

CONFIDENTIAL &amp; RESTRICTED

# Welcome to Python

Gaurav Gupta  
tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Setup and Workspace

- Installing Python
- Etherpad
- Testing Installation : The Interactive shell
- Tools for working environment
- Creating workspace: Directory structure
- Windows and Linux Command line
- Some shortcut keys

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Installing Python

---

Python available at the official website : <https://www.python.org/>

- Windows : Download the executable and run it.
- Linux : Run the command on Ubuntu shell  
`sudo apt-get install python3`

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Oh!! Did I Introduce you to Etherpad

---

Etherpad is a shared notepad available at the following link.

[https://etherpad.net/p/py\\_learnbay](https://etherpad.net/p/py_learnbay)

Consider it as your friend, you get to know why soon.

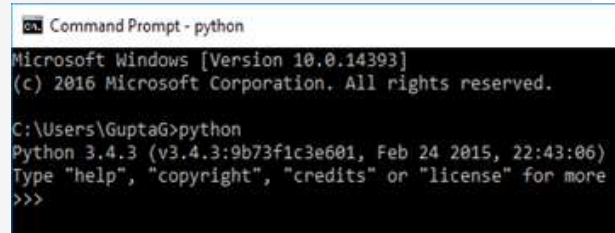
tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Testing Installation : The Interactive shell

- Windows : Press Windows Key and type cmd. On the Terminal type python



```
Command Prompt - python
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\GuptaG>python
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06)
Type "help", "copyright", "credits" or "license" for more
>>>
```

- Linux : Open a terminal (Ubuntu CTRL+ALT+T) and type python.

\*\* if you get error like command not found, add python installation path

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Tools for working environment

- Use an IDE

**Pycharm** IDE with Python 3.x.x

<https://www.jetbrains.com/pycharm/download/>

- Use any text editor and Command line (my preferred way)

Write Scripts using a text editor : Notepad++, vi, vim, Sublime Text..

Windows or Linux command line for executing.

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

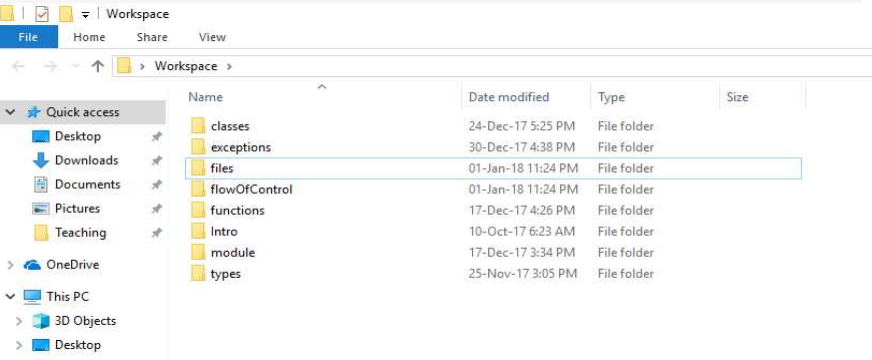
Creating workspace: Directory structure

- Create a folder **workspace** : all our scripts will be in this folder
- Maintain separate folders for each topic in **workspace** folder.
- Make sure to name the script files in following convention: **fN\_topic.py**

Ex:

f1\_ifStatement.py

f2\_ifElse.py



tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

Windows and Linux Command line

	Windows	Linux
Go to the folder	<code>cd &lt;folder Name&gt;</code> Ex: <code>cd Workspace</code>	<code>cd &lt;folder Name&gt;</code> Ex: <code>cd Workspace</code>
Go to the previous directory	<code>cd ..</code>	<code>cd ..</code>
List files in current directory	<code>dir</code>	<code>ls</code> <code>ls -la</code>

Use up and down arrow keys to view previous commands in cmd window

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Notepad++ Shortcuts

Ctrl + a	To select everything in current file
Ctrl + s	Save current file
Ctrl + Tab	To switch files
Ctrl + n	To open new file
Ctrl + c	To copy selected text
Ctrl + v	To paste selected text

- Press *Shift* and Arrow keys to make selection of a part of text (you can use Ctrl key while selecting to make selection faster)

