



**2nd Semester B. Tech**  
**CSS 1022 Data Visualization [1 0 3 2]**  
**LABORATORY MANUAL**

**Course Coordinator: Dr. Deepak Parashar**

School of Computer Engineering

Manipal Institute of Technology, Manipal, India

<b>Section</b>	<b>Faculty Instructor</b>
	Dr. Deepak Parashar
	Dr. Vidya Rao
	Dr. Divya S
	Dr. Raghavendra S
	Dr. Kamraj
	Dr. Rani Ooman Panicker
	Dr. Arpan Mandal
	Dr. Lella Kranthi Kumar
	Dr. Shankar Biradar
	Dr. N Siva Selvan
	Dr. Namrata Marium Chacko
	Mr. Mayur Anand Pandya
	Dr. Sameena Begum Pathan
	Dr. Venkatesh Bhandage
	Dr. Sivakumar
	Dr. Sivakumar

**JANUARY 2026**

## Table of Contents

<b>1. Course Outcomes (Cos).....</b>	<b>3</b>
<b>2. Assessment Plan .....</b>	<b>3</b>
<b>3. General Instructions .....</b>	<b>4</b>
<b>4. Software and Tools.....</b>	<b>5</b>
<b>5. Introduction to Anaconda, Jupyter Notebook5 .....</b>	<b>5</b>
<b>6. Python Basics .....</b>	<b>9</b>
<b>7. List of Experiments .....</b>	<b>10</b>
<b>8. References .....</b>	<b>13</b>

## 1. COURSE OUTCOMES (COS)

<b>At the end of this course, the student should be able to:</b>		<b>No. of Contact Hours</b>	<b>Marks</b>
<b>CO1</b>	Demonstrate ability to program in Python using built-in data structures	6	15
<b>CO2</b>	Perform vectorized computation with Pandas and NumPy	9	20
<b>CO3</b>	Implement wrangling, aggregation and summarisation of data	6	15
<b>CO4</b>	Develop insightful visualizations using Matplotlib and Seaborn	9	25
<b>CO5</b>	Apply data summarization and visualization techniques to write reports.	6	25
<b>Total</b>		<b>36</b>	<b>100</b>

## 2. ASSESSMENT PLAN

<b>Components</b>	<b>Continuous Evaluation</b>	<b>End semester Examination</b>
<b>Duration</b>	3 Hours per week	120 Minutes
<b>Weightage</b>	60%	40%
<b>Pattern</b>	<ul style="list-style-type: none"> <li>• <b>4 Evaluations:</b> <math>4 * 10M = 40M</math> <ol style="list-style-type: none"> <li>1. Code submission</li> <li>2. Program execution check</li> </ol> </li> <li>• <b>2 quizzes:</b> <math>2 * 10M = 20M</math></li> </ul>	For the given dataset 1. Data manipulation,cleaning :10M 2. Data Visualization: 20M 3. Data Summary & interpretation of results: 10 M <b>Total : 40M</b>

### **3. GENERAL INSTRUCTIONS**

Step 1: Listen to the faculty demonstrations and instructions on the exercise and datasets.

Step 2: Download the weekly exercise and data sets (if any) from LMS.

Step 3: Create a Python notebook with the following documentation and coding conventions for the exercise.

1. *Write documentation first, the experiment's description, the student's name, the register number, and the date.*
2. *Put Imports at the top of the file.*
3. *Use four spaces per indentation level.*
4. *Limit all lines to a maximum of 79 characters.*
5. *Use blank lines to separate logical sections.*
6. *Add sufficient comment lines in complete sentences. Block comments are indented to the same level as that code. Inline comments wherever necessary.*
7. *Use meaningful names for variables, functions and constants.*
8. *Follow the snake case for naming variables and functions, capitalize the first alphabet of each word for classes, constants should be in upper case and follow the snake case convention.*

Step 4: On completion submit the python notebook in LMS. Ensure that you submit well within the deadline.

Step 5: Show the Python notebooks and results to the instructors during program execution check.

- If a student misses a lab class, he/she must ensure the experiment is completed.
- Questions for lab tests and examinations are not necessarily limited to the questions in the manual but may involve some variations and/or combinations of the questions.
- Since this is an introductory course on Python programming and data visualization, the students cannot use AI tools to generate code.
- Please do not copy code from others.

