Python Programming

CS2001

Lecture-1 Introduction & Getting started

Autumn Sem 2024-25

Shivang Tripathi

Overview

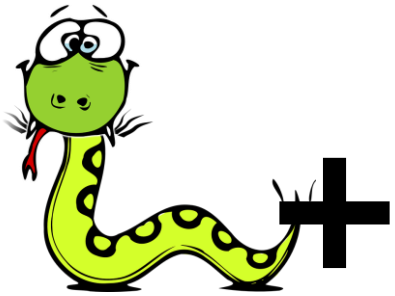
Introduction to Python

Course Syllabus & Plan

Assessment Criteria

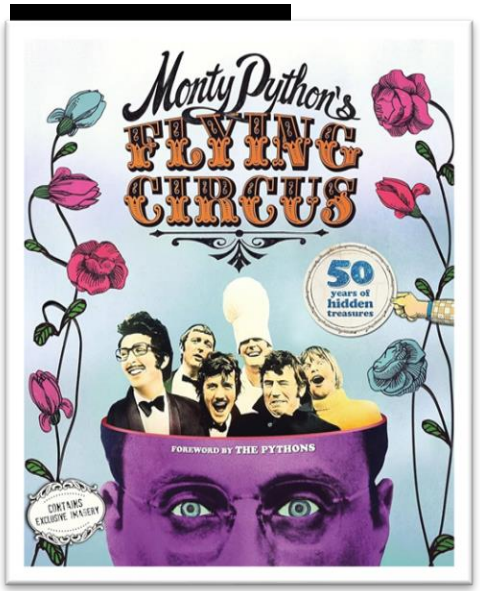
Installation, Running & tools introduction (with CS2101)

Additional Reading Resources, Tutorials



Python

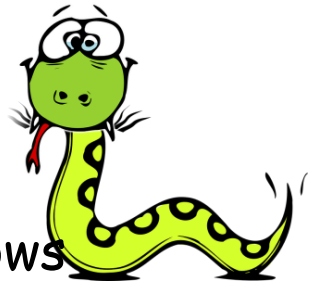
- Developed by Dutch scientist Guido van Rossum in early 90s
- Named after a TV show 'Monty Python's Flying Circus'
- Python is
  - High-level (human readable),
  - Object oriented,
  - open source,
  - portable,
  - scripting language, and much more…



Python features

Python requires interpreter

- High-level
  - Needs compiler or <u>interpreter</u> for translation into machine readable code
  - What's Low-level Language ?? (Hint: Assembly Language)
- Interpreted language
  - Interpreter reads and execute one statement at a time
  - What does compiler do ?? (Compile into object code)
- Open-source
  - Source code is available for you to explore, modify…
  - Examples of other popular open-source codes ??
- Portable
  - Take your friend's code written on Linux machine and run it on your Windows



Python features

- Object Oriented
  - Code is organized around *objects* created from *classes*. These objects encapsulate data (attributes) and functions (methods) that operate on the data.
  - Python supports key OOP concepts like:
    - Inheritance (creating new classes from existing ones),
    - Encapsulation (hiding internal details),
    - Polymorphism (different objects can be treated as instances of the same class through a common interface), and
    - Abstraction (simplifying complex reality by modeling classes appropriate to the problem)
- Dynamic Typing
  - No need to declare variable before it is used
- Auto memory management

Python Applications

widely used across various domains due to its simplicity and versatility

- Data Science & Analytics
- Automation & Scripting
- Web Development
- AI/ML
- Game Development
- Scientific Computing
- Cyber security
- And many more…

