**Github cheatsheet :**  
<https://github.com/tchapi/markdown-cheatsheet>  
<https://guides.github.com/activities/hello-world/>

**Study Plan for Robotics with ROS**  
Please complete all the instruction about Ubuntu, Python, and ROS in the virtual machine. However, the section, “ROS on Robotic Platform” should be completed on a native installation on the Raspberry Pi 4 computer.  
All the resources in this instruction are freely available online, except for the book “ROS Robotics Projects” by Ramkumar Gandhinathan, Lentin Joseph. However, you can access it with Purdue subscription.  
**Ubuntu Basics**  
Install Ubuntu in a virtual box rather than dual booting. This will be useful for Ubuntu learning because you will often break the install system during the learning process. With virtual boxes, recovering the system will be trivial.  
Robotics or complex mechatronics system are commonly controlled by Linux computers. These computers generally are “headless” – they do not have monitor, keyboard, or graphical user interfaces. To interact with these systems, we will use the command line interface. Therefore, you will need to memorize the most popular commands.

1. **Instal**[**l virtual box**](https://www.youtube.com/watch?v=8mns5yqMfZk)
2. **Install Ubuntu20.04 in**[**virtual box**](https://linuxhint.com/install_ubuntu_virtualbox_2004/)
3. [**Manage multiple terminals on Ubuntu with Terminator**](https://www.youtube.com/watch?v=NpphqfoQFmQ)
4. [**Memorize basic command lines**](https://ubuntu.com/tutorials/command-line-for-beginners#1-overview)**.**[**Most used commands**](http://www.linuxandubuntu.com/home/10-basic-linux-commands-that-every-linux-newbies-should-remember)

**ROS Basics**  
**In this section we will learn the very basic of ROS. Intermediate and advanced ROS concepts will be learn on demand.**

1. **Watch**[**Introduction to ROS capabilities**](https://www.youtube.com/watch?v=SNxZDGOLrxw)
2. **Install**[**ROS Noetic**](http://wiki.ros.org/noetic/Installation/Ubuntu)**in Ubuntu image using terminal . Install the full image, sudo apt install ros-noetic-desktop-full**
3. [**Complete Official Beginner Level Tutorials**](http://wiki.ros.org/ROS/Tutorials#Beginner_Level)
4. [**Memorize the basic ROS commands**](https://github.com/bosch-io/bcx18-openADx-examples/wiki/Useful-ROS-Commands)

**ROS on Robotic Platform**  
At the end of this instruction, you should be able to use ROS in a real system. We will use a Raspberry Pi 4 because it has input and output pins, and strong documentation support.

1. [Install Ubuntu 18 and ROS Melodic on Raspberry Pi 4](https://www.hackster.io/shahizat005/getting-started-with-ros-melodic-on-raspberry-pi-4-model-b-cbdec8) – Ignore the Lidar information. Alternatively, use [this official Ubuntu image](https://ubuntu.com/download/raspberry-pi)
2. [Install GPIO Zero for pin Input/Output](https://gpiozero.readthedocs.io/en/stable/installing.html)
3. [Complete GPIO Zero tutorials](https://gpiozero.readthedocs.io/en/stable/recipes.html)
4. [Use ROS on multiple machines](http://wiki.ros.org/ROS/Tutorials/MultipleMachines)

**Python Basics**  
To decrease the learning curve of Python, we will use notebooks. With notebooks, we can use Python code interactively – that is, we can type python code in a terminal and check the code output in real time. The notebook capability is available in the Visual Studio Code IDE.

1. [Install Visual Studio Code](https://code.visualstudio.com/docs/setup/linux#_debian-and-ubuntu-based-distributions) in Ubuntu image
2. [Learn support for Python using cells](https://code.visualstudio.com/docs/python/jupyter-support-py)
3. [Learn Python/Numpy/Scipy/Matplotlib](https://cs231n.github.io/python-numpy-tutorial/)