

[Back to Code in Place](#)

Booleans

A boolean is an expression that evaluates to either True or False. Here is an example of a boolean expression:

```
1 < 2
```

It asks the question: is the value on the left smaller than the value on the right?

Comparison Operators

To compare values in python, we can use comparison operators .

Here are the comparison operators:

Operator	Meaning	Example	Value
==	equals	1 + 1 == 2	True
!=	does not equal	3.2 != 2.5	True
<	less than	10 < 5	False
>	greater than	10 > 5	True
<=	less than or equal to	126 <= 100	False
>=	greater than or equal to	5.0 >= 5.0	True

We can use comparison operators to make conditions for if statements and while loops

For example:

```
if 1 < 2:  
    print("1 is less than 2")
```

Here's another example:

```
num = int(input("Enter a number: "))  
if num == 0:  
    print("That number is 0")  
else:  
    print("That number is not 0")
```

Logical Operators

Logical operators can be used to change or combine boolean statements.

Here are the logical operators:

Operator	Example	Result
not	not (2 == 3)	True
and	(2 == 3) and (-1 < 5)	False

Introduction

Interpreter

Variables

print

input

Math and expressions

Constants

random

Booleans

if

while

for

Operator	Example	Result
or	(2 == 3) or (-1 < 5)	True

Boolean Variables

Just like how you can have variables that store integer or string values, you can also have variables that store boolean values.

```
# Store expressions that evaluate to True/False
x = 1 < 2    #True
y = 5.0 == 4.0  #False

# Directly set to True/False
is_sheltering = True
is_raining = False

play_again = input('Play again? "y" or "n"') == 'y'
if play_again:
    ...
```

[Back to Code in Place](#)