## 4. Software and Tools

**Programming language:** Python

**IDE:** Anaconda distribution with Jupyter Notebook

**Pre-installed in Anaconda:**

NumPy: For numerical computations.

Pandas: For data manipulation and analysis.

Matplotlib: For plotting and visualization.

**Additional Package:** Seaborn: For advanced visualization

## 5. Introduction to Anaconda, Jupyter Notebook

### 5.1 Anaconda

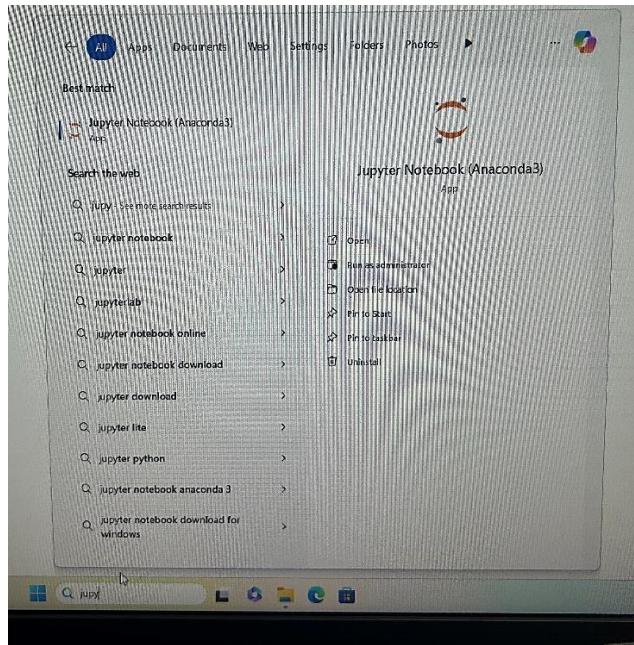
Anaconda is a open-source distribution for Python and R, specifically for data science. It can be used to create an isolated environment for making data intensive applications. It simplifies the process of managing packages and dependencies. Anaconda has over 1,500 packages, including essential tools like Jupyter Notebook, Pandas, NumPy, and Matplotlib, making it a one-stop solution for developers and researchers. Its package manager, Conda, allows seamless installation, updating, and management of software, ensuring compatibility and reducing conflicts.

### 5.2 Jupyter Notebook

Jupyter is a freely available web application that enables creation and sharing of documents containing equations, live coding, visualizations, and narrative text. Jupyter provides an interactive computing environment and it supports multiple programming languages, including Python, R, Julia. The major components of the Jupyter project is the notebook, a type of interactive document for code, text (including Markdown), data visualizations, and other output. The Jupyter notebook interacts with kernels, which are implementations of the Jupyter interactive computing protocol specific to different programming languages. Jupyter integrates data science libraries and frameworks, such as NumPy, Pandas, Matplotlib, sci-kit-learn, TensorFlow, and PyTorch. This allows users to leverage the full power of these tools within the notebook environment for tasks like data manipulation, visualization, machine learning, and deep learning.

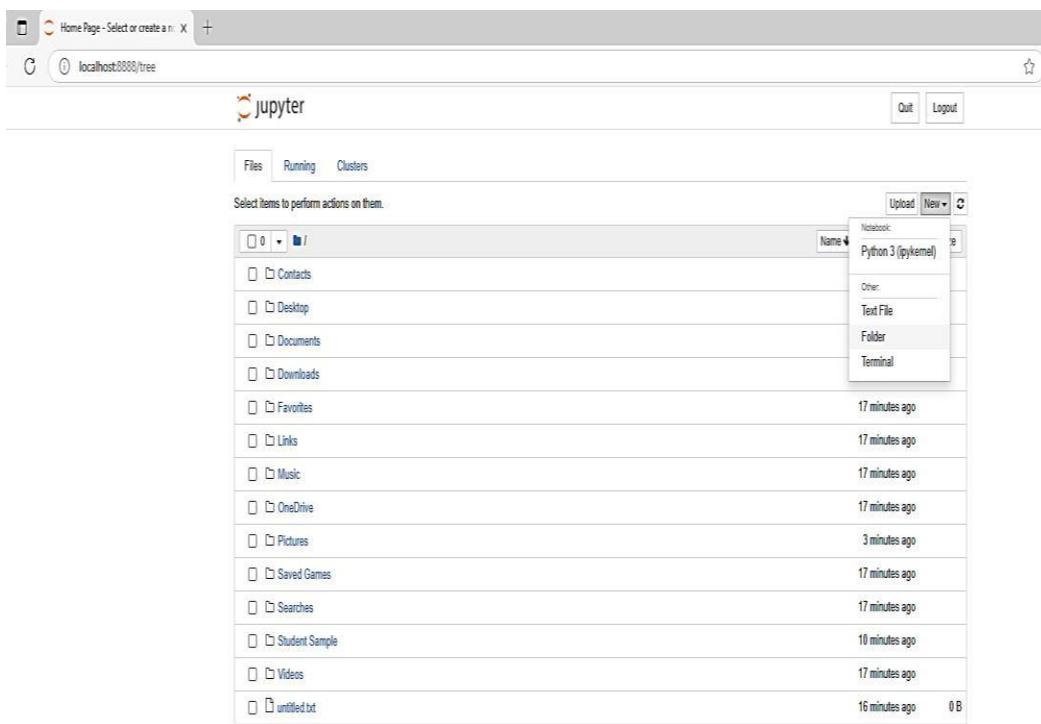
### **5.3. Getting started with Anaconda, Jupyter notebook on Windows**

