Python (Installation and Basic Commands) and Libraries, Jupyter Notebook, Set Up GitHub

# Python (Installation and Basic Commands) and Libraries

## Overview

Python is a high-level, interpreted programming language known for its simplicity and readability. It is widely used for web development, data analysis, artificial intelligence, scientific computing, and more.

## Installation

1. Download Python: Visit the official Python website and download the latest version of Python.  
2. Run Installer: Execute the downloaded installer. Ensure to check the box that says "Add Python to PATH."  
3. Verify Installation: Open a terminal or command prompt and type `python --version` to check the installation.

## Basic Commands

1. Printing:  
```python  
print("Hello, World!")  
```  
  
2. Variables:  
```python  
x = 5  
y = "Hello"  
```  
  
3. Data Types:  
```python  
integer = 10  
floating\_point = 10.5  
string = "Hello"  
boolean = True  
```  
  
4. Lists:  
```python  
my\_list = [1, 2, 3, 4, 5]  
```  
  
5. Dictionaries:  
```python  
my\_dict = {"name": "Alice", "age": 25}  
```  
  
6. Loops:  
```python  
for i in range(5):  
 print(i)  
```  
  
7. Functions:  
```python  
def greet(name):  
 return "Hello, " + name  
```

## Libraries

1. NumPy: Library for numerical computations.  
```python  
import numpy as np  
arr = np.array([1, 2, 3, 4, 5])  
```  
  
2. Pandas: Library for data manipulation and analysis.  
```python  
import pandas as pd  
df = pd.DataFrame({"Name": ["Alice", "Bob"], "Age": [25, 30]})  
```  
  
3. Matplotlib: Library for plotting and visualization.  
```python  
import matplotlib.pyplot as plt  
plt.plot([1, 2, 3], [4, 5, 6])  
plt.show()  
```

# Jupyter Notebook

## Overview

Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text.

## Installation

1. Install Anaconda: Download and install Anaconda, which comes with Jupyter Notebook pre-installed.  
2. Launch Jupyter Notebook: Open Anaconda Navigator and launch Jupyter Notebook.

## Basic Usage

1. Create a New Notebook: In the Jupyter dashboard, click "New" and select "Python 3".  
2. Running Code: Enter code in a cell and press Shift + Enter to run it.  
3. Markdown Cells: Switch a cell to Markdown to add formatted text.  
4. Saving Notebooks: Notebooks are saved with the `.ipynb` extension and can be exported to various formats.

## Example

```python  
# Importing Libraries  
import numpy as np  
import pandas as pd  
  
# Creating a DataFrame  
df = pd.DataFrame({  
 "Name": ["Alice", "Bob"],  
 "Age": [25, 30]  
})  
  
# Displaying the DataFrame  
df  
```

Integration of GitHub Gist Using Google Colab

# Overview

GitHub Gist is a simple way to share snippets and pastes with others. Google Colab is a free, cloud-based notebook environment that allows you to write and execute Python code in your browser. Integrating GitHub Gist with Google Colab can streamline the process of sharing and collaborating on code snippets.

# Steps for Integration

## Creating a Gist on GitHub

1. Go to [GitHub Gist](https://gist.github.com/).  
2. Click on "New gist".  
3. Add a description and paste your code into the provided text box.  
4. Choose the visibility of your gist (public or secret).  
5. Click "Create public gist" or "Create secret gist".

## Accessing Gist in Google Colab

1. Open [Google Colab](https://colab.research.google.com/).  
2. Click on "File" in the top menu and select "Open notebook".  
3. Navigate to the "GITHUB" tab.  
4. If it's your first time, you'll need to connect your GitHub account by clicking "Connect to GitHub" and authorizing the access.  
5. Once connected, enter the URL of the gist in the search bar and press Enter.

## Editing and Running Code from Gist

1. After loading the gist into Colab, you can edit and run the code as usual.  
2. Any changes made in Colab can be saved back to the gist by clicking "File" -> "Save a copy in GitHub..." -> "Gist".

## Sharing Your Work

1. Share the link to your Colab notebook or gist with collaborators.  
2. Collaborators can open the notebook in Colab, run the code, and make changes if they have edit permissions.

# Example Workflow

## Create a Gist

```python  
# Sample Python code to create a simple function  
def greet(name):  
 return f"Hello, {name}!"  
```

## Load Gist in Colab

1. Go to Google Colab.  
2. Open the notebook from the Gist URL.  
3. Edit the code in Colab:  
```python

# Modified function in Colab  
def greet(name, greeting="Hello"):  
 return f"{greeting}, {name}!"  
```  
4. Run the code in Colab to see the output.

## Save Changes Back to Gist

1. After editing, save the changes back to the Gist:  
```markdown  
# Save a copy in GitHub...  
- Select "Gist".  
- Provide a commit message.  
- Click "OK".  
```

# Benefits of Integration

Collaboration: Easily share code snippets and collaborate with others.  
Version Control: Maintain version control of your code snippets using Gist.  
Convenience: Work in a cloud-based environment without needing local setup.  
Portability: Access your code from any device with internet access.