**Python Command Prompt**

**Type cmd**

**Commands**

* **cd** list the current directory we are present in
* **dir** list the folders and files present in the current directory we are working in
* **cd <folder’s name>** to enter the folder present in the current working directory
* **cd ..** to move up one directory leave sub folder to go to folder
* **cls** clear screen

**Python Overview**

* Python was discovered by Guido Van Rossum in 1991
  + Easier to learn
  + Use indentation instead of parenthesis/ curly braces
  + Huge open source libraries available
  + Lot of libraries already included

**Environment**

* It is place where we write code. It can be of three types
  + Text Editors
  + Full Integrated Development Environments (IDEs)
  + Notebook Environments
* Text Editors
  + Can work with variety of files
  + Open files as a text
  + Not designed for python only (varieties of other languages too)
  + Plenty of plugins available
  + Famous ones are sublime editors and Atom editors by Github
* IDEs
  + Generally used by companies where a team of developer is working
  + Heavier in file size
  + Designed specifically for python so have a lot of functionality available
  + Famous ones are PyCharm and Spyder
* Notebook Environments
  + Great for learning
  + See input and output of code right next to each other as code runs in different blocks called cells where we can see input and output of each cell
  + Support in line notes, videos, images and a lot more
  + Special file formats that are not .py
  + Most famous is Jupyter Notebook for python

**Python data types**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Integers | int | Whole numbers, such as:    **3**     **300**    **200** |
| Floating point | float | Numbers with a decimal point:**2.3     4.6   100.0** |
| Strings | str | Ordered sequence of characters:   **"hello"**   **'Sammy'   "2000"  "楽しい"** |
| Lists | list | Ordered sequence of objects:   **[10,"hello",200.3]** |
| Dictionaries | dict | Unordered Key:Value pairs:  **{"mykey" : "value" , "name" : "Frankie"}** |
| Tuples | tup | Ordered immutable sequence of objects: **(10,"hello",200.3)** |
| Sets | set | Unordered collection of unique objects:  **{"a","b"}** |
| Booleans | bool | Logical value indicating **True** or **False** |

**Data types and data structures**

* **Strings**
  + **Sequences of characters**, using the syntax of either single quotes and double quotes
  + “Hello! I don’t do that” will be typed fully
  + **Ordered sequences** so that we can use indexing and slicing to grab sub sections of the string
  + Indexing allows us to grab a single character from the string
  + Indexing starts at zero
  + H – (0,0), E – (1,-4), L – (2,-3), L – (3,-2), O – (4,-1)
  + <Variable\_name\_string>[-1] will always gives the last letter of the string
  + <Variable\_name\_string>[::-1] will print the string from end to the beginning
  + Slicing helps us in grasping a sub section of the **string [start: stop: step\_size]** where start is the index for the slice start, stop is the index it goes up to but does not include and step is the size of the jump you take.
  + Spaces are counted as characters in string
  + \t for tab and \n for new line are some of the escape sequences
  + Strings are immutable which means they don’t support character assignment, i.e., one character can’t be supported can’t be reassigned though we can reassign the whole string.
  + Anything that begins with the hashtag is commented code and does not run
  + String concatenation which means strings can be added manually
    - Name =’Pam’
    - Name2 = ‘My name is’ + Name
      * Name = My name is Pam
    - Letter = ‘z’
    - Letter\*4
      * ‘zzzz’
    - ‘2’ + ‘2’
      * ‘22’
  + <String\_name>.<Tab\_key> will open a list of functions like name.upper() will print ‘PAM’ all in upper case , also remember parenthesis. Similarly Name2.split() will split the string in lists based on whitespaces
  + String interpolation where string can be used to print different variable. This can be done by two methods
    - .format() methods
      * print(‘This is a string {}’.format(‘Inserted’))
    - f-strings (formatted string literals)

**Lists**

* Lists are ordered sequences that can hold a variety of object types
* They use [] and commas to separate objects in the list
  + - * + [1,2,3,’Hello’,5]
      * Lists supports indexing and slicing.
      * Lists can also be nested.

**Operators**

**+** - Addition

**-** - Subtraction

**\* -** Multiplication

**/ -** Division

**% -** Modulo operator, gives back the remainder

**\*\* -** power operator gives back the power

**Features of Python**

* Python use **dynamic typing** which means a same variable name can be reassigned to different data types in a single program.

**Functions of Python**

* type(variable\_name) :- shows the type of variable type i.e, int or float
* print(variable\_name)- prints output which can be a variable/ string/ anything
* len(string or variable\_name) - will give the length of the string