**Step 1:** Type Jupyter notebook in the search bar as shown in Figure 1 and click on “Jupyter Notebook (Anaconda3).



**Fig 1:** Searching Jupyter application from the search bar.

**Step 2:** Change drive if required. Creating a new folder. Click on New-> Folder as shown in Figure 2.



**Fig 2:** Creating a new folder in Jupyter.

**Step 3:** Renaming the new folder. The new folder will be created with the default name “Untitled folder”. Click on rename and rename the folder appropriately as shown in Figure 3.

The screenshot shows the Jupyter Notebook interface on a Windows desktop. The window title is "jupyter". The main area displays a file tree with several standard folders like Contacts, Desktop, Documents, Downloads, Favorites, Links, Music, OneDrive, Pictures, Saved Games, Searches, Student Sample, and Videos. Below them is a folder named "Untitled Folder" which has a checked checkbox next to it. To the right of the file tree are three filter buttons: "Name", "Last Modified", and "File size". At the bottom right of the interface, there are "Upload" and "New" buttons. The status bar at the bottom shows the URL "localhost:8888/tree/Student\_Sample" and some system icons.

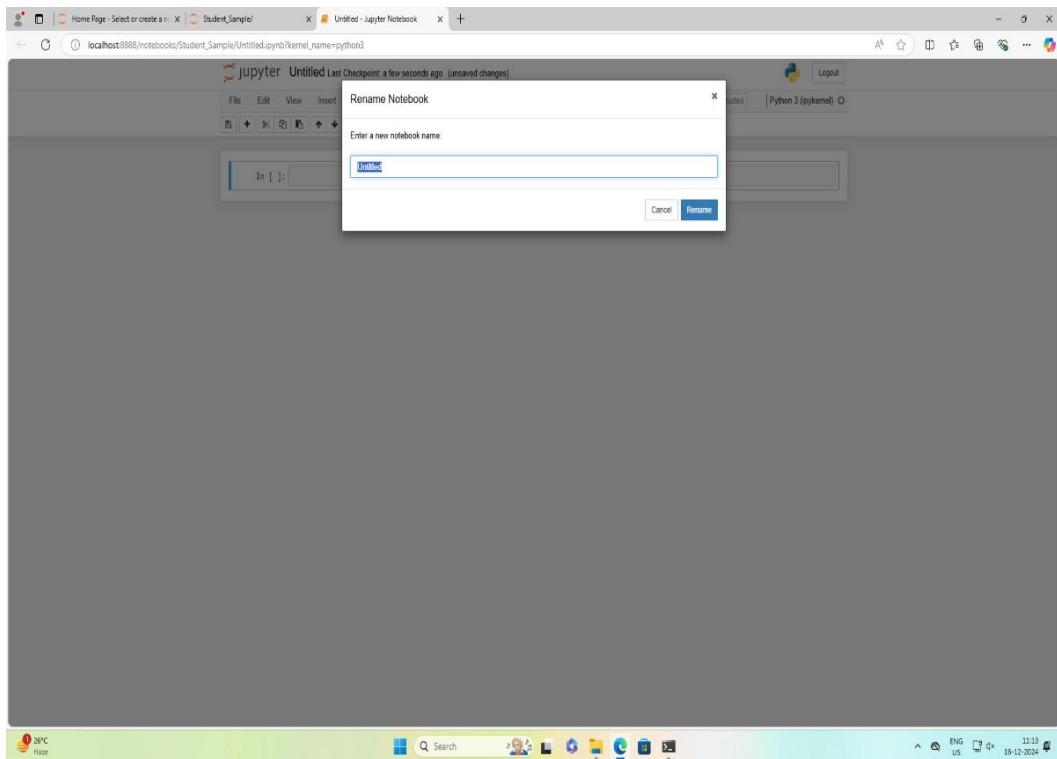
**Fig 3:** Renaming the new folder.

