

Operadores de comparación

¿Cuáles son los operadores de comparación?

Los operadores de comparación evalúan la relación que existe entre dos valores en Python. Existen diferentes tipos de operadores de comparación.

Operador	Ejemplo	Significado
<code>==</code>	<code>a == b</code>	Igual a
<code>!=</code>	<code>a != b</code>	No igual a
<code><</code>	<code>a < b</code>	Menor que
<code><=</code>	<code>a <= b</code>	Menor que o igual a
<code>></code>	<code>a > b</code>	Mayor que
<code>>=</code>	<code>a >= b</code>	Mayor que o igual a

1. Igualdad y desigualdad

Los operadores de comparación más utilizados son el de igualdad (`==`) y el de desigualdad (`!=`). Estos operadores pueden aplicarse a varios tipos de datos en Python.

1.1. Tipos de datos numéricos

```
num1 = 2
num2 = 5
```

```
num1 == 2
```

```
True
```

```
num1 == num2
```

```
False
```

```
num1 == 2.0
```

```
True
```

```
num1 != 2
```

```
False
```

```
num1 != num2
```

```
True
```

1.2. Strings

```
text1 = "cadena de texto"
text2 = "cadena de texto 2"
```

```
text1 == "cadena de texto"
```

```
True
```

```
text1 == text2
```

```
False
```

```
text1 != text2
```

```
True
```

2. Otros operadores de comparación

✓ 2.1. Tipos de datos numéricos

```
num1 = 2  
num2 = 5
```

```
num1 < num2
```

```
True
```

```
num2 < num1
```

```
False
```

```
num1 <= 2.0
```

```
True
```

✓ 2.2. Strings

```
text1 = "Cadena de texto"
```

```
text1 < "texto"
```

```
True
```

La comparación utiliza un orden lexicográfico: primero se comparan los dos primeros elementos, y si son diferentes, esto determina el resultado de la comparación; si son iguales, se comparan los dos siguientes elementos, y así sucesivamente, hasta que se agote cualquiera de las dos secuencias.

Esto quiere decir que la comparación se realiza utilizando los equivalentes numéricos (el resultado de la función por defecto `ord()`) de sus caracteres.

```
help(ord)
```

```
Help on built-in function ord in module builtins:
```

```
ord(character, /)
```

```
Return the ordinal value of a character.
```

```
If the argument is a one-character string, return the Unicode code  
point of that character.
```

```
If the argument is a bytes or bytearray object of length 1, return its  
single byte value.
```

```
ord('a')
```

```
97
```

```
"Cadena" < "Texto"
```

```
True
```

```
ord('C')
```

```
67
```

```
ord('T')
```

```
84
```

```
help(chr)
```

```
Help on built-in function chr in module builtins:
```

```
chr(i, /)
```

```
Return a Unicode string of one character with ordinal i; 0 <= i <= 0x10ffff.
```

```
chr(67)
```

```
'C'
```

```
chr(84)
```

```
'T'
```

```
help(print)
```

Help on built-in function print in module builtins:

```
print(*args, sep=' ', end='\n', file=None, flush=False)
  Prints the values to a stream, or to sys.stdout by default.
```

```
    sep
      string inserted between values, default a space.
    end
      string appended after the last value, default a newline.
    file
      a file-like object (stream); defaults to the current sys.stdout.
    flush
      whether to forcibly flush the stream.
```