

**Getting started: data analysis with VS Code**

**Introduction**

During this lab, you will explore Anaconda prompt and Visual Studio (VS) Code and run a simple Python script for data analysis via VS Code.

**Estimated Time**

30 minutes

**Objectives**

At the end of this lab, you will be able to:

* Create virtual environments and switch between them.
* Explore Visual Studio (VS) Code.
* Run a script on VS code.

**Logon Information**

Use the following credentials to sign into virtual environment.

* Username: **Administrator**
* Password: **Passw0rd!**

Table of Contents

[Lab: Getting start data analysis with VS Code 3](#_Toc525663037)

[Exercise 1: Create Virtual Environment 3](#_Toc525663038)

[Exercise 2: IDE walkthrough 9](#_Toc525663039)

[Exercise 3: Running a code 18](#_Toc525663040)

Lab: Getting started data analysis with VS Code

During this lab, you will learn about the Python data analysis environment using VS code.

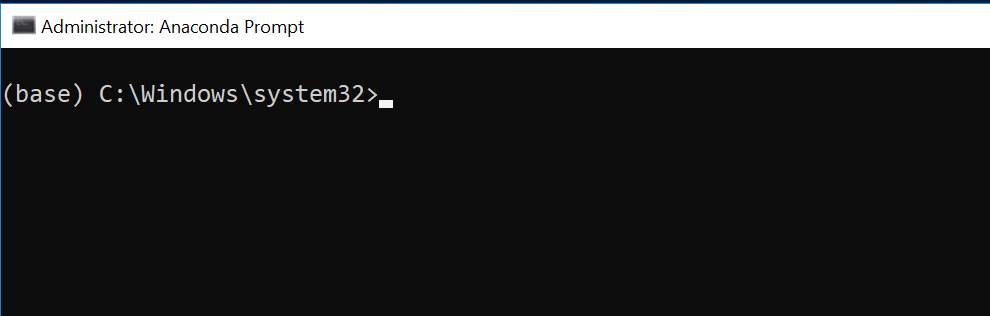
Exercise 1: Create Virtual Environment

In this exercise, you will learn the basics of virtual environment operations.

Tasks

1. Start Anaconda Prompt
2. Search for Anaconda Prompt by typing **Anaconda** in the Search box next to Windows icon. Then, Right-click **Anaconda Prompt** and select **Run as administrator**.
3. Select **Yes** if the User Account Control pop-up appears.
4. Check Anaconda Prompt

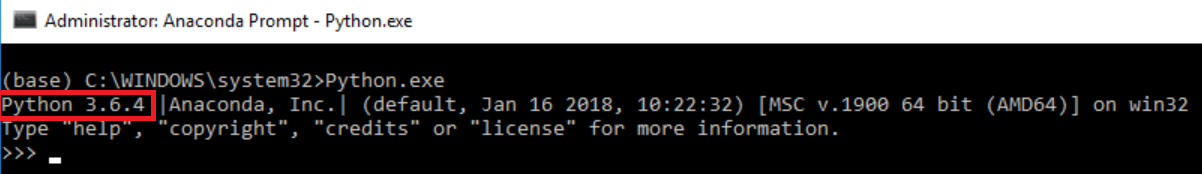
Anaconda prompt will start. By default, a virtual environment called base is selected.



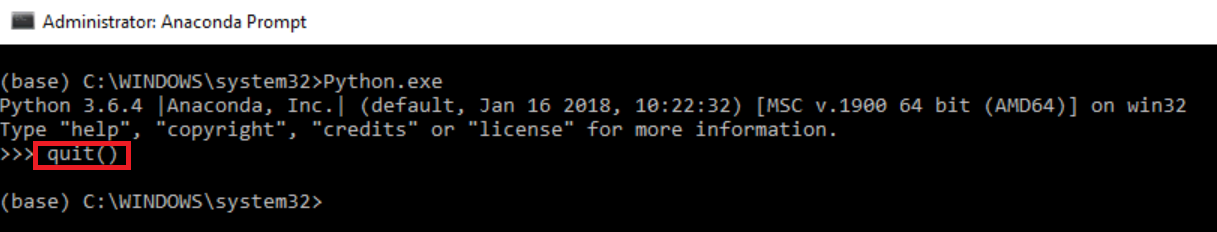
1. Check the base environment
2. Check what is installed in the base environment. First, let's check the Python version. Type the following command in Anaconda Prompt, and then hit the **Enter** key:

Python.exe

1. When Python starts, you can check the version.



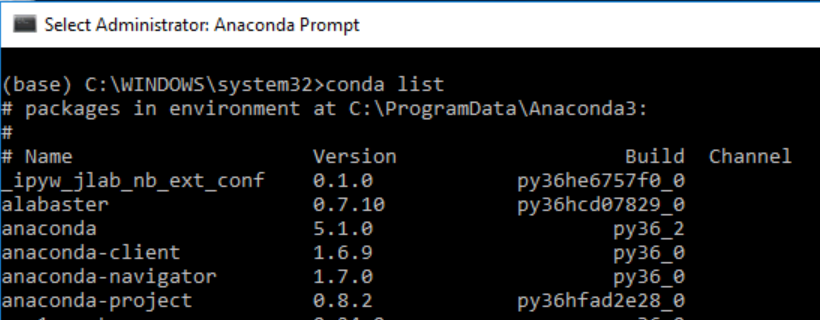
1. After checking the Python version, enter **quit()** in Anaconda Prompt and exit Python.



1. Second, let's look for the installed packages (management units of object, function, and method). Type the following command in Anaconda Prompt, and then hit the **Enter** key:

conda list

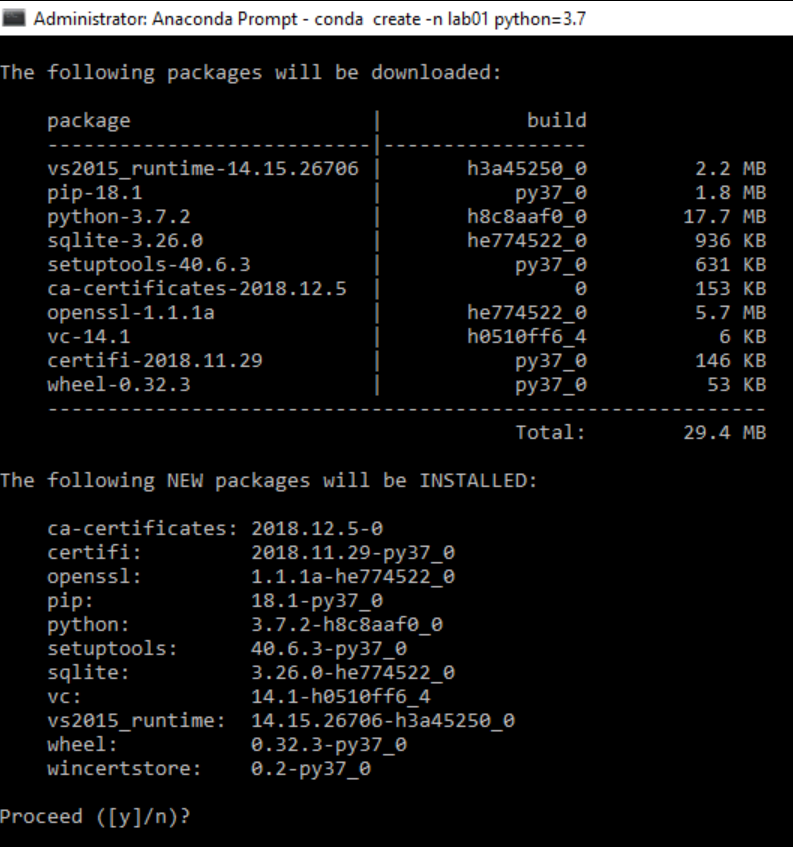
1. You see a list of packages installed by the Anaconda installer.



1. Creating a virtual environment
2. Use the **conda** utility to create a new virtual environment. You can create multiple virtual environments on a single device, so you can easily switch between **versions** **of Python** or **a set of packages**. Type the following command in Anaconda Prompt, and then hit the **Enter** key:

