variables, constantes, operadores y expresiones
- Operadores aritméticos
- Operadores de comparación
- Operadores Lógicos
- Tablas de verdad
- Sentencias condicionales
- Preguntas de revisión conceptual
- Bibliografía

Expresiones y operaciones aritméticas

- ✓ Las **variables y constantes** se pueden procesar utilizando operaciones y funciones adecuadas a sus tipos
- ✓ Una **expresión** es un conjunto de datos o funciones unidas por operadores aritméticos.

Operadores aritméticos

- ✓ Los operadores aritméticos pueden ser utilizados con tipos enteros o reales.

Operador	Significado	Ejemplo	Resultado
+	Suma	num1 + num2	Suma de num1 y num2
-	Resta	num1 - num2	Resta de num1 y num2
*	Multiplicación	num1*num2	Producto de num1 y num2
/	División	num1/num2	Cociente de num1 y num2
**	Potencia	num1 ** num2	Potencia de num 1 elevado a num2
%	Módulo	Num1 % num2	Resto de num1 por num2

Figura – Operadores aritméticos

Operadores de comparación

- ✓ Los operadores de comparación o relacionales se utilizan para expresar condiciones y describen una relación entre dos valores.

Operador	Significado	Ejemplo	Resultado
==	Igual	num1 + num2	Suma de num1 y num2
!=	Distinto	num1 - num2	Resta de num1 y num2
>	Mayor	num1*num2	Producto de num1 y num2
<	Menor	num1/num2	Cociente de num1 y num2
>=	Mayor o Igual que	num1 ** num2	Potencia de num 1 elevado a num2
<=	Menor o igual que	Num1 % num2	Resto de num1 por num2

Figura – Operadores de comparación

Operadores lógicos

- ✓ Los operadores lógicos se utilizan con expresiones lógicas para combinarlas y obtener expresiones más complejas..

Operador	Significado	Ejemplo
AND	Conjunción	Expresión1 AND Expresión2
OR	Disyunción	Expresión1 OR Expresión2
NOT	Negación	NOT(Expresión1)

Figura – Operadores Lógicos

Tablas de verdad/1 – operadores AND y OR

- ✓ Las tablas de verdad describen fácilmente el resultado de las expresiones lógicas.

Expresión1	Expresión2	Resultado
VERDADERO	VERDADERO	VERDADERO
VERDADERO	FALSO	FALSO
FALSO	VERDADERO	FALSO
FALSO	FALSO	FALSO

Figura – Tabla de Verdad Operador AND

Expresión1	Expresión2	Resultado
VERDADERO	VERDADERO	VERDADERO
VERDADERO	FALSO	VERDADERO
FALSO	VERDADERO	VERDADERO
FALSO	FALSO	FALSO

Figura – Tabla de Verdad Operador OR

Tablas de verdad/2 – operador NOT (negación)

Expresión1	Resultado
VERDADERO	FALSO
FALSO	VERDADERO

Figura – Tabla de Verdad Operador NOT

Tablas de verdad/2 – operador NOT (negación)

Expresión1	Resultado
VERDADERO	FALSO
FALSO	VERDADERO

Figura – Tabla de Verdad Operador NOT

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

✓ 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:  
    prnt("Es cero")
```

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

Bibliografía obligatoria

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