Programming with Python

Lesson 1: Setting up and basics

November 1st, 2016

What is Python? What is programming?

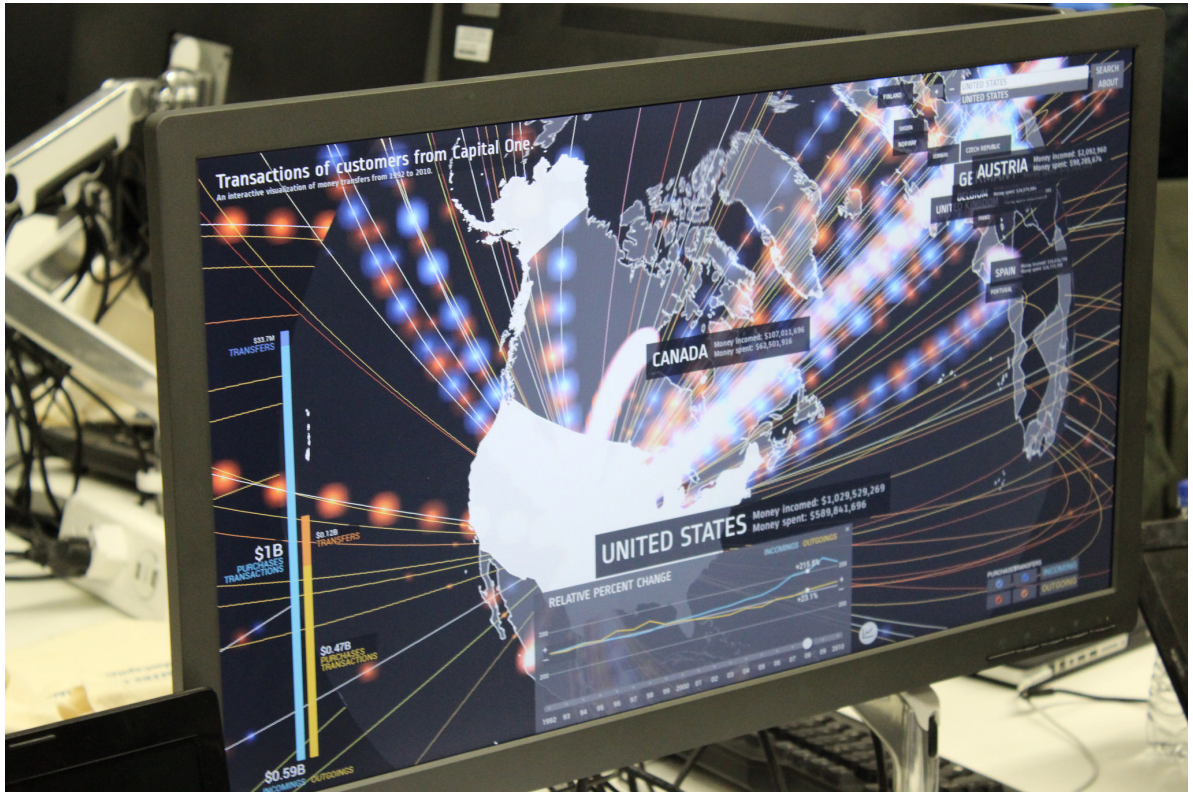


Figure: Image curtosy of HackSocNotts

Making a cup of tea

How do you make a cup of tea?

Making a cup of tea

How do you make a cup of tea?

1. Boil the kettle.

Making a cup of tea

How do you make a cup of tea?

1. Boil the kettle.
2. Get a mug and a teabag.

Making a cup of tea

How do you make a cup of tea?

1. Boil the kettle.
2. Get a mug and a teabag.
3. Put a teabag in the mug.

Making a cup of tea

How do you make a cup of tea?

1. Boil the kettle.
2. Get a mug and a teabag.
3. Put a teabag in the mug.
4. Stir and wait.

Making a cup of tea

How do you make a cup of tea?

