Comparison Operator

Covered In - Depth

Table of contents

01

Quick Recap

Quick Recap

==

Equality

Purpose: Checks if the value on the left side is *equal* to the value on the right side.

Syntax: a == b

Returns: True if a is equal to b, otherwise False.

!=

Inequality

Purpose: Checks if the value on the left side is *not equal* to the value on the right side.

Syntax: a != b

Returns: True if a is not equal to b, otherwise False.

Quick Recap

>

Greater Than

Purpose: Checks if the value on the left side is *greater than* the value on the right side.

Syntax: a > b

Returns: True if a is greater than b, otherwise False.

<

Less Than

Purpose: Checks if the value on the left side is *less than* the value on the right side.

Syntax: a < b

Returns: True if a is less than b, otherwise False.

Quick Recap

>=

Greater Than or Equal To

Purpose: Checks if the value on the left side is *greater than or equal* to the value on the right side.

Syntax: a >= b

Returns: True if a is greater than or equal to b, otherwise False.

<=

Less Than or Equal To

Purpose: Checks if the value on the left side is *less than or equal to* the value on the right side.

Syntax: a <= b

Returns: True if a is less than or equal to b, otherwise False.

02

Chaining Comparison Operators

Chaining Comparison Operators

Purpose: Python allows you to chain multiple comparison operators in a single expression.

Syntax: a <op> b <op> c

Returns: True if the entire chain of comparisons is true, otherwise False.

Python

```python
print(3 < 5 < 7)  # True
print(3 < 5 > 4)  # True
print(3 < 5 > 6)  # False
```

03

Comparing Different Datatype

Comparing Different Data Types

Python allows comparison between different data types in some cases, though the behavior may not always be intuitive.

Be **cautious** when comparing different data types, as Python 3 **does not allow arbitrary comparisons** between incompatible types (e.g., comparing a string with an integer).

Python

```python
print(3 == 3.0)  # True (integer and float comparison)
print(3 < "3")  # TypeError in Python 3 (int and str cannot be compared)
```

Comparison Between Integers and Floats

Python allows comparisons between integers and floats **because these data types are both numeric**.

Python

```
print(10 == 10.0)  # True
print(10 < 10.5)  # True
print(10.5 > 10)  # True
```

Explanation: Python automatically converts the integer to a float during the comparison, so 10 == 10.0 is True.

Comparison Between Integers/Floats and Strings

In Python 3, comparing numeric types (like integers or floats) directly with strings results in a TypeError. This is **because Python does not know how to logically compare numbers and strings.**

Python

```
print(10 < "10")
# TypeError: '<' not supported between instances of 'int' and 'str'
```

Explanation: The comparison fails because Python does not attempt to convert the types; it raises an error instead.

Comparing Strings with Other Strings

String comparisons in Python are based on the lexicographical order (dictionary order) of characters, where **each character is compared based on its ASCII value.**

Python

```
print("apple" < "banana")  # True
print("Apple" < "apple")   # True
print("apple" == "apple")  # True
print("apple" < "apricot")  # True
```

Explanation: Python compares strings character by character. In the ASCII table, capital letters come before lowercase letters, so "Apple" < "apple" is True.

Comparison Between Integers/Floats and Boolean Values

Python treats **True** as **1** and **False** as **0** in comparisons.

Python

```
print(True == 1)  # True
print(False == 0)  # True
print(True > 0)  # True
print(False < 1)  # True
```

Explanation: When comparing a Boolean value with an integer or float, Python internally converts the Boolean to its numeric equivalent (1 or 0).

04

Practical Example

Practical Example - 1

Using Comparison Operators in Conditional Statements

Code:

```
age = 20
if age >= 18:
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote.")
```

Explanation: This code *checks if age is greater than or equal to 18*. If true, it prints a message stating eligibility to vote.