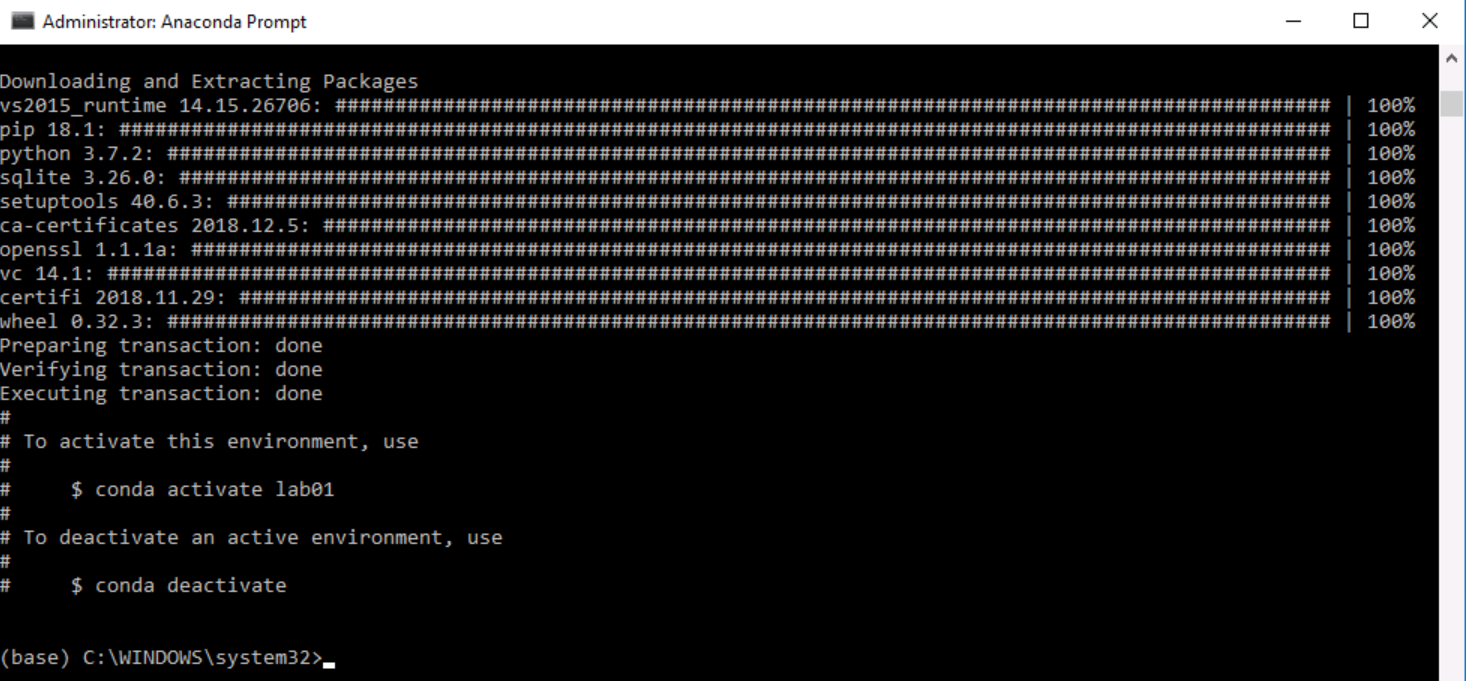
conda create -n lab01 python=3.7

1. This command specifies the name of the new virtual environment as an **-n** argument. Additionally, the version of Python you are installing is explicitly specified as 3.7 If you do not specify a python version, the same Python version is used as the base virtual Environment version which is 3.6.4.
2. The conda utility checks the environment you are using and extracts the components that need to be installed. The packages that are required to create a new virtual environment are listed in Anaconda Prompt. You will be prompted to confirm that you want to continue with the installation, so type **y**.



The process of creating this virtual environment requires a device to connect to the Internet.

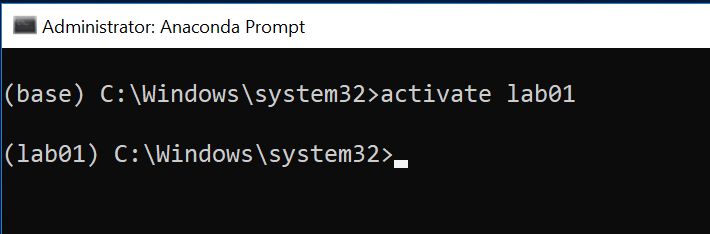
1. If you wait a while, the virtual environment is completed, and you see how the virtual environment is being used.



1. Activate Virtual Environment
2. The virtual environment is not available when you install it. To take advantage of the target virtual environment, you must activate it before you can use it. Type the following command in Anaconda Prompt, and then hit the **Enter** key:

activate lab01

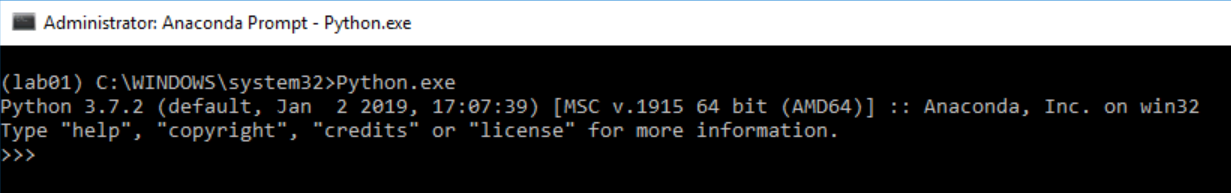
1. When the command completes, the notation in Anaconda prompt is changed.



