GOVERNMENT OF WESTERN AUSTRALIA

TAFE WA South Metropolitan

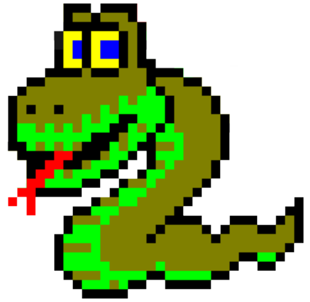
Introduction to Python Programming

Session 1

Overview

- Python History
- Python Philosophy
- Features
- Why use Python
- Popular use of Python
- Installing and Running Python

Python History



- Invented in the Netherlands
  - Born on 20th February 1991, by Guido van Rossum
  - Named after Monty Python (British comedy)
- Interpreted *high-level programming* language for general-purpose programming
  - Open sourced
  - Administered by the Python Software Foundation.
- Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace.

Python History

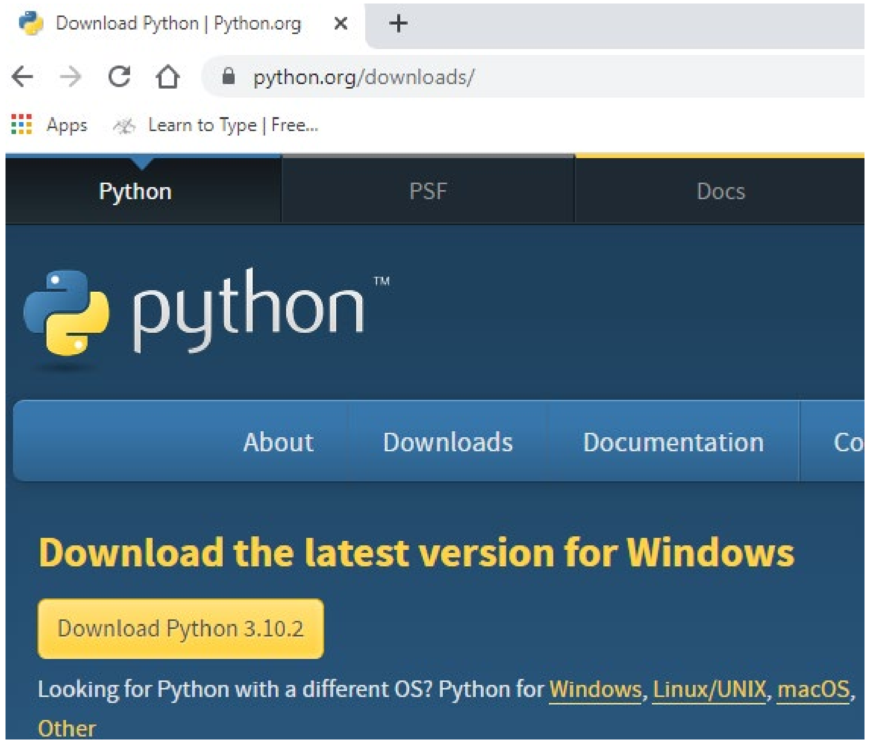
- On the origins of Python, Van Rossum wrote in 1996:
- *...In December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus).*

Python History

- **Python 2.x**
  - was released on 16 October 2000 and had many major new features, including a cycle-detecting garbage collector and support for Unicode. With this release, the development process became more transparent and community-backed.
- **Python 3.x**
  - was released on 3 December 2008 after a long testing period. It is a major revision of the language that is *not backward-compatible* with previous versions. However, many of its major features have been backported to the backward-compatible Python 2.6.x and 2.7.x version series.
  - We will be using Python 3.x.

Installing Python

- It is pre-installed on Linux/UNIX and MAC OS X
- Download from:
  - https://www.python.org/downloads/
- Currently (Jan, 2022)
  - Latest Python 3 Release - Python 3.10.2



Download the latest version for Windows

Download Python 3.10.2

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Python IDE

- IDE - Integrated Development Environment
  - increases programmer productivity by editing source code, running files, and debugging.
- They are several options available, for example:
  - IDLE – is a default editor that comes with Python
  - PyCharm is one of the widely used Python IDE
  - Visual Studio Code - is an open-source environment developed by Microsoft
  - Atom - is a useful code editor tool

Python Documentation

- The official documentation for Python
  - https://docs.python.org/3/
- Topic example: Whetting Your Appetite
  - "Python is an interpreted language, which can save you considerable time during program development because *no compilation and linking* is necessary. The interpreter can be used interactively, which makes it easy to experiment with features of the language, to write throw-away programs, or to test functions during bottom-up program development."

