

TRUTHINESS

An Introduction to Computer Science



Let's learn about Truthiness.

Truthiness

- No Booleans required!
- Or `<`, `>`, `!=`, `==`, `<=`, `>=`, `or`, `and`, `not`



Unlike some languages, Python does not require that IF statements have boolean expressions in their conditional.

This may seem surprising, but it is actually an extremely convenient feature named Truthiness.

The idea is that any expression, whether it is an integer, string, boolean, or otherwise, can be evaluated in a conditional.

Type Truthiness

Type	False	True
Boolean	False	True
Integer	0	Non-zero number (e.g., 5, -3)
Float	0.0	Non-zero number (e.g., -3.0, 2.4)
String	""	Non-empty string (e.g., "Hello")
None	None	



Any expression can be evaluated as a conditional according to the rules of Truthiness. How it is evaluated depends on its type.

String Example

Truthy
use of
variable

```
name = input("What is your name?")  
if name:  
    print("Your name is", name)  
else:  
    print("No name given.")
```



Let's look at a simple example, where we take some input from a user.
If the user entered an empty string, then we'll print out a different message.

Unnecessary Comparisons

```
if a_string != "":  
    ...  
→ if a_string:  
    ...
```

```
if a_number != 0:  
    ...  
→ if a_number:  
    ...
```



When you need to check if a string is not empty, or a number is not 0, truthiness is the way to go.

Notice how each of the following can become much more concise and clear.