CS2001 Syllabus

Unit 0 : Introduction, Installation & Getting Started

Unit I : Python Basics – Data types & operators

Unit II : Control Statements in Python

Unit III : List, Tuple, Sets and Dictionary

Unit IV : Functions and Modules in Python

Unit V : File Management

Unit VI : Matplotlib, numpy and pandas

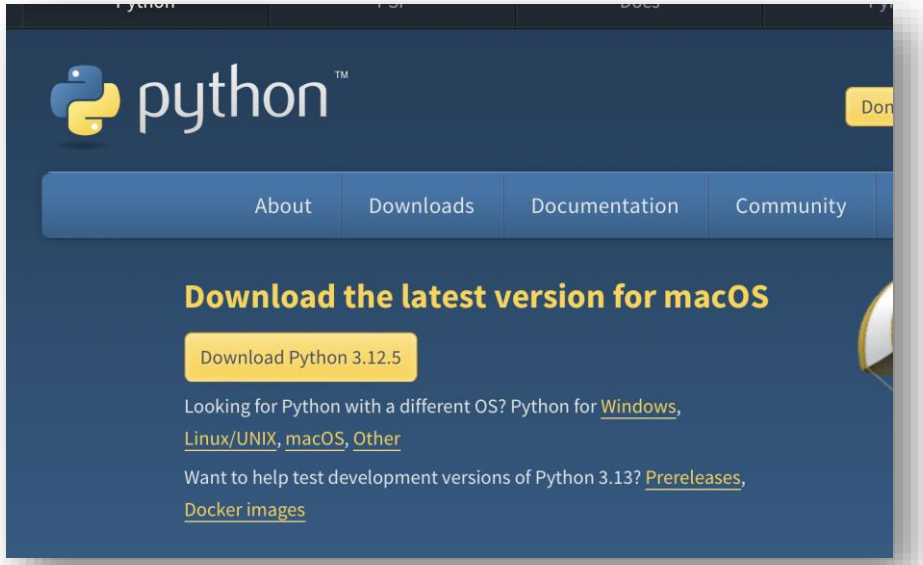
Unit VII : Machine Learning and IoT using Python

Assessment Criteria (CS2001, CS2101 (lab))

OUT of 100 marks

- 50 for End Sem Exam, 30 for Mid Sem Exam
- 10 marks for Assignment/Quiz/Mini-project (Lab)
- 5 marks for attendance (If total attendance is >= 75%)
- 5 marks daily activities

Python Installation



Windows:

- Go to https://www.python.org/downloads/
- Download latest version
- Run the .exe file and tick "**Add to the Path**" dialog box and install
- Open Windows terminal or Command Prompt and type python or python3 and enter
- IDLE Integrated Development Environment (IDE) is also installed (check all apps)

MacOS/Linux:

- Generally pre-installed. Check version on terminal with python --version

IDLE

Interactive mode:

- Write code line by line and execute in python IDLE shell or on terminal

IDLE Shell 3.12.5

```
Python 3.12.5 (v3.12.5:ff3bc82f7c9, Aug  7 2024, 05:32:06) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 3
>>> b = 5
>>> a + b
8
>>> c = a**b
>>> print(c)
243
>>> c
243
>>> exit()
```

Scripting mode:

- Write complete code in editor of your choice or IDLE's File menu -> New File, save it and run (F5)
- On terminal type: python file_name.py
- Note: save the file with .py extension if coding on editor of your choice like vscode, notepad, sublime, vim, etc...

IDLE Shell 3.12.5

```
Python 3.12.5 (v3.12.5:ff3bc82f7c9, Aug  7 2024, 05:32:06) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 3
>>> b = 5
>>> a + b
8
>>> c = a**b
>>> print(c)
243
>>> c
243
>>>
================== RESTART: /Users/shivang/Documents/test.py ==================
Hi this is CS2001 class, Welcome to new sem
>>>
```

test.py

```
print("Hi this is CS2001 class, Welcome to new sem")
```

Ln: 2 Col: 0

Ln: 15 Col: 0

Running interactively on terminal (win, linux, Mac)


```
% python  or  python3  or python3.x
>>> 1+1
2
>>>
```

Note: In 3.x, the 'x' denotes the major version revision
E.g.: latest version is 3.12

Python prompts with '>>>'

Exit python:

Win: Ctrl-Z + enter or exit()

Linux/MacOS: Ctrl-D

Python module can also be imported in another python file

Make a python file directly executable by Adding the appropriate path to your python interpreter as the first line of your file

```
#!/usr/bin/python
```

Making the file executable

```
% chmod a+x file_name.py
```

Invoking file from terminal:

```
% file_name.py
```

Running interactively on Jupyter Notebook

- Jupyter notebook is server-client application
- Wonderful tool to learn & debug scripting languages

Installation: (in windows terminal/powershell)

- `pip3 install notebook`
- Make sure the installation PATH is added to the Env variable