Available at: https://docs.python.org/3/tutorial/appetite.html

Python Philosophy

- An open-source cross platform
  - freely usable and distributable
- An interpreted high-level and general-purpose
- Coherence
  - not hard to read, write and maintain (simple is better than complex)
- Scope
  - rapid development + large systems
- Object oriented
- Easy Integration
  - hybrid systems

Python Features

run-time program construction	handles unforeseen needs, end-user coding
interactive, dynamic nature	incremental development and testing
access to interpreter information	metaprogramming, introspective objects
wide portability	cross-platform programming without ports
compilation to portable byte-code	execution speed, protecting source code
built-in interfaces to external services	system tools, GUIs, persistence, databases, etc.

Python Features

no compiling or linking	rapid development cycle
no type declarations	simpler, shorter, more flexible
automatic memory management	garbage collection
high-level data types and operations	fast development
object-oriented programming	code structuring and reuse
classes, modules, exceptions	"programming-in-the-large" support

Why we use Python

- Shell tools
  - System admin tools, command line programs
- Extension-language work
  - Used to extend the original application
- Rapid prototyping and development
  - Quick and iterative creation
- Language-based modules
  - instead of special-purpose parsers
- Graphical user interfaces
- Database access
- Distributed programming
  - Writing programs that can coordinate over networked/distributed computer systems
- Internet scripting
  - Generally server-side scripting

Python Structure

- Modules: Python source files
  - Python can import a large variety of modules that allow for extended functionality though either the standard library, or through third party modules from the internet.
- Statements
  - *indentation matters* – instead of { }, tab indentation controls the ‘sections’ of our code
- Data structures
  - Basic data structures such as floats, integers and strings
  - Non-basic data structures such as lists, tuples and sets.

Popular uses of Python

- Video games:
  - Civilization IV
  - The Sims 4
- Commercial uses:
  - Youtube
  - Google, Facebook
  - Reddit
- Other applications
  - Dropbox
  - Blender

Popular uses of Python

- Cyber Security
  - Python is widely used as the language for penetration testing, attack automation, and scripting automation to generate data or to facilitate exploiting vulnerabilities.
  - Having a grasp on Python and continuing your learning after will definitely help in your journey.
  - Commonly used in CTF challenges as a good way to test skills.

Nervous about coding?

- While coding and scripting is essentially learning a different language, it is something that can be both puzzling and fun (like, well, a puzzle).
- **How much you get out of this units depends on how much you put into it.**
- Like learning a new language, repetition and practice is required.
- Your weekly slides are designed to give you an introduction to the major topics, piece-by-piece, but they also serve as a tool for revision.
- We will generally have a lab each week, these are to test you in the skills you learn each week and to prepare you for your assessments.
- If you can do the labs, you can do the assessments.

Try it Out!!

