1. Install Python 3.6

Windows

1. Download and install the executable for **version 3.6.5** from official website https://www.python.org/downloads/ .
2. Please make sure tick the checkbox to "*Add Python 3.6 to PATH*" at the time of installing
3. Verify if python 3.6 got installed by going to command prompt and type

```
> python --version
```

Make sure this shows version 3.6.5

4. In case there was already a previous version of python on the system, the above would return a different version. To get python 3.6 working,
    a. Make sure that the newly installed python 3.6 was added in the PATH variable as follows. To check Hit Start, type "Advanced system settings," and then select the "View advanced system settings" option. In the "System Properties" window that opens, on the "Advanced" tab, click the "Environment Variables" button. Check if the newly installed path is present in the PATH variable.
    b. To start using the the new version of python, fire up File Manager and head to the folder where you installed Python 3 (C:\Users\[username]\AppData\Local\Programs\Python\Python36 by default). Make a copy of the "python.exe" file, and rename that copy (not the original) to "python3.exe".
    c. Open a new command prompt (the environmental variables refresh with each new command prompt you open), and type

```
> python3 --version
```

Linux

Reference - https://www.tecmint.com/install-python-in-ubuntu/

1. Install python using apt as shown below

```
$ sudo apt update
```

```
$ sudo apt install python3.6
```

2. To use/verify Python 3.6 installation, invoke the following command.

```
$ python3.6
```

2. Install pip

Windows

Refer - https://programwithus.com/learn-to-code/Pip-and-virtualenv-on-Windows/

1. Usually Python3 comes with pip preinstalled. Check by typing in command prompt

```
> pip --version
```

2. If it does not point to python 3.6.5 version (due to multiple versions of python installed on the system) then uninstall existing pip. In command prompt type

```
> pip uninstall pip
```

   Now verify the install by checking the pip version

```
> pip --version
```

3. If it still doesn't point to the correct version of python 3.6.5 then download **get-pip.py** bootstrapper from https://bootstrap.pypa.io/get-pip.py and run

```
> python3 get-pip.py
```

4. Once you have pip installed,make sure it is upgraded to the latest version by running the following command

```
> python -m pip install --upgrade pip
```

   OR in case you have multiple versions of python then run using *python3*

```
> python3 -m pip install --upgrade pip
```

Linux

Refer - https://gist.github.com/frfahim/73c0fad6350332cef7a653bcd762f08d

1. Usually Python3 comes with pip preinstalled. Check by typing in command prompt

```
$ pip --version
```

2. If it doesn't point to python 3.6.5 version (due to multiple versions of python installed on the system) then install pip for version 3 by running following command

```
$ sudo apt-get install python3-pip
```

3. Verify pip installation by checking version. It would be pointing to python 3.6

```
$ pip --version
```

3. Install Virtualenv

Refer - https://programwithus.com/learn-to-code/Pip-and-virtualenv-on-Windows

virtualenv is a tool to create isolated Python environments. Virtualenv creates a folder which contains all the necessary executables to use the packages that a Python project would need.

Windows

1. **Install virtualenv**, In your Command Prompt enter:

```
> pip install virtualenv
```

2. Create a new folder where you would be setting up the project (In below example our folder is named “*rasa*” and is located at the Desktop).

```
> mkdir c:\Users\[username]\Desktop\rasa
```

3. Go to that folder on the command prompt and **create a virtual environment** having the name “*rasaenv*” (or anything of your choice)

```
> cd c:\Users\[username]\Desktop\rasa
```

```
> python3 -m venv rasaenv
```

4. **Activate the virtual environment** by running following command (Assuming you are inside project folder “rasa”)

```
> rasaenv\Scripts\activate.bat
```

5. Once you have the virtual environment activated, you can go ahead and install any python dependencies for this project inside it.

