

# Objects, Types, Operators, and Expressions

---

To do anything useful in our programs, Python will need to represent data such as numbers, words and booleans. We will learn more about these in Units 2 and 3.

For now we will introduce the most common 4 built-in data types.

## 1. Objects and Types

In Python, we will work with **objects**. Every **object** has a **type** that tells the program what can be done with the **object**. Python has a number of built-in **data types** which Python uses internally to represent data.

We will look at 4 of these **data types** to start. We will cover more later in the course as we progress.

Data Type	Description
<code>int</code>	An integer. This is a whole number, it can be positive, negative or zero. e.g. <code>5</code>
<code>float</code>	A decimal number. e.g. <code>3.14</code>
<code>str</code>	Text. It consists of individual characters. Strings are enclosed in single quotation marks <code>'</code> or double quotation marks <code>"</code> . e.g. <code>"Hello World"</code>
<code>bool</code>	The values <code>True</code> or <code>False</code> . Used to make decisions, more in Unit 3

A very useful function in python that we can use is `type()`. This lets us ask Python what the **type** of an **object** is.

Type the following into the **console**:

```
type(10)
```

You will see the output:

```
<class 'int'>
```

This is python telling you that `10` is an **object** of **type** `int`. For now, read class as **type**.

Try it for these other **objects**.

```
type(10.3)
```

```
type("This is a string")
```

```
type(True)
```

The console automatically prints out the last command. To do this in **main.py** you will need to use the **print()** function. Try the following in **main.py**.

```
print(type("This is a string"))
```

This first gets the type of the object, here a **str** and then passes that to the **print()** function.

## 2. Operators and Expressions

We can combine **objects** with **operators** to form **expressions**. When evaluated, these **expressions** produce a new **object**.

For example, we can combine the **objects** **10** and **5** with the **+** **operator**,

```
10 + 5
```

to create a new object **15** of type **int**.

All built-in **data types** have **operators** that we can use to form **expressions**.

For example, we can test if two **expressions** are equal with the **==** comparison **operator**. Type the following **expression** into the **console**,

```
10 + 5 == 3 * 5
```

This will return **True**. Python knows how to test whether two numbers are equal and returns you a new **object** of type **bool**.

We will cover operators for numbers, strings, and bools in their respective lessons in this unit.

---

## 3. Help Function

The **help()** function gets information about an object (it can also be used for other things.)

Try typing the following into the console:

```
help(str)
```

Press **Ctrl** + **c** to exit the help.

## === TASK ===

---

Update **main.py** to print out the type of the following expressions. You will need to use the **print()** and the **type()** function.

Make sure you have read all of the above before attempting this.

`12 + 5`

`5 + 3.0`

`12 / 5`

`12 + 5 == 5 * 3.0`