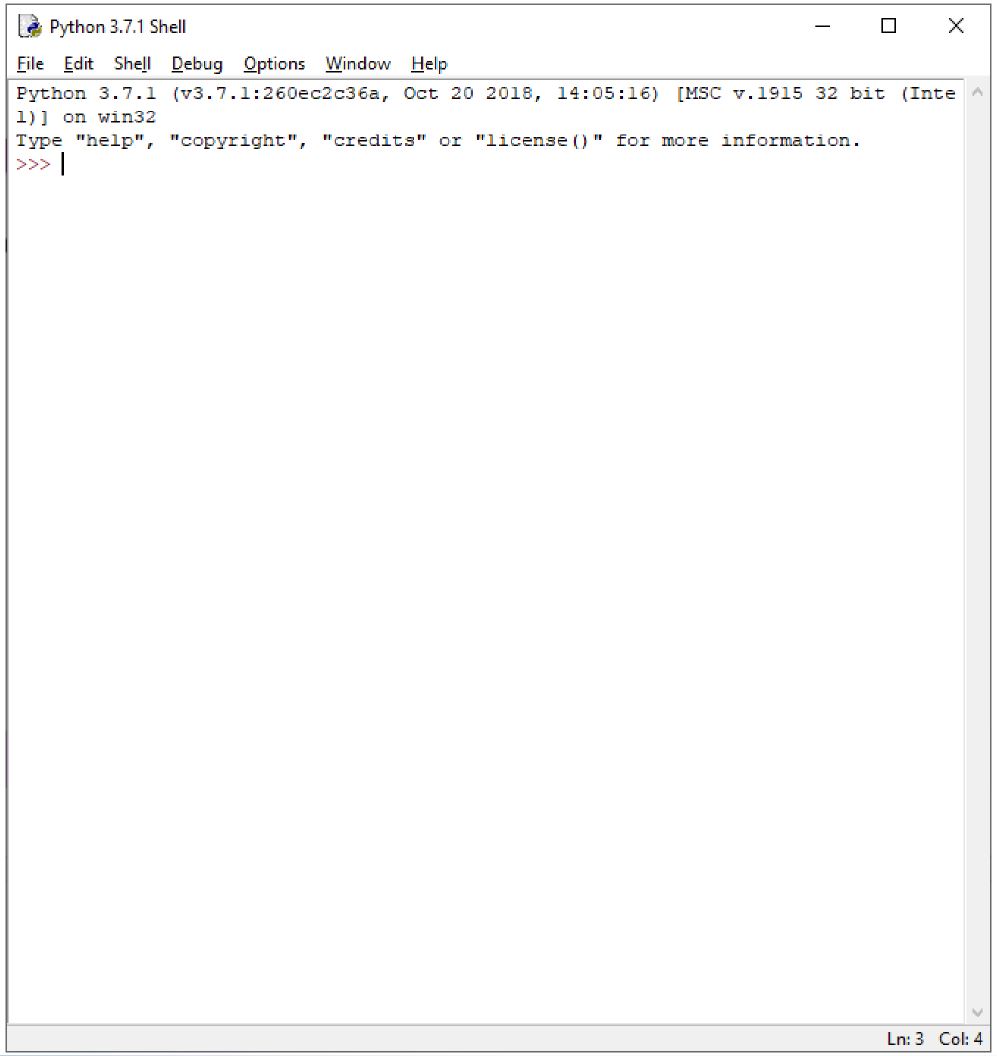
- **We should already have Python installed on our machines try looking for Python X.X in the start menu (where X is the version number), then click IDLE.**
- **Alternatively you can type “IDLE” into the start bar to find it, too.**
- If it isn’t on your computer, download Python from www.python.org
- Any version of 3.x will do for this class
  - By and large they are all mutually compatible
  - Recommended version: The latest one.

Try it Out!!

- Integrated Development Environments (or IDEs) are a set of software packaged together to give you the ability to easily write, review, debug, and run code in a singular location.
- Use IDLE initially, a simple IDE that comes pre-installed with Python. All the demonstrations in the slides will be shown with IDLE.
- If you're experienced with coding, or later down the track once you have more confidence check out more advanced IDEs:
  - *Visual Studio Code* - a free coding IDE, which has a lot of helpful plugins (like autopep8, which formats your code for you).
  - *PyCharm* – Similar to the above, which has a free community edition.

Try it Out!!

