**Hands-On Setup and Introduction to Python Environment**

* **Objective**: Set up a programming environment on Google Colab, install essential libraries, and familiarize interns with Python basics for NLP projects.

### **Set Up Google Colab Programming Environment**

**Step-by-Step Procedure to Install and Set Up Google Colab**

1. **Access Google Colab**:
   * Open your web browser and go to https://colab.research.google.com.
   * Google Colab runs directly in the browser and doesn’t need any installation.
2. **Sign in with Google**:
   * You’ll need a Google account to access Google Colab. If you’re not already signed in, sign in with your Google credentials.
3. **Create a New Notebook**:
   * Once you’re signed in, you’ll see the main Colab interface.
   * Click on the **‘File’** menu, then select **‘New Notebook’**. This will create a fresh notebook where you can start coding in Python.

**Demonstration of How to Open and Navigate Google Colab**

1. **Notebook Interface**:
   * You’ll see an empty code cell with a “+ Code” button for adding more cells.
   * You can type code into each cell and execute it independently.
2. **Basic Menu Options**:
   * **File**: Options to create a new notebook, open existing notebooks, and save your work.
   * **Edit**: Offers basic editing tools.
   * **Runtime**: Manage your notebook’s runtime (the backend server that executes code), reset runtime, or select GPU/TPU resources.
3. **Running a Code Cell**:
   * Type any Python code (e.g., print("Hello, World!")) in the cell and press **Shift + Enter** or click the **Play icon** on the left of the cell to execute the code.

### **Installing Essential Python Libraries**

**Step-by-Step Procedure to Install Basic Libraries: pandas, numpy, nltk, and spacy**

1. **Installing Libraries**:
   * In Google Colab, use pip to install packages.
   * Type the following commands in a new code cell and run each to install the libraries:

python

*!pip install pandas*

*!pip install numpy*

*!pip install nltk*

*!pip install spacy*

1. **Downloading NLTK Data**:
   * After installing NLTK, download essential datasets used by the library:

python

*import nltk*

*nltk.download('punkt')*

*nltk.download('stopwords')*

1. **Verify SpaCy Installation**:
   * SpaCy also requires language models to perform NLP tasks. Run this command to download the English model:

python

*!python -m spacy download en\_core\_web\_sm*

**Procedure to Check Verification of Successful Installation**

1. **Import Libraries to Confirm Installation**:
   * In a new code cell, type and run the following code:

python

*import pandas as pd*

*import numpy as np*

*import nltk*

*import spacy*

*print("All libraries installed successfully!")*

* + If the cell executes without errors, the libraries are correctly installed.

**Explanation of Each Library’s Purpose in NLP**

1. **pandas for Data Handling**:
   * **Purpose**: pandas is a powerful data manipulation library that helps to load, analyze, and manipulate data in various formats.
   * **Usage in NLP**: Commonly used for reading datasets (CSV, Excel), filtering, grouping, and preparing text data for processing.
2. **numpy for Numerical Operations**:
   * **Purpose**: numpy handles arrays and complex mathematical operations.
   * **Usage in NLP**: Supports vectorization of text data, matrix operations, and working with embeddings and large datasets efficiently.
3. **nltk for Text Processing**:
   * **Purpose**: nltk (Natural Language Toolkit) provides tools for text processing, such as tokenization, stemming, lemmatization, and stop-word removal.
   * **Usage in NLP**: Used for tokenizing sentences, cleaning text, and extracting linguistic features.
4. **spacy for Advanced NLP Functionalities**:
   * **Purpose**: spacy is a high-performance NLP library for tasks like part-of-speech tagging, named entity recognition, and dependency parsing.
   * **Usage in NLP**: Allows for more sophisticated NLP tasks and includes pre-trained models that improve NLP accuracy.

### 

### **Run a Basic Python Script (10 minutes)**

**Python Script to Print "Hello, NLP"**

1. **Open a New Code Cell**:
   * Click on the **+ Code** button to add a new cell to the Colab notebook.
2. **Write and Execute Code**:
   * In the new cell, type the following Python code to print a welcome message:

Python

*print("Hello, NLP")*

* + Run the cell by pressing **Shift + Enter** or clicking the **Play icon**.

**Walkthrough of Python Syntax Basics in Google Colab**

1. **Variables**:
   * Python uses simple syntax to define variables:

python

*text = "NLP with Python"*

*print(text)*

1. **Basic Data Types:**
   * Strings (str), Integers (int), and Lists (list) are fundamental in Python:

python

*message = "Welcome to NLP"*

*number = 10*

*words = ["token", "stemming", "lemmatization"]*

*print(message, number, words)*

1. **Simple Loop:**
   * Use loops to iterate through text or data elements:

python

*for word in words:*

*print(word)*

**How to Save and Run Scripts in Google Colab**

1. **Saving Your Work**:
   * Colab automatically saves notebooks to your Google Drive.
   * You can also download a notebook by going to **File > Download .ipynb** for sharing or later use.
2. **Running Cells**:
   * Cells in Colab can be run independently. This feature helps with debugging and allows you to test individual sections of your code.
3. **Conclusion**:
   * Once familiar with Google Colab’s basics, participants can confidently move on to writing and running more complex scripts in NLP.