# Supplementary notes PYP1-AUG24

## Post-course consolidation

Beginner
https://www.w3schools.com/python/
NEW: host code on W3 Spaces
ALTERNATIVE: https://replit.com/~
BEST: local machine:
FIRST Python
https://www.python.org/downloads/
THEN VS Code
https://code.visualstudio.com/download

Intermediate
https://www.geeksforgeeks.org/python-programming-language-tutorial/

Advanced
https://docs.python.org/3/tutorial/index.html

Paywall
https://realpython.com/start-here/
Real Python is excellent, in our opinion, and although many video courses are behind a paywall, there are many scholarly but clearly-written articles that are free.

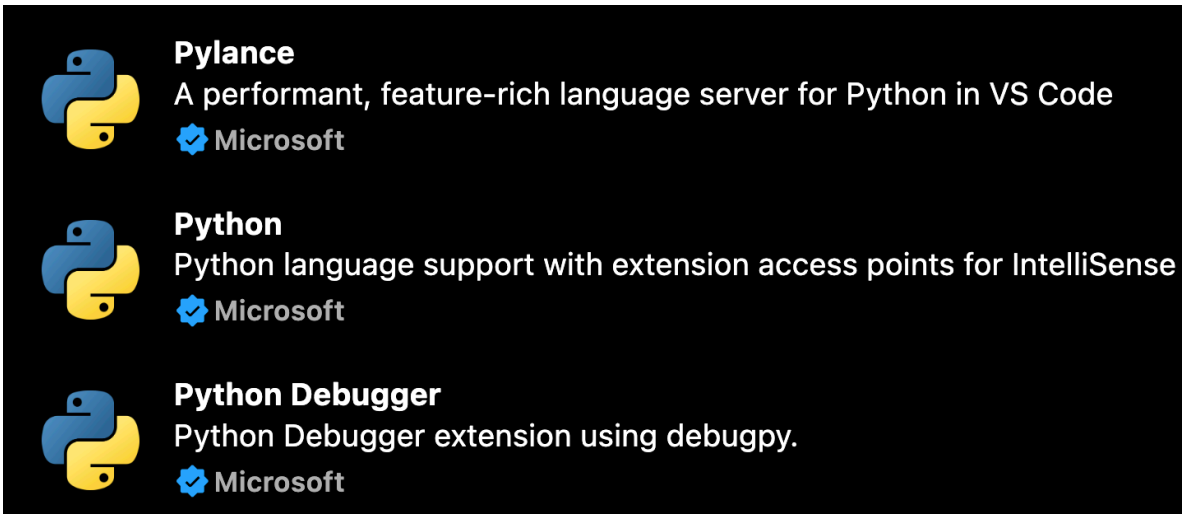
Manual exercises - code review available from me via alanl@stayahead.com

---

# Installs and extensions used in course

| | |
|---|---|
| Python latest - do first and check “add to path”. | https://www.python.org/downloads/ |
| VS Code - do second then it will pick up the Python interpreter. | https://code.visualstudio.com/download |

| | |
|---|---|
| Microsoft extensions - will prompt for these 3 as a pack | See below |
| Auto docstring generator | See below |





## VS Code shortcuts

| | |
|---|---|
| CTRL + B | toggle explorer pane / full screen |
| CTRL + +/- | up/down font size |
| CTRL + / | toggle comment on/off |
| ALT + SHIFT + click | multi-cursor select |
| ALT + SHIFT + DOWN/UP | duplicate line down/up |
| """ + RETURN | Auto docstring |

---

**notes / references on topics arising during course:**

https://docs.python.org/3/tutorial/modules.html

NOTES - README
**dir(__builtins__)**

This will output a list containing firstly builtin Exception Objects, then dunder data attributes and then what looks like a listing of the built-in functions (not built-in modules). The functions begin from abs which is at index 82. So to display just the functions from *__builtins__ we can use the slicing operator as follows:*

***>>> dir (__builtins__)[82:]***

*or more generically:*
***>>> dir (__builtins__)[dir(__builtins__).index('abs'):]***

***>>> q** #to quit help*

*to get command line commands to work copied in from PDF you may need to re-type quotes, which may be non-standard characters*

*https://docs.python.org/3/library/builtins.html*

*lists diff tuples*
*https://www.geeksforgeeks.org/python-difference-between-list-and-tuple/*

objects to string
https://www.geeksforgeeks.org/str-vs-repr-in-python/

object equality and identity
https://www.geeksforgeeks.org/difference-between-and-is-operator-in-python/

https://www.geeksforgeeks.org/difference-between-__eq__-vs-is-vs-in-python/

variable args
https://www.geeksforgeeks.org/args-kwargs-python/

number format to two decimal places:
number = 3
>>> print(f'number: {number:.2f}')
number: 3.00

## String efficiency best practice

https://www.tracedynamics.com/python-string-builder/

str.join()
str concatenation
str += concatenation assignment
io.StringIO streams

| Method | Efficiency | Ease of Use | Suitable for Large Strings |
|---|---|---|---|
| `str.join()` Method | High | High | Yes |
| String Concatenation | Medium | High | No |
| Concatenation Assignment | Low | High | No |
| `io.StringIO` Class | High | Medium | Yes |

## Short-circuiting in Python

https://realpython.com/python-or-operator/

https://mathspp.com/blog/pydonts/boolean-short-circuiting

https://www.geeksforgeeks.org/short-circuiting-techniques-python/
(First two are better imo but this has useful examples of all() and any() builtins)

See operators/**short-circuiting.md**

To run code in a module ONLY - and PREVENT it running in a script that imports from it

```
if __name__ == '__main__':
    # code here will NOT run in the importing script
```