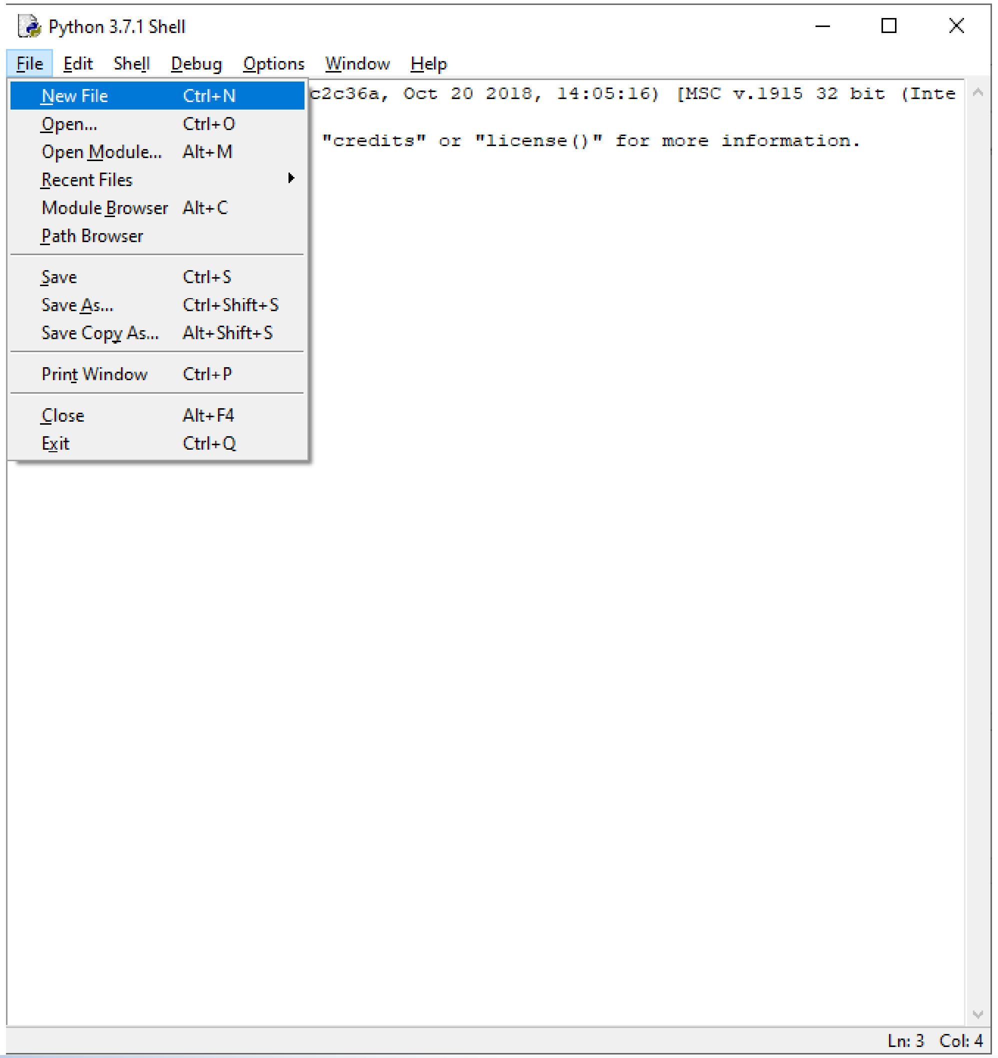
- So, once you have it up and running, you should see a screen like this:
- This is what we call the '***shell***', it's where our code will run, but it's not the place we would want to write our code.



```
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
```

Try it Out!!

- To start writing our code, we would want to click on **File > New file** (or press **Ctrl + N**).
- This will bring up a new window. From here, we can write our code in the new window, and run it in our shell.