6. To **de-activate a virtual environment**,

```
> deactivate
```

Linux

Reference link - https://gist.github.com/frfahim/73c0fad6350332cef7a653bcd762f08d

1. **Install virtualenv** using pip3 by running following shell command

```
$ sudo pip3 install virtualenv
```

2. Create a new folder where you would be setting up the project. Go to that folder on the terminal and **create a virtual environment** having the name "*rasaenv*" (or anything of your choice)

```
$ python3.6 -m venv rasaenv
```

3. **Activate the virtual environment** by running following command (Assuming you are inside project folder)

```
$ source rasaenv/bin/activate
```

4. Once you have the virtual environment activated, you can go ahead and install any python dependencies for this project inside it.

5. To **de-activate a virtual environment**,

```
$ deactivate
```

4. Install RASA stack

Windows and Linux

1. Download this text file and place it inside the project folder where the virtual environment "*rasaenv"* is created.
2. **Make sure virtual environment is activated** before installing RASA dependencies.
    a. To activate the virtual environment, Go to the project folder in the command prompt/ terminal and activate the virtual environment "*rasaenv"* created previously.

    For windows

```
> rasaenv\Scripts\activate.bat
```

For Linux

```
$ source rasaenv/bin/activate
```

3. Once inside the virtualenv, run the following to install RASA dependencies

```
> pip install -r requirements.txt
```

This will install RASA nlu, RASA core and some other python dependencies.

4. Install a spaCy English language model

```
> python -m spacy download en
```

Please note: You may face issues installing RASA core or RASA NLU if you don't have C++ compiler installed.

1. In case of Windows, this can be installed by installing C++ compiler tools and Windows SDK via Visual Studio. Simplest way to do so will be to:
    a. Open Visual Studio.
    b. Choose File | New | Project | C++ and you will be prompted to install the necessary components.
2. In case of Ubuntu, GCC is already installed. In case it is not you can do so by running following shell command:
    a. `sudo apt-get install gcc g++`
3. In case of Mac,
    a. xcode-select --install

6. Install Git

Reference - https://www.atlassian.com/git/tutorials/install-git

Windows

Reference - https://www.atlassian.com/git/tutorials/install-git#windows

1. Download the executable from the official site - https://git-scm.com/download/win
2. Install from the downloaded executable

Linux (Debian/Ubuntu)

Reference - https://www.atlassian.com/git/tutorials/install-git#linux

1. From your shell, install Git using apt-get:

```
$ sudo apt-get update
$ sudo apt-get install git
```

5. Install nvm

Windows

1. Download NVM for windows from the following link: https://github.com/coreybutler/nvm-windows/releases/download/1.1.6/nvm-setup.zip
2. Extract the *nvm-setup.exe* file and run the setup
3. Change destination location to "*C:\Program Files\nvm*"
4. Click *Next* & *Finish* to complete the setup
5. Open cmd and type "*nvm*" to verify installation.If you get an output, the program is successfully installed.
6. If you get an error add the nvm path to environment variables. To do so:
    a. right click on the "*Start*", goto "*system*", click on "*Advanced system settings*" from the left pane, click on "*Environment Variables...*", under "*System variables*" double click on "*Path*" and Add ";c:\Program Files\nvm" to the end of the Variable value
    b. Deleting the existing values of path can stop many programs from working. PLease do not delete the existing value. Add the path to the end of the value.
    c. Start a new command prompt and type in "*nvm*" to check if it is working

Linux

1. Open terminal and type

```
$ "wget -qO-
https://raw.githubusercontent.com/creationix/nvm/v0.33.8/install.sh |
bash"
```

6. Install NPM and Node.js

Windows

1. Close the cmd prompt, right click on the start and open "*Command Prompt (Admin)*"
2. Type in "*nvm install latest*" to install node.js and npm
3. Verify the installation by typing "*npm*" on the command prompt
4. Note: if you have a 64bit machine, u will have to go to the directory where node is installed and rename it from "*node64*" to "*node*"

Linux

1. Type "*nvm install node*" on your terminal
2. Verify installation by typing “nvm”