PYTHON

Command Line methods

cd = change directory

cd .. = go to prev. directory

pwd = print working directory (Linux)

ls = list files (Linux)

dir = to view all files in a directory

cls = to clear screen (terminal

cd Name = to jump to a directory

Three types on environ
→ Text Editors
→ Full IDEs
→ Notebook

→ Text Editors
- General editors for any text file
- Work with variety of file types
- Can be customized with plugins & add-ons
- Most pop. → Sublime Text & Atom
- Not specifically for Python

→ Full IDE's
- Specifically for Python
- Larger programs

→ Comm. editions are free
→ Most pop. → PyCharm & Spyder

• Notebook Environment
→ see i/p & o/p next to each other
→ support on-line markdown notes, visualization videos & more
→ special for .ipynb

You can't just double click & open python notebook, you need to use Jupyter notebooks only

Types of Numbers

Eg Arithmetic Operations

Addn → + Division → /
Subtract → - Mult^n → *
Modulo → %

Floor Division: //
E.g: 7//4
⇒ 1

Powers: **
2**3 | 4**0.5
⇒ 8 | ⇒ 2

Variable Assignments

Reassignment is allowed in Python

E.g.

```
>>> a=5
>>> a+a
>>> $10
>>> a=10
>>> a
>>> 10
```

Rules for creating variable names

→ Names cannot start with a number
→ No spaces allowed, _ allowed
→ Can't use any symbols
→ Considered as best practice to keep names lower case (PEP8)
→ Avoid using 'l' (small), 'O' (capital), I (uppercase) as single character variable names
→ Avoid keywords

Name	Type	Description
Integers	int	3, 100 etc.
Float	float	3.5, 4.6
String	str	"hello", 'Sam', '2000', Ordered
Lists	list	[10, "hello", 200.3] (Ordered seq.)
Dictionary	dict	Unordered key:value pairs {"mykey": "value", "name": "Frankie"}

Name	Type	Description
Tuples	tuple	Ordered immutable seq. of ~~characters~~ objects (10, "hello", 200.3)
Sets	set	Unordered collection of unique objects {"a", "b"}
Booleans	bool	Logical value indicating True or False

Representation error:

It refers to the fact that decimal fractions cannot be represented exactly as binary fractions (base 2). This is the main reason why many languages won't display exact decimal rq. you expect:

```
>>> 0.1 + 0.2
0.30000000000004
```

Hence 0.1+0.2 - 0.3 is not equal to 0

Dynamic Typing

→ This means you can reassign variables to diff. data types.
→ This makes Python very flexible in assigning data types as compared to other languages (statically typed).

For e.g.

```
a = 2
a = ["abc", "bcd"]
```

This is okay in Python

Pros & Cons of Dynamic Typing

Pros

→ Very easy to work with
→ faster development time

Cons

→ may result in unexpected bugs
→ need to be aware of type()

type() function

This is used to check the type of object

```
a = 60
type(a)
```

⇒ int