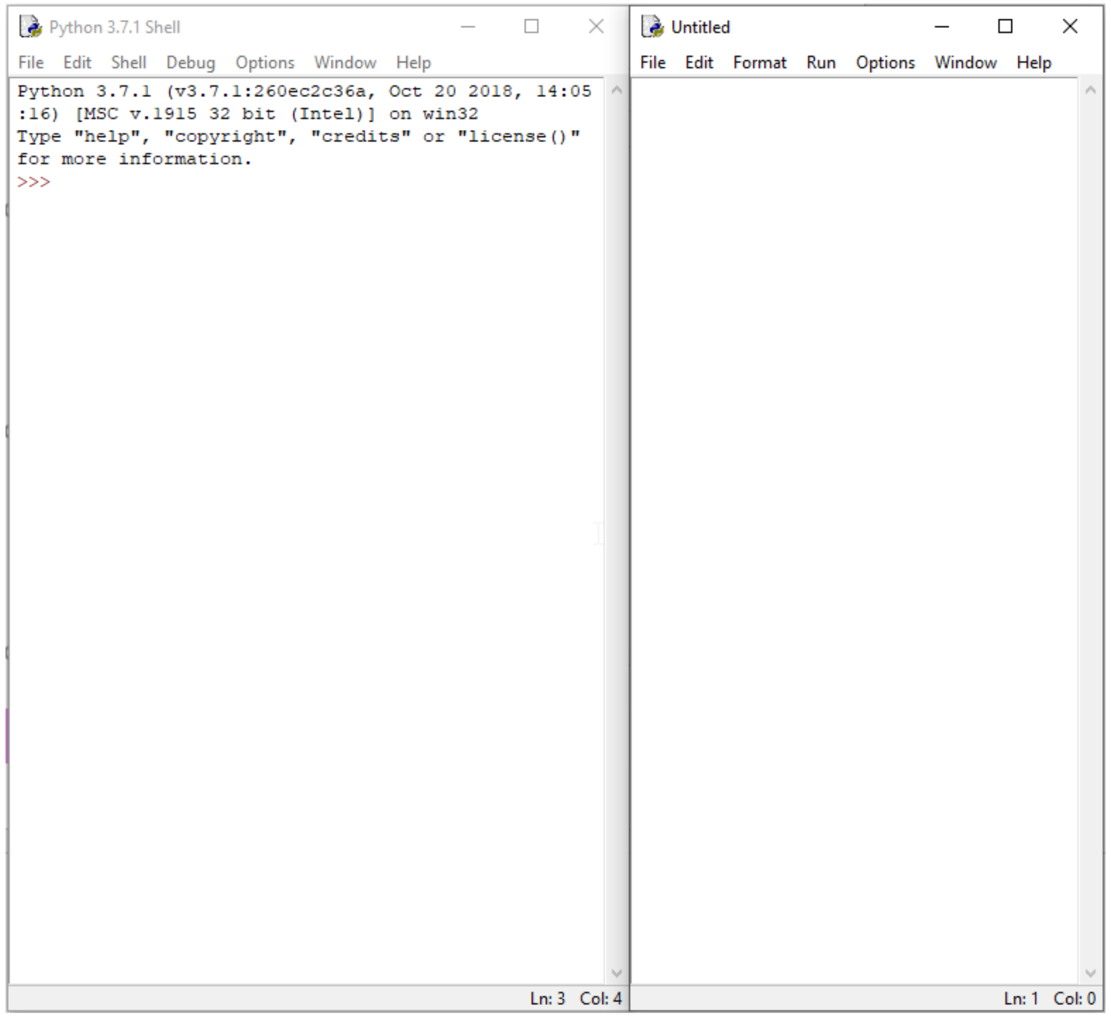


```
c2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Inte

 "credits" or "license()" for more information.
```

Try it Out!!

- So, your usual setup when coding will involve having both the windows open for the shell and the code you're writing.
- We can see this on the right.


Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Untitled
File Edit Format Run Options Window Help
Ln: 3 Col: 4
Ln: 1 Col: 0

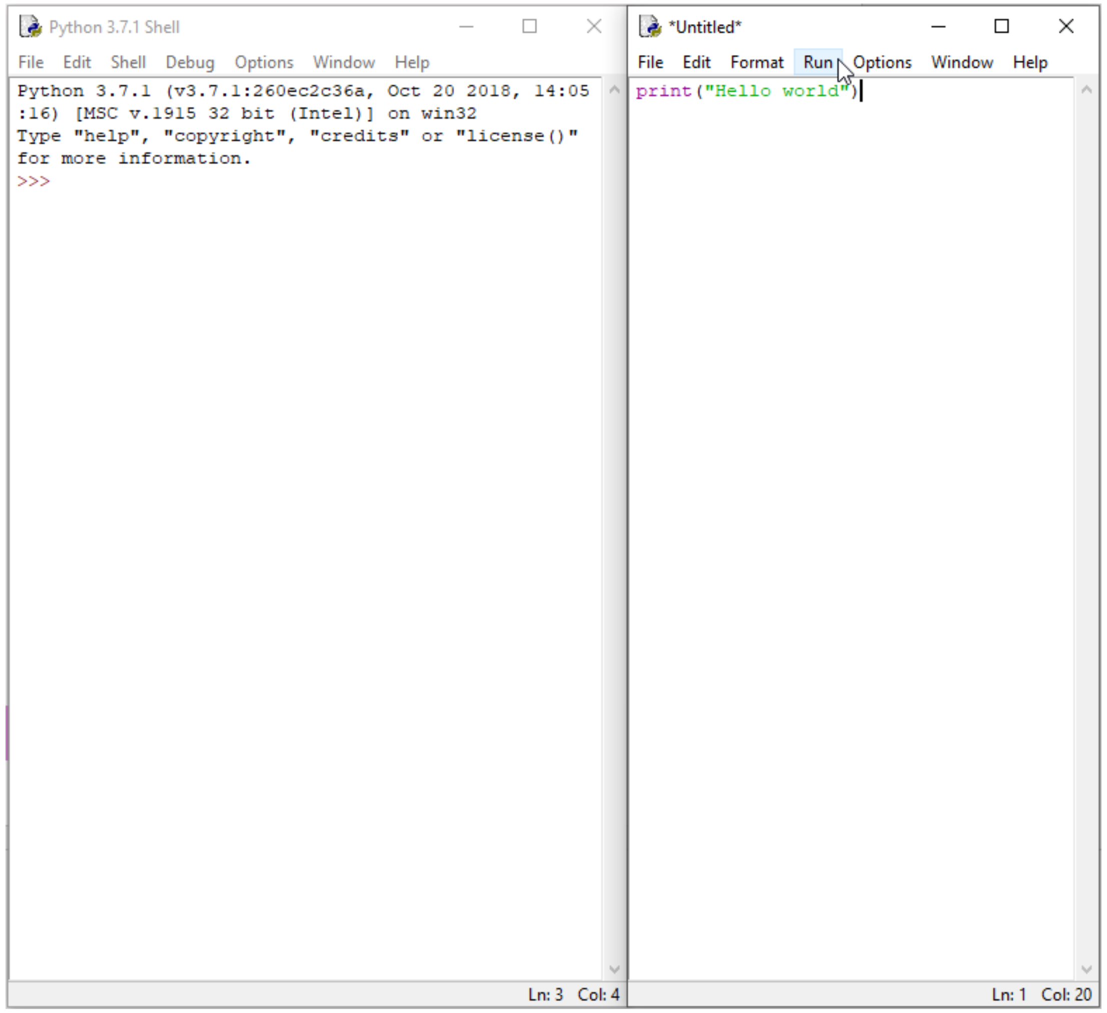
```
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05
:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()"
for more information.
>>>
```

