Writing Our First Code

In this lesson, we'll examine one of the simplest codes in Python syntax.

We'll cover the following

By now, we've learned what kind of language Python is. We are finally ready to start writing code! So, let's move on to the fun stuff.

The `print` Statement

Whenever we learn a new language, it is an age-old tradition to start by displaying the text "**Hello World**" on the screen. For the remainder of this course, the terminal will act as our screen.

Every language has a different syntax for displaying or *printing* something on the screen.

Since Python is one of the most readable languages out there, we can print data on the terminal by simply using the `print` statement.

Here's what the statement looks like:

```
print (data)
```

Whatever we need to print is encapsulated in the parentheses following the `print` keyword. Let's try printing "Hello World" on the terminal:

```
print("Hello World")
```

The text `Hello World` is bounded by quotation marks because it is a *string* or a group of characters, more on this later.

Next, we'll print a few numbers. Each call to `print` moves the output to a new line:

```
print(50)
print(1000)
print(3.142)
```

Printing Multiple Pieces of Data

We can even print multiple things in a single `print` command; we just have to separate them using **commas**.

Multiple values separated by commas

Let's see this in action:

```
print(50, 1000, 3.142, "Hello World")
```

By default, each `print` statement prints text in a new line. If we want multiple `print` statements to print in the same line, we can use the following code:

```
print("Hello", end="")
print("World")

print("Hello", end=" ")
print("World")
```

The value of `end` is appended to the output and the next `print` will continue from here.

Comments

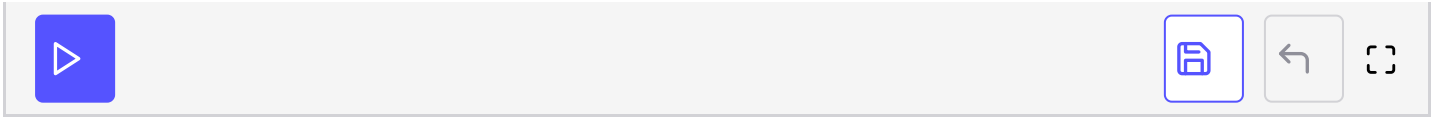
Comments are pieces of text used to describe what is happening in the code. They have no effect on the code whatsoever.

A comment can be written using the `#` character:

```
print(50)  # This line prints 50
print("Hello World")  # This line prints Hello World

# This is just a comment hanging out on its own!

# For multi-line comments, we must
# add the hashtag symbol
# each time
```



An alternative to these multi-line comments (line 4 - 8) are **docstrings**. They are encased in triple quotes, `"""`, and can be used to replace multi-line comments:

```
""" Docstrings are pretty cool
for writing longer comments
or notes about the code"""
```

That brings us to the end of this section. Be sure to check out the quiz in order to test what you have learned so far.

In the next section, we will learn about the different data types and operators in Python.