- Alternatively, included with Anaconda
  - Anaconda is another package, easy to install across platforms
  - A bit heavy though
  - Can be used to install python, jupyter, and several other tools/packages used for scientific computing & data science

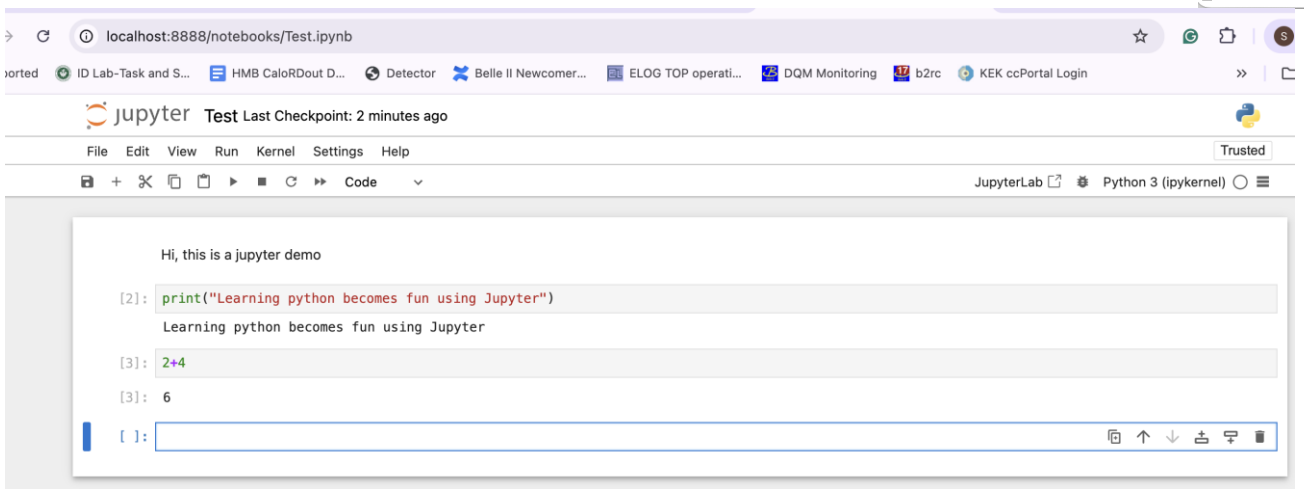
Jupyter Notebook

If the path is added correctly, on terminal type:

- `jupyter-notebook`

bin — jupyter-notebook — 109×41

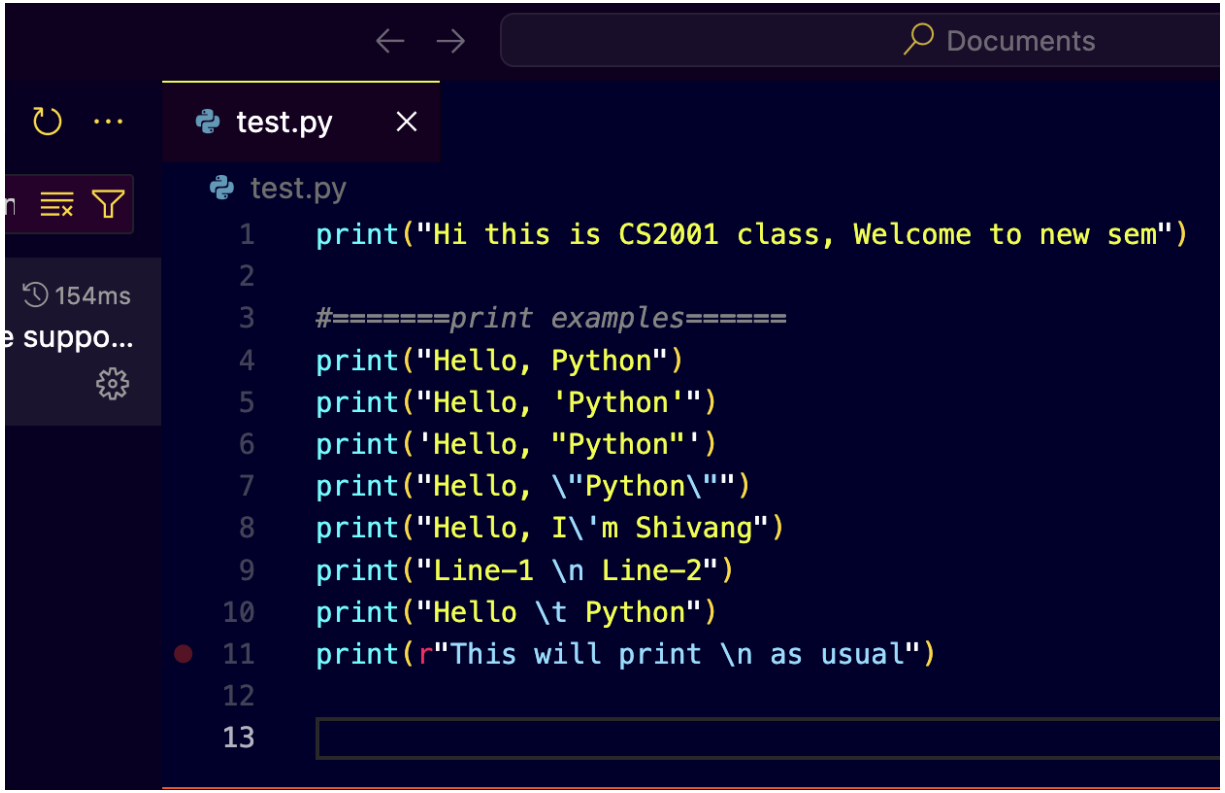
```
shivang@Shivangs-MacBook-Pro bin % jupyter-notebook
/Users/shivang/Library/Python/3.9/lib/python/site-packages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3
 v2 only supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. See: https://
github.com/urllib3/urllib3/issues/3020
  warnings.warn(
[I 2024-08-09 11:38:49.964 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2024-08-09 11:38:49.967 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2024-08-09 11:38:49.970 ServerApp] jupyterlab | extension was successfully linked.
[I 2024-08-09 11:38:49.973 ServerApp] notebook | extension was successfully linked.
[I 2024-08-09 11:38:50.196 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-08-09 11:38:50.223 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-08-09 11:38:50.225 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-08-09 11:38:50.226 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-08-09 11:38:50.227 LabApp] JupyterLab extension loaded from /Users/shivang/Library/Python/3.9/lib/pyt
hon/site-packages/jupyterlab
[I 2024-08-09 11:38:50.227 LabApp] JupyterLab application directory is /Users/shivang/Library/Python/3.9/shar
e/jupyter/lab
[I 2024-08-09 11:38:50.228 LabApp] Extension Manager is 'pypi'.
[I 2024-08-09 11:38:50.238 ServerApp] jupyterlab | extension was successfully loaded.
[I 2024-08-09 11:38:50.241 ServerApp] notebook | extension was successfully loaded.
[I 2024-08-09 11:38:50.242 ServerApp] Serving notebooks from local directory: /usr/local/bin
[I 2024-08-09 11:38:50.242 ServerApp] Jupyter Server 2.14.2 is running at:
[I 2024-08-09 11:38:50.242 ServerApp] http://localhost:8888/tree?token=22e89a4811f8d2567dcba98776cf502d3f530b
7912908691
[I 2024-08-09 11:38:50.242 ServerApp]     http://127.0.0.1:8888/tree?token=22e89a4811f8d2567dcba98776cf502d3f
530b7912908691
[I 2024-08-09 11:38:50.242 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to s
kip confirmation).
[C 2024-08-09 11:38:50.247 ServerApp]

    To access the server, open this file in a browser:
        file:///Users/shivang/Library/Jupyter/runtime/jpserver-8566-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/tree?token=22e89a4811f8d2567dcba98776cf502d3f530b7912908691
        http://127.0.0.1:8888/tree?token=22e89a4811f8d2567dcba98776cf502d3f530b7912908691
[I 2024-08-09 11:38:50.258 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-langu
age-server-nodejs, javascript-typescript-langserver, jedi-language-server, julia-language-server, pyright, py
thon-language-server, python-lsp-server, r-languageserver, sql-language-server, texlab, typescript-language-s
erver, unified-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-la
nguageserver-bin, yaml-language-server
```

