

Operadores y expresiones en Python (1 hora)

Objetivos:

1. Entender los diferentes tipos de operadores en Python: aritméticos, de comparación y lógicos.
2. Aprender a utilizar estos operadores en expresiones.
3. Practicar la creación de ejemplos de código que utilicen estos operadores.

Contenido

I. Introducción (5 minutos)

- Explicar la importancia de los operadores y expresiones en la programación.
- Presentar los tres tipos de operadores que se estudiarán en la clase.

II. Operadores aritméticos (20 minutos)

- Suma (+)
- Resta (-)
- Multiplicación (*)
- División (/)
- Módulo (%)
- Exponenciación (**)
- División entera (//)

Ejemplo de código:

```
1  a = 7
2  b = 3
3
4  suma = a + b
5  resta = a - b
6  multiplicacion = a * b
7  division = a / b
8  modulo = a % b
9  exponenciacion = a ** b
10 division_entera = a // b
11
12 print("Suma:", suma)
13 print("Resta:", resta)
14 print("Multiplicación:", multiplicacion)
15 print("División:", division)
16 print("Módulo:", modulo)
```

```
17 print("Exponenciación:", exponenciacion)
18 print("División entera:", division_entera)
```

III. Operadores de comparación (15 minutos)

- Igual (==)
- Diferente (!=)
- Mayor que (>)
- Menor que (<)
- Mayor o igual que (>=)
- Menor o igual que (<=)

Ejemplo de código:

```
1 x = 5
2 y = 8
3
4 igual = x == y
5 diferente = x != y
6 mayor_que = x > y
7 menor_que = x < y
8 mayor_o_igual = x >= y
9 menor_o_igual = x <= y
10
11 print("Igual:", igual)
12 print("Diferente:", diferente)
13 print("Mayor que:", mayor_que)
14 print("Menor que:", menor_que)
15 print("Mayor o igual que:", mayor_o_igual)
16 print("Menor o igual que:", menor_o_igual)
```

IV. Operadores lógicos (20 minutos)

- AND (and)
- OR (or)
- NOT (not)

Ejemplo de código:

```
1 verdadero = True
2 falso = False
3
4 conjuncion = verdadero and falso
5 disyuncion = verdadero or falso
6 negacion = not verdadero
7
8 print("AND:", conjuncion)
9 print("OR:", disyuncion)
10 print("NOT:", negacion)
```

V. Conclusión (5 minutos)

- Resumir los conceptos aprendidos en la clase.
- Recordar la importancia de practicar con ejercicios y proyectos para mejorar el dominio de los operadores y expresiones en Python.

VI. Ejercicios adicionales (opcional)

- Crear ejercicios prácticos que utilicen operadores aritméticos, de comparación y lógicos para resolver problemas y situaciones reales.