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Windows shortcuts

### Open command window in current folder

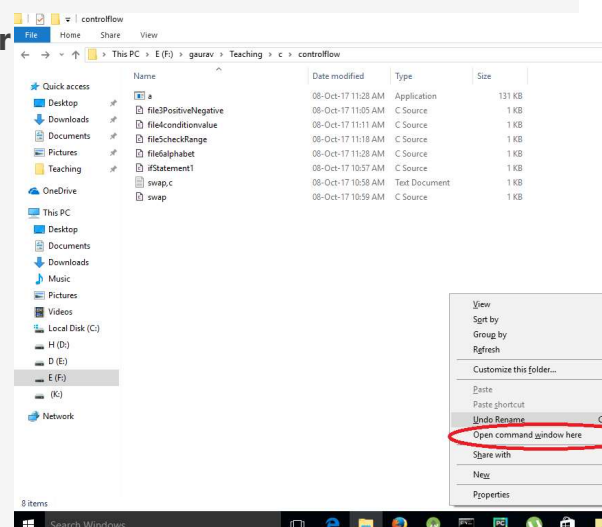
Press the Shift Key and right click

You will see the option :

*Open Command Window Here*

### To Switch Tabs/Windows

Alt + Tab      and      Alt + Shift + Tab



tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Python Kickstart

- Using Interpreter and a Script
- Intro to print function
- Dir and help functions

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Using Interpreter

---

- Open cmd window and type:  
 $1 + 2$
- Create a python script and type the same thing there.  
Save at **f1.py**
- Now run from the command line as:

**python f1.py**      ***#before doing this just check version of python***

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Intro to print function

---

- In a python script type:  
`print( 1 + 2 )`  
now save it and run again.
- Now try working with variables.
- Printing multiple values from single print function
- And yes **PRINT IS A FUNCTION**

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Creating a Variable: Dir And Help functions

---

- Create a variable in the current scope and check what all things are available there
- **Dir** gives the list of available attributes and objects in the current scope or of the object if passed and argument.
- **Help** method returns help information, depending on how it is invoked.
- Help can be called without argument, with the names of builtins, or with names specified as a string

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Python Syntax ,Keywords and Operators

- Tokens : building blocks
- Python Comments
- Print Method
- Input()
- Type() and basic types in python
- Conversion Between Types

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

### Tokens : building blocks

---

- Smallest individual components that make up a program.
- 4 Types :
  - Keywords
  - Identifiers
  - Operators
  - Literals

tuteur.py@gmail.com



CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Keywords

- Special reserved words predefined or reserved by the language.

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

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Identifiers

- **Identifiers** can be a combination of letters in lowercase (**a to z**) or uppercase (**A to Z**) or digits (**0 to 9**) or an underscore (**\_**)
- Variable names, class names, function names and module names are all identifiers.
- Some special identifiers in Python :

`__*__` : Special Reserved system defined names

`__*` : Used to define private class members

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Operators

---

- `+, -, *, /, >, <, =, <=, >=, ==, !=, >>, <<, &, |, ~, ^`
- `+=, -=, *=, /=, =`
- `() , [] , {}`

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Literals

---

These are just constant values:

integer	:	1, -1, 0, ...
Floating	:	-1.0, 0.0, 3.14
string	:	<code>'', ''', 'a', 'abcd'</code>
Boolean	:	True, False

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## String Dilemma

---

- Single, Double or Triple Quotes??
- 'Quoted String' "Quoted String" """ Quoted String""" ''' Quoted String'''
- Single quote can be used in double quoted string and vice versa:
  - 'single ' in single ' ; "double " in double" : **Wrong**
  - 'double " in single' ; "single ' in double" : **Right**
- """ Multi Line string"""

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Comments

---

- **Single line** comments start with #.  
*# This is a single line comment in python*
- **Multi line** comments can use the triple quote syntax.  

```

"""
This is a multi line
comment in python.
"""

```

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Print Function

---

- Print method prints to the standard output
- Syntax:  

```
print(<var/const>, ..., sep= '<separator>', end = '<delimiter>', file = <file object>)
```

**sep**, **file** and **end**, arguments are optional and should appear in the end.
- Escape Sequences : **\n** and **\t**

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Type Method

---

- Syntax:  

```
type(<object argument>)
```
- Returns the type of the argument
- Argument might be variables, objects ....
- Some basic types are:  
int, float, string, bool, complex

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Converting Between types

- `int(<string>), int(<int>), int(<float>)`      # converts string containing digits to int
- `str(<int/float/....>)`      # converts any type to its string representation

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Input()

- The input method returns the value entered by user as a string
- Also allows to specify a string argument for a message to displayed

```
1 x = input('Enter one Number')
2 x = int(x)
3 y = x*x
4 print("Square of " + str(x) + " is %d" % y )
```

tuteur.py@gmail.com

CONFIDENTIAL &amp; RESTRICTED

Gaurav Gupta

## Conversion Between Types

---

- String to **Int** : `int(<string variable/constant>)`
- String to **float** : `float(<string variable /constant >)`
- Any Type to **String** : `str(<variable /constant >)`
- **bin()** method returns the binary representation of an **integer**

tuteur.py@gmail.com