Practical Example - 2

Filtering a List with a Comparison

Code:

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

even_numbers = [num for num in numbers if num % 2 == 0]

print(even_numbers)
```

Explanation: This list comprehension *filters out even numbers from the numbers list* using the equality operator (==).

Practical Example - 3

Chaining Comparisons

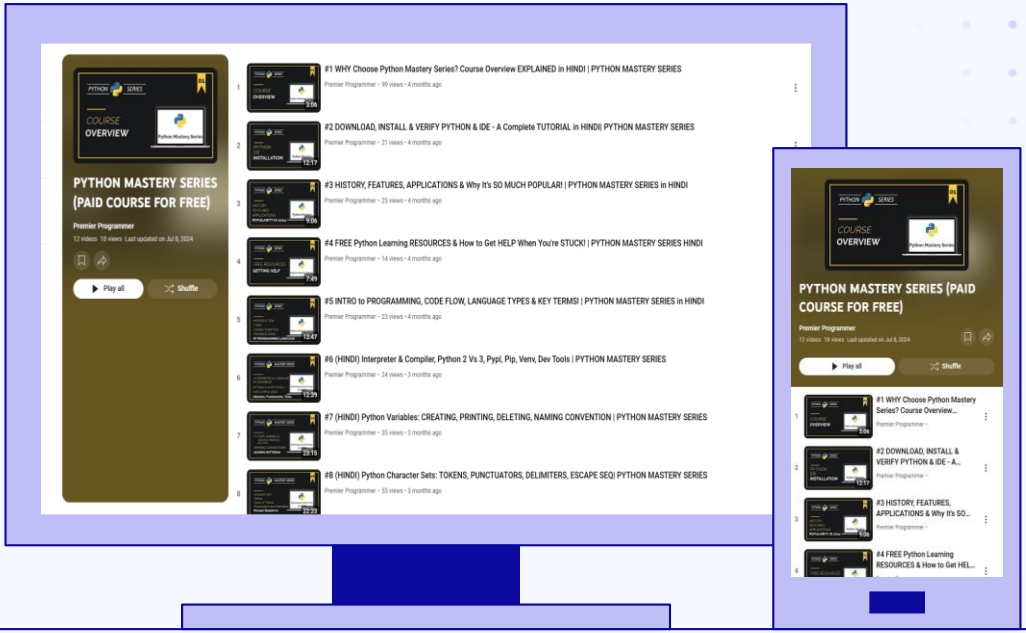
Code:

```
temperature = 25
if 20 <= temperature <= 30:
    print("The temperature is in the comfortable range.")
else:
    print("The temperature is outside the comfortable range.")
```

Explanation: This example checks *if the temperature is within the range of 20 to 30*, inclusive.

WATCH

Level up your coding with each episode in this focused Python series.



PYTHON MASTERY SERIES (PAID COURSE FOR FREE)

Premier Programmer

12 videos 18 views Last updated on Jul 8, 2024

1. #1 WHY Choose Python Mastery Series? Course Overview EXPLAINED in HINDI | PYTHON MASTERY SERIES
   Premier Programmer • 99 views • 4 months ago
2. #2 DOWNLOAD, INSTALL & VERIFY PYTHON & IDE - A Complete TUTORIAL in HINDI| PYTHON MASTERY SERIES
   Premier Programmer • 21 views • 4 months ago
3. #3 HISTORY, FEATURES, APPLICATIONS & Why It's SO MUCH POPULAR! | PYTHON MASTERY SERIES in HINDI
   Premier Programmer • 25 views • 4 months ago
4. #4 FREE Python Learning RESOURCES & How to Get HELP When You're STUCK! | PYTHON MASTERY SERIES HINDI
   Premier Programmer • 14 views • 4 months ago
5. #5 INTRO to PROGRAMMING, CODE FLOW, LANGUAGE TYPES & KEY TERMS! | PYTHON MASTERY SERIES in HINDI
   Premier Programmer • 23 views • 4 months ago
6. #6 (HINDI) Interpreter & Compiler, Python 2 Vs 3, PypI, Pip, Venv, Dev Tools | PYTHON MASTERY SERIES
   Premier Programmer • 24 views • 3 months ago
7. #7 (HINDI) Python Variables: CREATING, PRINTING, DELETING, NAMING CONVENTION | PYTHON MASTERY SERIES
   Premier Programmer • 35 views • 3 months ago
8. #8 (HINDI) Python Character Sets: TOKENS, PUNCTUATORS, DELIMITERS, ESCAPE SEQ| PYTHON MASTERY SERIES
   Premier Programmer • 55 views • 3 months ago

Next Video!

Logical Operator–
In Depth