8.2. Boolean Values and Boolean Expressions

The Python type for storing true and false values is called `bool`, named after the British mathematician, George Boole. George Boole created *Boolean Algebra*, which is the basis of all modern computer arithmetic.

There are only two **boolean values**. They are `True` and `False`. Capitalization is important, since `true` and `false` are not boolean values (remember Python is case sensitive).

Save & Run 5/9/2021, 9:07:35 AM - 2 of 2 Show in CodeLens

```
print(True)
print(type(True))
print(type(False))

```

```
True
<class 'bool'>
<class 'bool'>
```

Activity: 1 -- ActiveCode (ac7_2_1)

Note

Boolean values are not strings!

It is extremely important to realize that True and False are not strings. They are not surrounded by quotes. They are the only two values in the data type `bool`. Take a close look at the types shown below.

Save & Run 5/9/2021, 9:07:37 AM - 2 of 2 Show in CodeLens

```
print(type(True))
print(type("True"))

```

```
<class 'bool'>
<class 'str'>
```

Activity: 2 -- ActiveCode (ac7_2_2)

A **boolean expression** is an expression that evaluates to a boolean value. The equality operator, `==`, compares two values and produces a boolean value related to whether the two values are equal to one another.

Save & Run 5/9/2021, 9:07:40 AM - 2 of 2 Show in CodeLens

```
print(5 == 5)
print(5 == 6)

```

```
True
False
```

Activity: 3 -- ActiveCode (ac7_2_3)

In the first statement, the two operands are equal, so the expression evaluates to `True`. In the second statement, 5 is not equal to 6, so we get `False`.

The `==` operator is one of six common **comparison operators**; the others are:

```
x != y               # x is not equal to y
x > y                # x is greater than y
x < y                # x is less than y
x >= y               # x is greater than or equal to y
x <= y               # x is less than or equal to y
```

Although these operations are probably familiar to you, the Python symbols are different from the mathematical symbols. A common error is to use a single equal sign (`=`) instead of a double equal sign (`==`). Remember that `=` is an assignment operator and `==` is a comparison operator. Also, there is no such thing as `=<` or `=>`.

Note too that an equality test is symmetric, but assignment is not. For example, if `a == 7` then `7 == a`. But in Python, the statement `a = 7` is legal and `7 = a` is not. (Can you explain why?)

Check your understanding

condition-2-1: Which of the following is a Boolean expression? Select all that apply.

- [x] A. True
- [x] B. 3 == 4
- [ ] C. 3 + 4
- [x] D. 3 + 4 == 7
- [ ] E. "False"

Check me Compare me

✔ Correct.
A. True and False are both Boolean literals.
B. The comparison between two numbers via == results in either True or False (in this case False), both Boolean values.
D. 3+4 evaluates to 7. 7 == 7 then evaluates to True, which is a Boolean value.

Activity: 4 -- Multiple Choice (question7_2_1)

You have attempted 6 of 5 activities on this page

8.1. Intro: What we can do with Turtles and Conditionals">
8.1. Intro: What we can do with Turtles and Conditionals">
✔ Completed. Well Done!
8.3. Logical operators">
8.3. Logical operators">Next Section - 8.3. Logical operators

| Back to top