Try it Out!!

- So, let's partake in a time-honoured tradition of coding…
- We use the print statement to send a message to the screen. We'll get to the complexities later, but for now, try writing the following in the new file:

  print("Hello world")

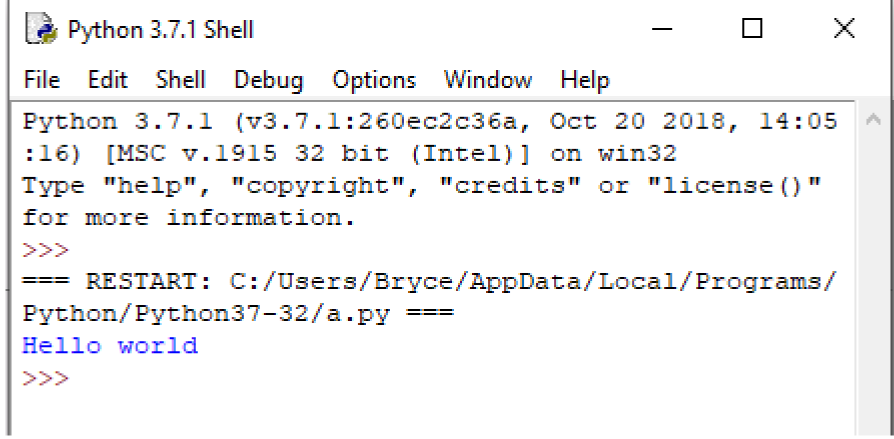
- Once done, press **Run > Run Module** (or press **F5**). Save the script somewhere, and then we should see a message appear on the shell.


Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Ln: 3 Col: 4
Untitled
File Edit Format Run Options Window Help
Ln: 1 Col: 20

```
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05
:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()"
for more information.
>>>
```

```
print("Hello world")
```

Try it Out!!



```
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05
:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()"
for more information.
>>>
=== RESTART: C:/Users/Bryce/AppData/Local/Programs/
Python/Python37-32/a.py ===
Hello world
>>>
```

- For this week, as it is an introductory class, we don’t have a lab.
- However, in the learning content section on Blackboard, there is a folder called “*Python Web Resources*”.
- This folder has links to the python website, the two previously mentioned advanced IDEs, and a “Learn Python” link.
- For the rest of this class take you time you familiarise yourself with the IDE, look at the others if you’re interested, and try your hand at the Learn Python resource to give it a bit of a go, at your own pace.

Questions?