1. Boil the kettle.
2. Get a mug and a teabag.
3. Put a teabag in the mug.
4. Stir and wait.
5. Enjoy!

Making a cup of tea

Computers need a little more help...

Making a cup of tea

Computers need a little more help...

1. GET Water.

Making a cup of tea

Computers need a little more help...

1. GET Water.
2. GET Kettle.

Making a cup of tea

Computers need a little more help...

1. GET Water.
2. GET Kettle.
3. ADD Water to Kettle.

Making a cup of tea

Computers need a little more help...

1. GET Water.
2. GET Kettle.
3. ADD Water to Kettle.
4. SWITCH ON Kettle.

Making a cup of tea

Computers need a little more help...

1. GET Water.
2. GET Kettle.
3. ADD Water to Kettle.
4. SWITCH ON Kettle.
5. SLEEP until Kettle finished.

Making a cup of tea

Computers need a little more help...

1. GET Water.
2. GET Kettle.
3. ADD Water to Kettle.
4. SWITCH ON Kettle.
5. SLEEP until Kettle finished.
6. GET Mug.

Making a cup of tea

Computers need a little more help...

1. GET Water.
2. GET Kettle.
3. ADD Water to Kettle.
4. SWITCH ON Kettle.
5. SLEEP until Kettle finished.
6. GET Mug.
7. Etc...

What is Python?

- A procedural, interpreted scripting language.

What is Python?

- A procedural, interpreted scripting language.
- Two major versions: 2.7 (2010) & 3.x (Currently at 3.5 as of 2015).

What is Python?

- A procedural, interpreted scripting language.
- Two major versions: 2.7 (2010) & 3.x (Currently at 3.5 as of 2015).
- We will be using the *pycharm* IDE to write our python code.

Installing the Python 3.5.2 interpreter

- Windows: **https://www.python.org/downloads/windows/**
- Mac OSX: **https://www.python.org/downloads/mac-osx/**
- Linux: Either **sudo apt-get install python3** or **https://www.python.org/downloads/source/**

If at any time you get stuck, just stick your hand up and we'll come help.

Installing PyCharm

https://www.jetbrains.com/pycharm/download/

Keywords

Before we begin, below are a couple of keywords used quite often when programming:

Syntax - How Python understands your code. Kind of like the grammar of your code.

Console - Where text is printed to and where you can input text.

Interpreter - The thing Python uses to understand your code.

Operator - Things like plus, minus, multiply and divide. They do things.

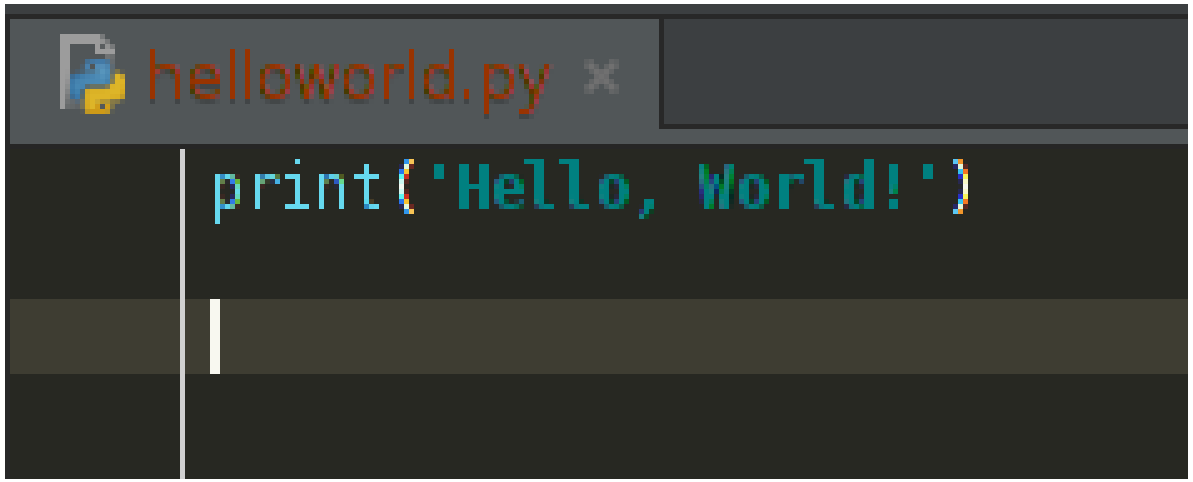
Hello World!