**Step 4:** Creating a new notebook inside the newly created folder. Click on New-> Python 3(ipykernel) as shown in Figure 4.

This screenshot shows the same Jupyter Notebook interface as Figure 3, but with a context menu open over the "Untitled Folder". The menu is triggered by a right-click on the folder. The "New" option is highlighted, and a dropdown menu appears with the following options: "Notebook", "Python 3 (ipykernel)" (which is currently selected), "Text File", "Folder", and "Terminal". The rest of the interface and the status bar at the bottom remain the same as in Figure 3.

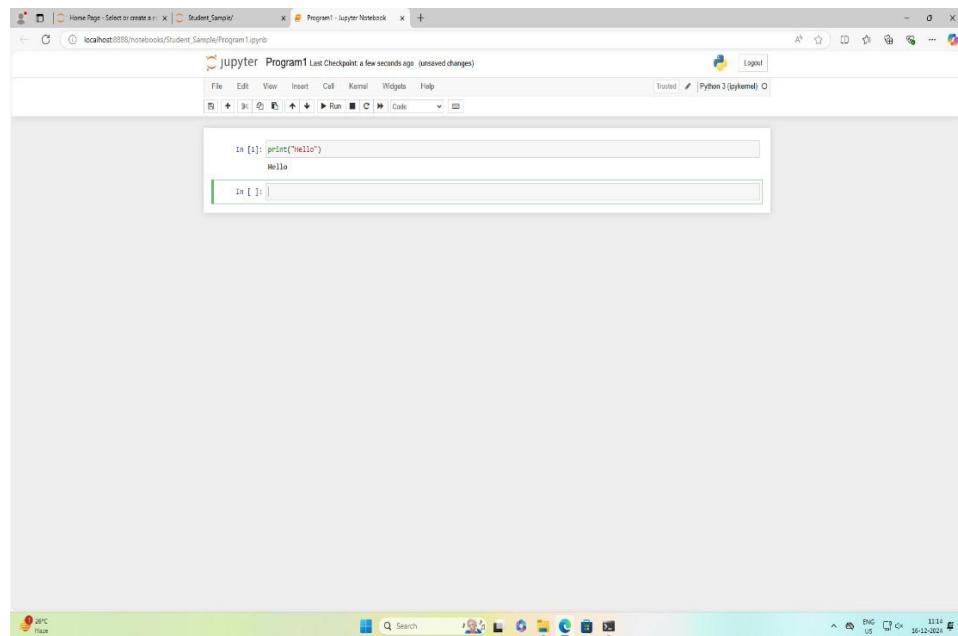
**Fig 4:** Creating a python notebook.

**Step 5:** Renaming the python notebook. Initially, the python program will be created with the default name “Untitled” as shown in Figure 5. Click on it to rename the notebook appropriately.



**Fig 5:** Renaming the python notebook.

**Step 6:** Start running the python program using notebook cells. After renaming the notebook cells, we can start coding. A sample print statement is displayed in Figure 6. After typing the code, press the run button to run the appropriate cell. The output is displayed below the cell.



