

# Control Through True and False

## Contents

- [Use](#)
- [Conditional Statements](#)
- [Boolean Operators](#)
- [Booleans Are Data Types](#)

This page is under construction ...

Booleans are one of five basic **types** of objects in Python:

1. **int** (integer)
2. **float** (float)
3. **bool** (Boolean)
4. **str** (string)
5. **None**

Mentioned at the start of this part is the need for control over programs. That comprises the ability to repeat operations and the ability to determine whether something is or is not something else. Boolean expressions are a large part of the latter, and they are fairly simple in both syntax and operation.

```
print(1 == 1)
# True

print(1 < 2)
# True

print(1 > 2)
# False

print (1 <= 2)
# True
```

```
True
True
False
True
```

## Use

If boolean expressions are used to control your programs, how is that done? In two primary contexts:

- **while** loops (to be covered in a bit) and
- conditional statements marked by keywords **if**, **elif**, and **else**

```
# using for conditional statements

# data is a string of hexadecimal characters
data =
'6e6f77206973207468652074696d6520746f2073656520746865206b696e6720746865207469
6d6520746'

# but the length will determine what we do with it ...

if len(data) > 200:
    print('Data is too large to process.')

elif len(data) > 50 and len(data) <= 200:
    print('Process data with XYZ protocol.')

else:
    print('Data is useless.')
```

```
Process data with XYZ protocol.
```

## Conditional Statements

Above the conditional statements begin with **if**, **elif**, and **else**. A series of conditional statements always begins with **if**. It may end there, i.e., you may want to know only whether something is **True** and don't care if it isn't. But if you do, depending on the number of possibilities, you use either **else** or **elif** or both.

Let's look closer at the syntax, first the **if**:

```
if <boolean expression>:
    <statement>
```

- **if** is the keyword that kicks off the code
- **<boolean expr>** is an expression that evaluates to either **True** or **False**
- **:** colon is required immediately after the expression; it causes Python to anticipate an indented code block to follow
- **<statement>** is a Python statement, i.e., the command to follow; however, that command is only executed if and only if the expression above can be rendered true.

## Boolean Operators

### Booleans Are Data Types

```
type(True)
```

```
bool
```

```
type(False)
```

```
bool
```