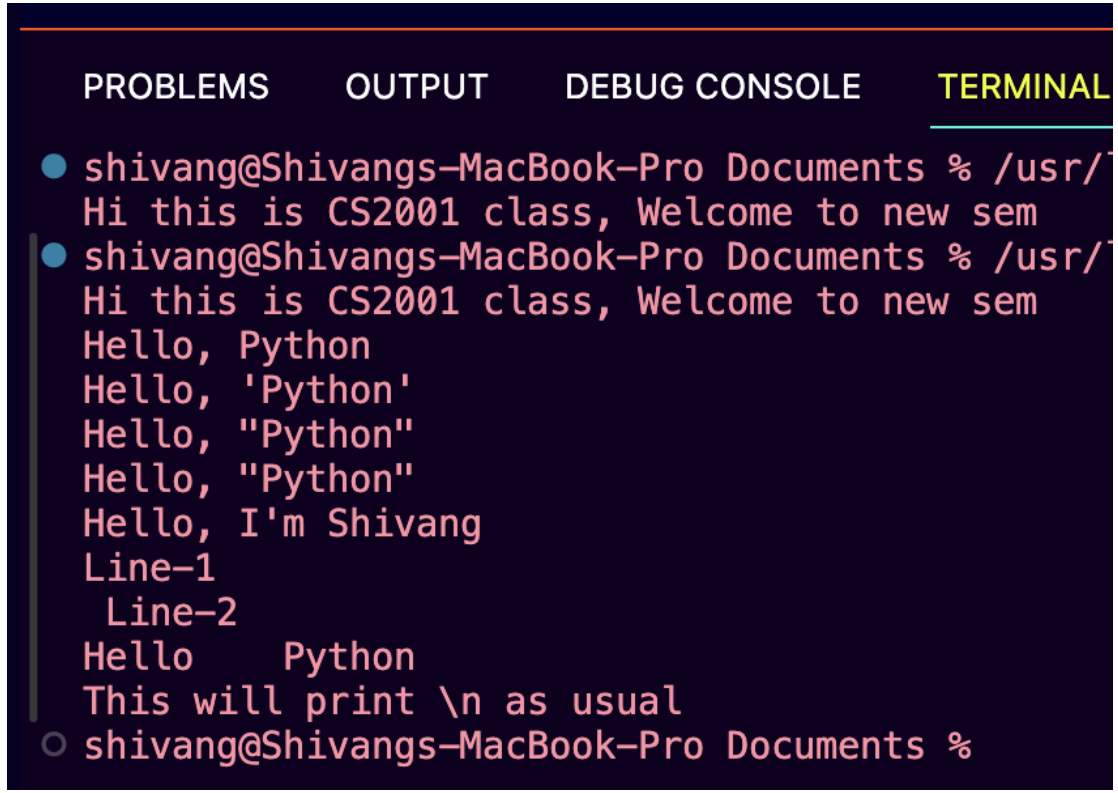


Visual Studio Code

A very popular, open-source code editor by Microsoft

- Rich features, light weight, customizable
- Extension for multiple programming languages





Git & Github

- A version control tool
- Helps in keeping track of changes in code & Great for Collaboration
- Install: https://git-scm.com/downloads
- Create an account on github
  - GitHub makes tools that use Git.
  - GitHub is the largest host (repository) of source code in the world
  - Tutorial & Setup:
    - https://www.w3schools.com/git/default.asp?remote=github

Python Starters

Python Starters: *Observe the code below*

[7]:
```
# This is one line Comment
''' This
is a
multi-line
Comment
'''

a = 2
A = 4

""" what would be
the value of
a and A ??? """

print("The value of a is", a)
print("The value of A is", A)
```

```
The value of a is 2
The value of A is 4
```

- Python is case sensitive
- # indicate single line comment
- Triple quotes """ or ''' for multi-line
- Explicit line Continuation '\' (Back-slash)
- Implicit line Cont. [ ], ( ), { }

```
>>> 1+1\
... +2
4
>>> list = ['item1', 'item2',
... 'item3']
```

Multiple statement in single line using ;

```
>>> a=1;b=2; print(a+b)
```

Python Starters

- A code block in python uses "indentation"

```
[1]: a = 2
     b = 4

     if a<b:
         print(a, 'is less than', b)
     else:
         print(a, 'is not less than', b)

     2 is less than 4
```