1. Check the new environment
2. Try to see the Python version in the new environment. Type the following command in Anaconda Prompt, and then hit the **Enter** key:

Python.exe

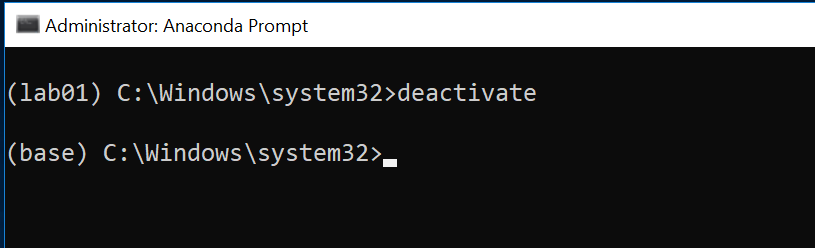
1. When Python starts, you can check the version. You can see that version 3.7 of Python is running though the base version is 3.6.



1. After checking the Python version, enter **quit()** in Anaconda Prompt and exit Python.
2. Deactivate Virtual Environment
3. If you want to revert to the virtual environment you switched to, run deactivate. Type the following command in Anaconda Prompt, and then hit the **Enter** key:

deactivate

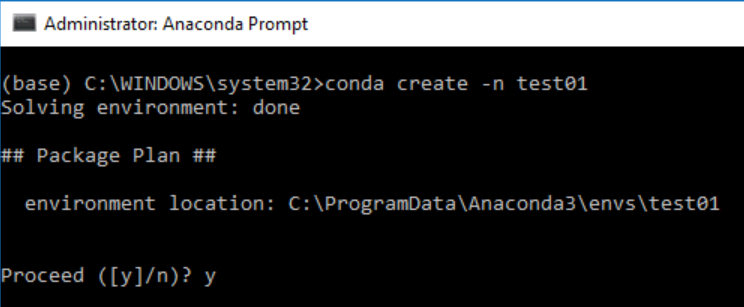
1. You will be reverted back to the base environment.



1. View a list of virtual environments
2. You typically switch between the packages you want to use and the projects that you are responsible for. Therefore, you might want to see what virtual environments exist on your device.
3. Before you list virtual environments, add another virtual environment. Type the following command in Anaconda Prompt, and then hit the **Enter** key:

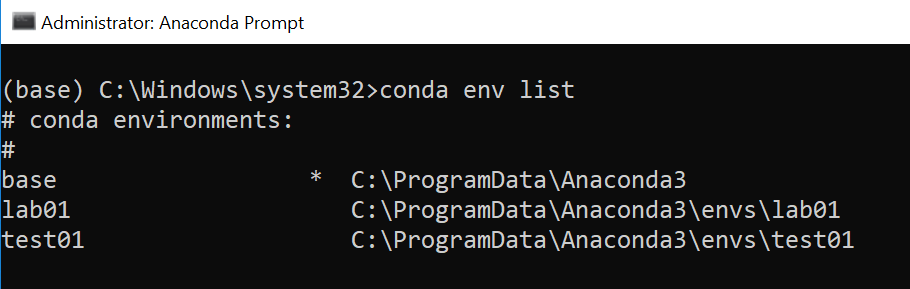
conda create -n test01

1. No Python version is specified while creating test01 virtual environment.



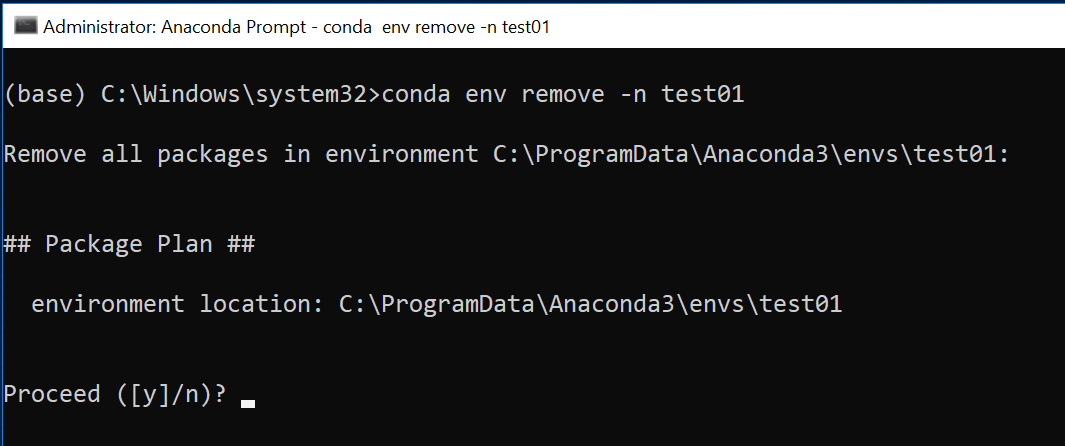
1. You will be prompted to confirm that you want to continue with the installation, so type in **y**.
2. Now you have a total of three virtual environments on the device you are using. Let's try to list the virtual environment. Type the following command in Anaconda Prompt, and then hit the **Enter** key:

conda env list



1. Delete Virtual Environment
2. You may want to delete the virtual environment as the project ends or the POC ends. Type the following command in Anaconda Prompt, and then hit the **Enter** key:

conda env remove -n test01

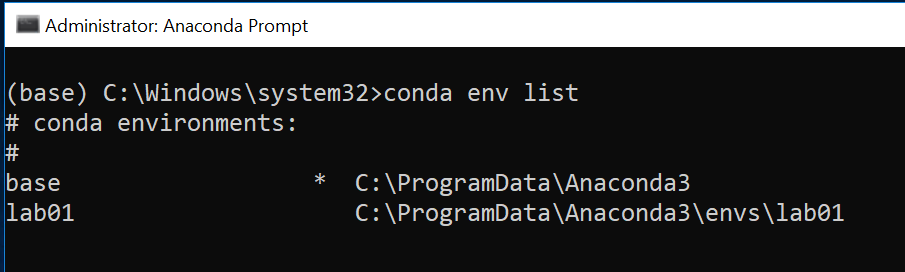


1. You will be prompted to confirm that you want to continue with the installation, so type **y**.

If you have activated the environment you want to delete, the delete operation will fail. Please run deactivate before deleting.

1. Confirm that the deletion was possible. Type the following command in Anaconda Prompt, and then hit the **Enter** key:

conda env list



Exercise 1 has been completed.

Exercise 2: IDE walkthrough

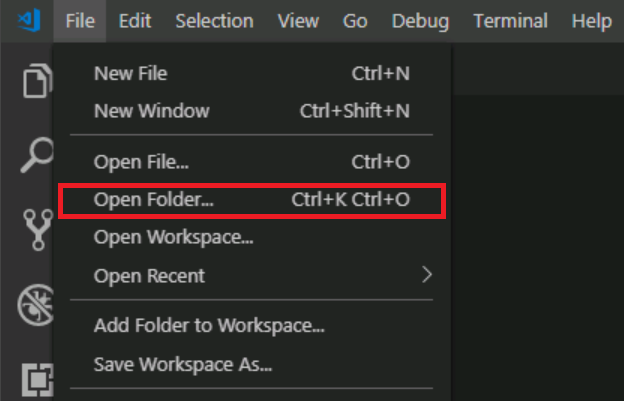
In this exercise, you'll learn how to use VS code.

Tasks

1. Start Visual Studio Code

Start Visual Studio Code from the **Start Menu**. If you can't find it in the Start Menu, search for it by typing **Visual** in the search bar.

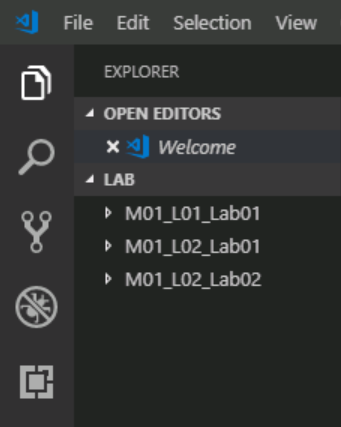
1. Open Folder
2. Open the Lab folder from the File menu. Select the File menu from the menu bar, then select **Open folder…**.