Now that we have all prerequisites installed, load up PyCharm and start a new project.

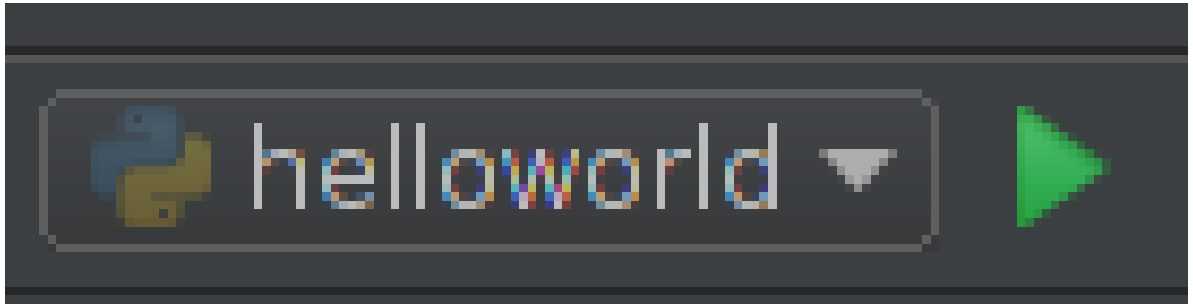
Hello World!

Now that we have all prerequisites installed, load up PyCharm and start a new project.
We are going to build a simple program which prints the words "Hello, World!".

The *print* command prints something to the console:



Now that we have our code, we want to execute it. This can be done by hitting the green arrow in the top right corner.



Feel free to change the code inside the print to print whatever message you want!

Reading in input

A program that just prints a string is pretty rubbish. What'd be great would be if we could tell it what to print.

Reading in input

A program that just prints a string is pretty rubbish. What'd be great would be if we could tell it what to print.
The *input* command allows us to input what we want into the program. However, we need somewhere to put our input, before we can display it.

Now our program can display our name:

helloworld.py

```
print("Please input your name.")

user_input = input()

print('Hello, ' + user_input + '!')
```

Now our program can display our name:

helloworld.py

```
print("Please input your name.")

user_input = input()

print('Hello, ' + user_input + '!')
```

What have we actually done here?

Variables

Variables are a container for things. They can change when you want them to and have any name you want.

Variables

Variables are a container for things. They can change when you want them to and have any name you want.
In the previous example we specified a new variable called *name* where we stored the value we wanted to input.

Variables

Variables are a container for things. They can change when you want them to and have any name you want.
In the previous example we specified a new variable called *name* where we stored the value we wanted to input.
Variables can be many different things including:

Variables

Variables are a container for things. They can change when you want them to and have any name you want.
In the previous example we specified a new variable called *name* where we stored the value we wanted to input.
Variables can be many different things including:

- An integer (1,2,3 etc)

Variables

Variables are a container for things. They can change when you want them to and have any name you want.
In the previous example we specified a new variable called *name* where we stored the value we wanted to input.
Variables can be many different things including:

- An integer (1,2,3 etc)
- A character ('a','b','c' etc)

Variables

Variables are a container for things. They can change when you want them to and have any name you want.
In the previous example we specified a new variable called *name* where we stored the value we wanted to input.
Variables can be many different things including:

- An integer (1,2,3 etc)
- A character ('a','b','c' etc)
- A string ("Hello, World!", "I love python xo" etc)

Variables

Variables are a container for things. They can change when you want them to and have any name you want.
In the previous example we specified a new variable called *name* where we stored the value we wanted to input.
Variables can be many different things including:

- An integer (1,2,3 etc)
- A character ('a','b','c' etc)
- A string ("Hello, World!", "I love python xo" etc)
- A float (1.45345, 24.4562389, 7.4234 etc)

Variables

Variables are a container for things. They can change when you want them to and have any name you want.
In the previous example we specified a new variable called *name* where we stored the value we wanted to input.
Variables can be many different things including:

- An integer (1,2,3 etc)
- A character ('a','b','c' etc)
- A string ("Hello, World!", "I love python xo" etc)
- A float (1.45345, 24.4562389, 7.4234 etc)
- An array ([1,2,7,4,7], ['p', 'y', 't', 'h', 'o', 'n'] etc)

Variables

Variables are a container for things. They can change when you want them to and have any name you want.
In the previous example we specified a new variable called *name* where we stored the value we wanted to input.
Variables can be many different things including:

- An integer (1,2,3 etc)
- A character ('a','b','c' etc)
- A string ("Hello, World!", "I love python xo" etc)
- A float (1.45345, 24.4562389, 7.4234 etc)
- An array ([1,2,7,4,7], ['p', 'y', 't', 'h', 'o', 'n'] etc)
- Or more!

Variables

Variables are a container for things. They can change when you want them to and have any name you want.
In the previous example we specified a new variable called *name* where we stored the value we wanted to input.
Variables can be many different things including:

- An integer (1,2,3 etc)
- A character ('a','b','c' etc)
- A string ("Hello, World!", "I love python xo" etc)
- A float (1.45345, 24.4562389, 7.4234 etc)
- An array ([1,2,7,4,7], ['p', 'y', 't', 'h', 'o', 'n'] etc)
- Or more!

Some of these types can do things others can't. For example, we can subtract an integer from another, but we can't subtract a string from another.

A simple adding calculator

Using these variable types, we can convert our input into a different type! To convert a variable into a different type, we wrap the variable in the type we want to convert it to.

A simple adding calculator

Using these variable types, we can convert our input into a different type! To convert a variable into a different type, we wrap the variable in the type we want to convert it to.

```
print("Please input a number.")

input1 = int(input())
```

A simple adding calculator

Using these variable types, we can convert our input into a different type! To convert a variable into a different type, we wrap the variable in the type we want to convert it to.

```
print("Please input a number.")

input1 = int(input())
```

Here, the input we are asking for is being turned into an integer. Another name for this is we are **casting** the input to an integer.
Q: What happens if you put in something other than a number?

Our lovely calculator

Building on from that, we can combine what we have talked about to make a very simple calculator, which asks for two numbers, adds them together (using the $+$ operator), and then outputs them to the user.

Our lovely calculator

Building on from that, we can combine what we have talked about to make a very simple calculator, which asks for two numbers, adds them together (using the $+$ operator), and then outputs them to the user.

helloworld.py

```
print("Welcome to our number adder!!")
print("Please input a number.")

input1 = int(input())

print("Please input another number.")

input2 = int(input())

output = input1 + input2

print(str(input1) + " + " + str(input2) + " = " + str(output))
```

Our lovely calculator

Building on from that, we can combine what we have talked about to make a very simple calculator, which asks for two numbers, adds them together (using the $+$ operator), and then outputs them to the user.

helloworld.py

```
print("Welcome to our number adder!!")
print("Please input a number.")

input1 = int(input())

print("Please input another number.")

input2 = int(input())

output = input1 + input2

print(str(input1) + " + " + str(input2) + " = " + str(output))
```

Note we had to **cast** our integer variables to strings using *str()* for *print* to work.

That's all for tonight!

To summarise:

- We have installed Python and PyCharm
- We have learnt about variables and variable types
- We have learnt about printing and asking for input from the user
- We have made a sweet ass calculator!

For next week

Source code plus lecture slides will be available online soon after the lesson.
If you are new to HackSocNotts, please join us on *http://hacksocnotts.slack.com*.
If you have any questions, feel free to ask now or over slack.