- A code block in C uses {...}

```
while(TRUE)
 {
   do this;
 }
```

- Standard practice is to use 4 white spaces
- You may also use >1 white spaces but not preferred
- 'IDE's are recommended for auto indentation

Python Starters

- Assignment of data to a variable:
  - In C/C++ datatype of the variable is declared first (int, float, …)
  - assignment to data first creates a place in memory and then stores
  - In python, think it of as a 'tag' with no type of its own
    - But the data (to which tag is attached) has a "type"
    - Hence, you could make more than one assignment to the same variable

```
[6]: a = 2   # a "points" to 2
     b = a   # now b also "points" to 2

     print(a)
     print(b)

     print("=====")
     b = 5   # now b points to 5
     print (a)
     print(b)

     2
     2
     =====
     2
     5
```

Create a variable without assigning a value. What's the O/P ??

Python Starters

– Assignment of data to a multiple variable at the same time:

```
>>> x, y = 2, 3
>>> x = y = z = 2
>>> x, y = y, x
```

- "=" is Assignment, '==' is comparison
- For numbers **+ - * / %** are as expected
  - Special use for **+** for string concatenation and **%** for string formatting (old method)
- The basic printing command is **`print( )`**

Python Starters

- **Iterable:** collection that can be iterated over, using a loop
- **Ordered Collection:** elements are stored in the same order in which they are inserted. They can be accessed using an "index"
- **Unordered Collection:** elements are not stored in the same order in which they are inserted. So, can't access them based on "index"
- **Immutable:** unchangeable collection of items
- **Mutable:** changeable collection
- Types in python and their features: (will see types in detail in next chapter)
    - String : ordered, immutable, iterable
    - List : ordered, mutable, iterable
    - Tuple : ordered, immutable, iterable
    - Set : unordered, mutable, iterable
    - Dictionary : unordered, mutable, iterable

Additional Learning Resources

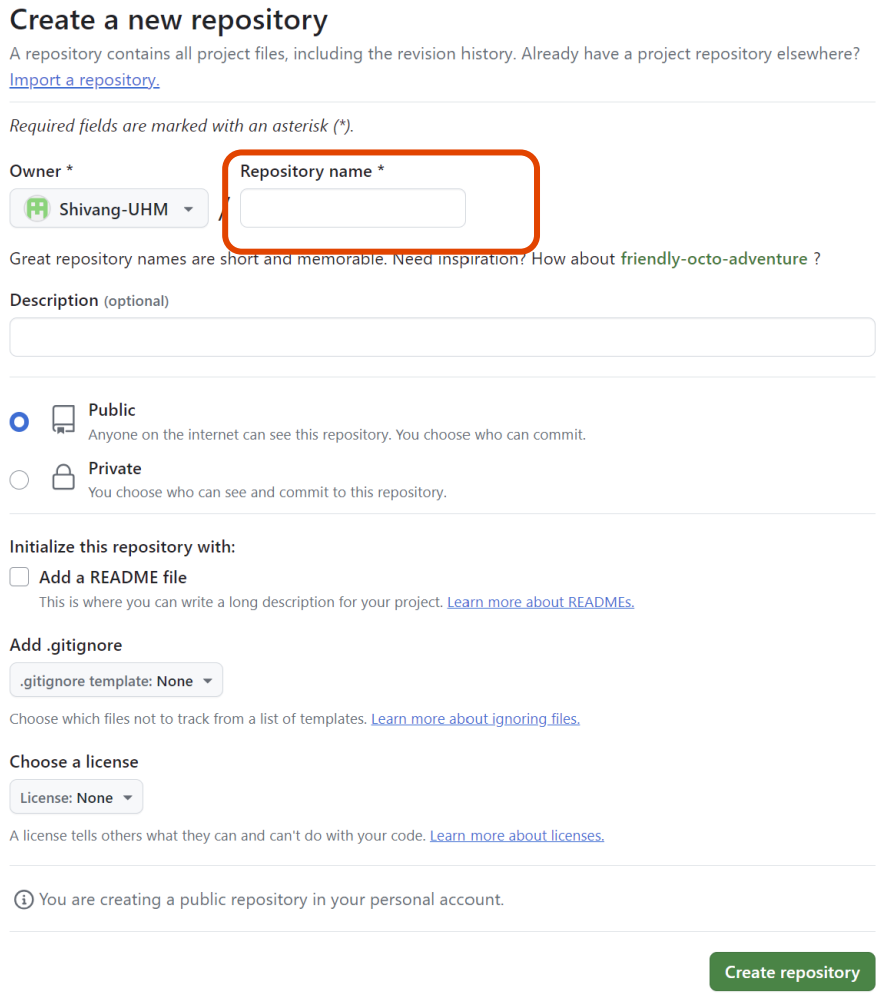
- Books:
  - Let us python (kanitkar & Son)
  - Python Programming (Anurag Gupta & GP Biswas) [McGraw Hill]
- Official python documentation & tutorial
- 3.12.5 Documentation (python.org)
- Online python Compiler : Online Python Compiler - Python Examples
- Interesting e-book: Automate the Boring Stuff with Python
- Learn Python in Y Minutes (learnxinyminutes.com)
- Python Cheat sheet: Python Crash Course by ehmatthes
- Infinite tutorials online