**Fig 6:** Sample print statement successfully executed in a Jupyter notebook.

## 6. Python Basics

Python is a high-level, interpreted programming language with simple and readable syntax. Python is dynamically typed, and variable types are determined at runtime. Its numerous third-party packages make it suitable for various applications, such as web development, data analysis, artificial intelligence, and scientific computing. For data manipulation, libraries like Pandas and NumPy provide efficient tools for handling structured and unstructured data, performing calculations, and managing large datasets. Python also offers various data visualization libraries, such as Matplotlib and Seaborn, to create interactive charts, graphs, and dashboards. Machine learning and statistical modeling can seamlessly integrate using libraries like Scikit-learn and Statsmodels. Jupyter Notebook enhances the workflow by allowing live coding, visualization, and storytelling in a single environment. Python's scalability, active community support, and compatibility with big data frameworks (e.g., PySpark) make it a go-to language for data-driven projects across industries. Important features of Python are as follows:

- Python uses whitespace (tabs or spaces) to structure code.
- To add comments to code use the hash mark (pound sign) #.
- Variables in Python have no inherent type associated with them; a variable can refer to a different type of object simply by doing an assignment.
- Every number, string, data structure, function, class, module etc. exists in the Python interpreter as a Python object. Each object has an associated type (e.g., integer, string, or function) and internal data.
- Objects in Python typically have both attributes (other Python objects stored “inside” the object) and methods (functions associated with an object that can have access to the object’s internal data).
- In Python, a module is a file with the .py extension containing Python code.
- Python’s data structures like tuples, lists, dictionaries, sets, and sequences, are important aspects of Python programming.
- Vectorization in Python refers to performing operations on entire arrays or data sequences without iterating through individual elements using loops. Vectorised operations are supported in packages like NumPy and Pandas. Python is optimized for vectorized operations and it is much faster than looping constructs.
- Python provides an array of specialized libraries for creating a variety of data visualizations. The most important packages are Matplotlib and Seaborn.

## 7. List of Experiments

Week No	TOPICS	Course Outcome Addressed
Week 1	<p><b>Demo 1: Python Language Basics Exercises</b></p> <ol style="list-style-type: none"> <li>1. Write a Python function to input two numbers and perform the Calculator operations of (+, -, *, /).</li> <li>2. Write a Python function that takes an integer and returns True if it's a prime number and False otherwise.</li> <li>3. Create a Python function that creates a sequence between 1 and 100 and prints all the odd numbers. Compute and display the sum of all the even numbers.</li> <li>4. Write a Python function to add two elements and display the result. The elements can be of type integer, float or string.</li> <li>5. Write a Python function that takes a string input from the user and counts the number of vowels and consonants in the string.</li> </ol>	CO1
Week 2	<p><b>Demo 2: Python built-in Data structures, Functions, modules, packages Exercises</b></p> <ol style="list-style-type: none"> <li>1. Write a Python code block that inputs numbers into a list. Print the largest, smallest, the sum, and the average of the numbers. Count occurrences of a specific number in the list.</li> <li>2. Write a Python code block to create a tuple with five elements. Try to change one of the elements and handle the error that occurs. Print a message that explains why the error occurred.</li> <li>3. Write a Python code block to create a dictionary of cricket World Cup winners. Let the key be the year; the value is the country that won the World Cup that year. Print the name of the best-performing country. Display the unique list of countries that have won the World Cup.</li> <li>4. Write a Python code block that inputs a sentence from the user. Count the frequency of each word in the sentence and store the result in a dictionary. Prints the dictionary with words as keys and their frequencies as values.</li> <li>5. Write a Python code block to input numbers into two sets. Perform union, intersection, and difference operations on the sets and print the results.</li> </ol>	CO1
Week 3	<p><b>Demo 3: NumPy basics and vectorized computation Exercises</b></p> <ol style="list-style-type: none"> <li>1. Generate a 3x4 NumPy array with random integers between 1 and 50.             <ol style="list-style-type: none"> <li>a. Calculate and print the Mean, Median, and Standard Deviation of the array</li> <li>b. Print the Sum of all elements and the sum of each row.</li> <li>c. Reshape the 3x4 array into a 2x6 array and print it.</li> </ol> </li> </ol>	CO2

