Preparing Environment for Coding

Creating a virtual environment in Python is essential for managing dependencies for different projects. Here's how to set up a virtual environment on a Windows machine:

Step 1: Open Command Prompt / Powershell

Press `win+r`, type `cmd`, and hit `Enter` to open the Command Prompt.

Step 2: Navigate to Your Project Directory

Use the `cd` command to navigate to the directory where you want to create the virtual environment.

```
cd path\to\your\project\directory
```

> **Caution**
>
> [Optional] For this entire training we recommend using the same directory or Environment.

Step 3: Install the Virtualenv Package

If you haven't already installed `virtualenv`, install it using pip:

```
pip install virtualenv
```

Step 4: Creater the virtual environment

Run the following command to create the virtual environment. Requires **env_name** with your desired environment name:

```
python -m venv env_name
```

> **Tip**
>
> In general, we recommend using **.env** as the virtual environment name.

Step 5: Activate the virtual environment

Run the following command to activate the virtual environment. Once the environment is created, you need to activate with the following command:

On "Command Prompt"

```
env_name\Scripts\activate
```

On "Powershell"

```
env_name\Scripts\Activate.ps1
```

> ⚠ **Warning**
>
> If you face any problems on permisssions you should contact your IT Team.
> Consider the solution if you're Windows User along with powershell
> Support Activating Virtualenv Issue - Solution

Step 6: Update the pip package version with the latest version

Once you have activated the virtual environment, You should update the pip package with the latest version

```
pip install --upgrade pip
```

Step 7: Verify the virtual environment

Your command prompt should now reflect the virtual environment's name, indicating that it is active. You can verify that you're working within the virtual environment by checking the Python executable path:

```
python
import sys
sys.executable
```

It should return your Python Enviroment Installed Location.

Step 8: Install the required packages and dependencies within the virtual environment

With the virtual environment activated, you can install packages that will only be available within this environment:

We reommend you to install the following packages for the hands on for the Training:

1. requests
2. numpy
3. pandas
4. matplotlib
5. jupyterlab

```
pip install requests numpy pandas ...
```

Step 9: Deactivating the Virtual ENvironment

When you're done working in the virtual environment, You can deactivate it by simply running:

```
deactivate
```

> **Note**
>
> you should ensure that `deactivate` command executes and you're out of the environment.