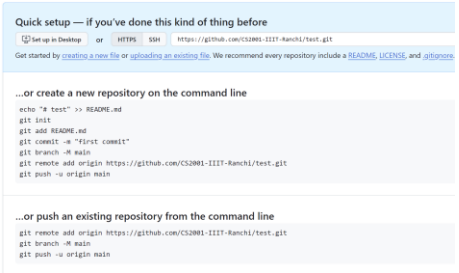
Github setup

- Git & Github are different
- Github provides platform for code repository and uses Git
- Create account/Sign up on Github https://github.com/
- Open Terminal:
  - Configure your github account
  - **`git config --global user.email ABC@iiitranchi.ac.in`**
  - **`git config --global user.name "Your github_user_Name"`**
  - It will redirect to login page ->Login to your github account
  - Create or go to folder for which you wanna push to repository
- Now, Initialize git using:
  - **`git init`**

Github setup

- Go to github website -> from your dashboard
- Create New repository
  - Write a name for your repo
  - Write Description
  - Choose to make it public or private
  - Add a readme file
  - New repo will be created at:
  - https://github.com/Owner-name/Repo-name.git





Github setup

- Next, push your local repo (PC) to the remote (github) repository
- Go back to the terminal -> Go to same directory where you did "git init"
- **`git remote add origin`** https://github.com/owner/repo-name.git
- Check status-> `git status`
- Create some files, e.g. test.py
- Check status-> `git status`

```
PS C:\Users\shiva\Documents\github_test> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        test.py

nothing added to commit but untracked files present (use "git add" to track)
```

Github setup

- Now Git is aware of the file, but has not added it to our repository!
- Files in your Git repository folder can be in one of 2 states:
  - Tracked - files that Git knows about and are added to the repository
  - Untracked - files that are in your working directory, but not added to the repository
- When you first add files to an empty repository, they are all untracked. To get Git to track them, you need to stage them:
  - **`git add test.py`**
- You could add more than one file
  - **`git add *`**
  - **`git add --all`**

```
PS C:\Users\shiva\Documents\github_test> git add .\test.py
PS C:\Users\shiva\Documents\github_test> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   test.py

PS C:\Users\shiva\Documents\github_test>
```

Github setup

- Git commit
  - Commits keep track of our progress and changes as we work.
  - It's a point where you can go back to if you find a bug, or want to make a change.
  - When we commit, we should always include a message.
  - By adding clear messages to each commit, it is easy for yourself (and others) to see what has changed and when.

```
PS C:\Users\shiva\Documents\github_test> git commit -m "First commit"
[main fdcbe4a] First commit
 1 file changed, 17 insertions(+)
 create mode 100644 test.py
PS C:\Users\shiva\Documents\github_test>
```

Push your local stuffs to the remote repo:

- git push --set-upstream origin master

```
PS C:\Users\shiva\Documents\github_test> git push --set-upstream origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 22 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 442 bytes | 442.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/CS2001-IIIT-Ranchi/test.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\shiva\Documents\github_test> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```