	<p>2. Create two (<math>3 * 3</math>) matrices using NumPy and print it. Perform and print the results of the following linear algebra operations</p> <ol style="list-style-type: none"> <li>Matrix addition</li> <li>Matrix subtraction</li> <li>Matrix multiplication (element-wise and dot product)</li> <li>Transpose of a matrix</li> <li>Determinant and inverse (if applicable)</li> </ol>	
Week 4	<p><b>Demo 4: Pandas, Data loading, Storage and File formats Exercises</b></p> <ol style="list-style-type: none"> <li>Create a Series from a list of integers representing daily temperatures (in Celsius) over a week. Assign index labels as day of the week.             <ol style="list-style-type: none"> <li>Find and print the average (mean) temperature for the week.</li> <li>Identify and print the maximum and minimum temperatures and their respective days.</li> <li>Display the temperatures greater than a specific value.</li> <li>Convert all temperatures to Fahrenheit.</li> <li>Print the days had temperatures above the average.</li> </ol> </li> <li>Create a data frame with details of 10 students and columns as Roll Number, Name, Gender, Marks1, Marks2, Marks3.             <ol style="list-style-type: none"> <li>Create a new column with total marks</li> <li>Find the lowest marks in Marks1</li> <li>Find the Highest marks in Marks2</li> <li>Find the average marks in Marks3</li> <li>Find student name with highest average</li> <li>Find how many students failed in Marks2 (&lt;40)</li> </ol> </li> </ol>	CO2
Week 5	<p><b>Demo 5: Data Cleaning and Preparation</b></p> <ol style="list-style-type: none"> <li>Create a CSV file called “Movies.csv” with details of 10 movies- Movie Name, Language, Genre, Rating, Review.             <ol style="list-style-type: none"> <li>Read CSV file into a dataframe and find the movie with the highest rating.</li> <li>Write the details of all “Hindi movies into a file “HindiMovies.csv”.</li> </ol> </li> <li>For the CEREALS dataset, perform data preprocessing and answer the following questions.             <ol style="list-style-type: none"> <li>Create a table with the 5 number summary of all the numeric attributes.</li> <li>For each of the numeric attributes (proteins upto vitamins) , identify and replace all missing data(indicated with -1) with the arithmetic mean of the attribute.</li> <li>Create a table with the 5 number summary of all the numeric attributes after treating missing values. Do you think the strategy used in dealing with missing values was effective?</li> <li>For each numeric attribute (proteins upto vitamins), identify and replace all noisy data with the median of attribute.</li> <li>Create a table with the 5 number summary of all the numeric attributes after treating noisy values. Do you think the strategy used in dealing with noisy values was effective?</li> </ol> </li> </ol>	CO3

Week 6	<b>Demo 6: Data Visualization: context, effective visuals and storytelling</b> <b>Exercise</b> 1. For the MTCARS dataset, answer the specified questions with summarization and effective visuals. 2. For the CEREALS dataset, answer the specified questions with summarization and effective visuals.	CO3, CO4
Week 7	<b>Demo 7: Plotting and Visualization using Matplotlib &amp; Seaborn</b> <b>Exercise</b> 1. For the IPL dataset, answer the specified questions with summarization and effective visuals using Matplotlib & Seaborn libraries	CO4
Week 8	<b>Demo 8: Data Aggregation and Group Operations</b> <b>Exercise</b> 1. For the NORTHWIND dataset, answer the specified questions with summarization and effective visuals.	CO3
Week 9	<b>Demo 9: String Manipulation and Data Wrangling</b> <b>Exercise</b> 1. For the SENTIMENT dataset, answer the specified questions with string operations and effective visuals.	CO3
Week 10	<b>Discussion of case study and data set.</b> 1. For the case study given, answer the questions with a report with story, visuals and data summaries.	CO5
Week 11	<b>Discussion of case study and data set.</b> 1. For the case study given, answer the questions with a report with story, visuals and data summaries.	CO5
Week 12	<b><i>End-term lab examination</i></b>	

## 8. References:

SL.No	References
1	<b>Text Book:</b> Wes McKinney , Python for Data Analysis: Data Wrangling with pandas, NumPy & Jupyter. 3rd edition. O'Reilly Media, 2022.
2	Cole Nussbaumer Knaflic, Storytelling With Data: A Data Visualization Guide for Business Professionals, John Wiley and Sons, 2015.
3	Jake VanderPlas, Python Data Science Handbook. O'Reilly Media, 2016.
4	Alberto Boschetti and Luca Massaron, Python Data Science Essentials, 3rd edition, Packt Publishing Ltd. 2018.
5	Manaranjan Pradhan, U Dinesh Kumar, "Machine Learning using Python", Wiley India, 2019.
6	Python documentation: <a href="https://docs.python.org/3/">https://docs.python.org/3/</a>