

Git and Git Hub

Afsaan, Learbay

- 1) Git and Git hub comes under VCS-Version Control System.
- 2) VS Code is 98 percent Company for Development.
- 3) Jupyter Notebook is starting.
- 4) BitBucket.
- 5) Google Collab.

VS Code is for any coding.

Git and Git Hub

Afsaan,Learbay

Command Line basics:

- 1) ls(List) is command used for list in that directory.
- 2) ls -a (List) list with hidden files. 3) pwd(Present working directory) is present working directory.
- 4) cd change Directory is present working directory.
- 5) Ctrl+C for out of loop
- 6) Tab for help in selecting file or folder
- 7) mkdir for make directory.
- 8) rmdir for removing directory, it should be empty.
- 9) rm -r for deleting any directory irrespective of empty or some file.
rm -r path file.extension
- 10) Create file by touch command, ex- touch shivam.txt
if folder name has space then syntax is cd 'Shivam Asthana'

Git and Git Hub

Afsaan, Learbay

Git basics:

```
git config --global user.name "Shivam123iit"
```

```
git config --global user.email "shivam.asthana123@gmail.com"
```

```
git config --global --list
```

```
git clone https://github.com/shivam123iit/learnbay_afsaan.git
```

```
git config --global https.sslverify false
```

```
git status
```

```
git add
```

```
git commit My file.pdf -m 'My first commit'
```

```
git push origin master
```

```
git add . Adds all files to staging area
```

```
git log – To get Log of all commits
```

```
git diff – Will tell about diff in files
```

Python Components

Afsaan, Learnbay

- 1) Literals
- 2) Constants
- 3) Variables
- 4) Identifier
- 5) Reserve Words
- 6) Statement and Expression
- 7) Block and Indentation
- 8) Comments.

Jupyter NoteBook

Afsaan, Learnbay

Open Jupyter Notebook

>>Jupyter Notebook

LITERALS

Literals :Value that we use in Python is a literals, eg Integer, float, etc.

Integer 10, 2, 9999 etc.

float 10.2, 2.3, 4.35 etc.

Complex number : Anything which can be written in form of $a+bj$.

None

Boolean : True and False.

String : Any thing in single Quote '....', Double Quote "...." and triple Single Quote '''.....''' and triple double quotes """ """

Examples:

`a=10`, `a` is a variable, `=` is a operator and `10` is a Literal. `b= True` , this is boolean.

`c= None` , this is None

`d= 'Shivam'` , this is a String Literal.

Identifiers

Identifiers: Any Name which we give in Python is a Identifier , this can be variable name, Constant Name, function Name and Class Name

Rules:

- 1) Anything A-Z, a-z, 0-9 and _
- 2) Special Characters are not allowed.
- 3) It should not start with Number.

Ex1 a= 9

```
print(a)
```

```
>>9
```

Ex2 12=9

```
print(12)
```

```
>> Error
```

Ex3 _12=3

```
print(_12)
```

```
>>3
```

Keyword

Keywords should not be used import keyword

```
print(keyword.kwlist)
```

```
['False', 'None', 'True', '__peg_parser__', 'and', 'as', 'assert', 'async',  
'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except',  
'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda',  
'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with',  
'yield']
```

>> You should never start and end variable with __

Expression and Statements:

Expression: Gives Output -- > 10+20, 'Data'+'Science'

Statement: No Output -- > a=20, for i in range(4)

Block and Indentations

```
"""
```

```
Stat1
```

```
Stat2
```

```
Stat3
```

```
Stat4
```

```
"""  """
```

```
Stat1
```

```
Stat2
```

```
if Condition:
```

```
    Stat3
```

```
    Stat4
```

```
Stat5
```

```
Stat6
```

```
Stat7
```

```
Stat8
```

```
"""
```

Standard we give 4 space which is one Tab

Comments in python

```
pi =3.14      # value of Pi  
r =5          # radius of circle  
a =pi by 4*r*r      # Area of circle
```

Data Types

Basic Data types

Integers->Without decimal point, **function type(a)**

float->With decimal point

complex->With imaginary part, $a+ib$

string->Without decimal point

None

Boolean

Derived Data types

List

Tuple

set

dic

range

byte

frozen set

Comparison Operator

More than $a > b$

Less than $a < b$

More than equal to $a \geq b$

Less than equal to $a \leq b$

Return is always True or False

It can perform int to float

It can perform str to str

complex and None data type do not support

Equality operator $==$

It gives True and False only

Logical Operators

AND
OR
NOT