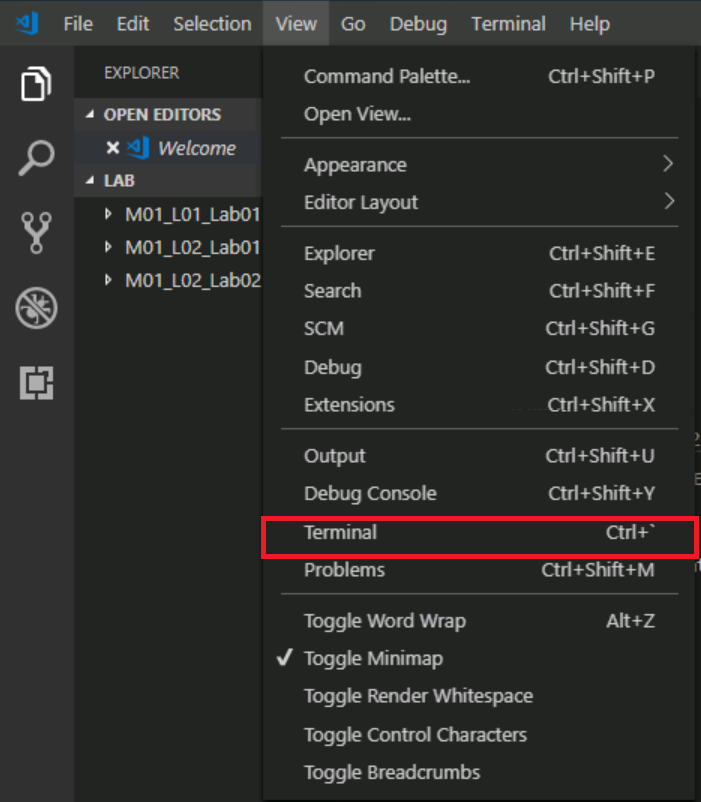
1. In the dialog box that appears, type the folder path as shown below:

C:\Labs\Module1\LAB

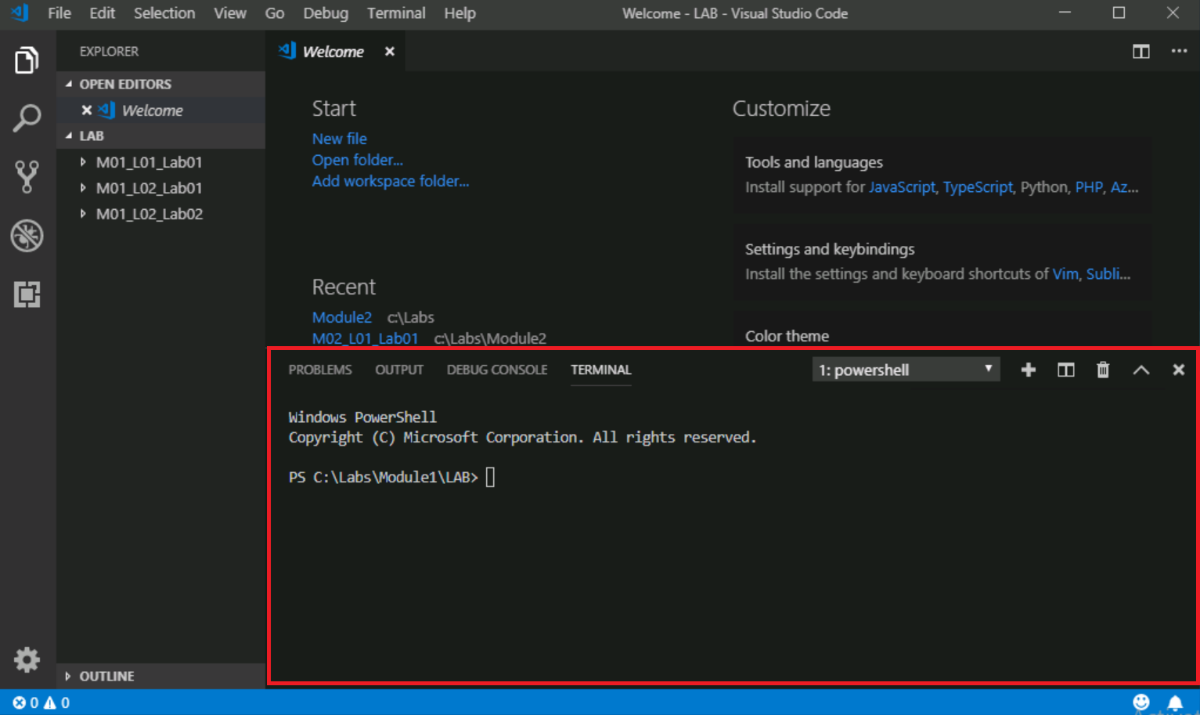
1. Then, click the **Select Folder** button. When the folder is opened, an explorer appears in the **Side bar**.



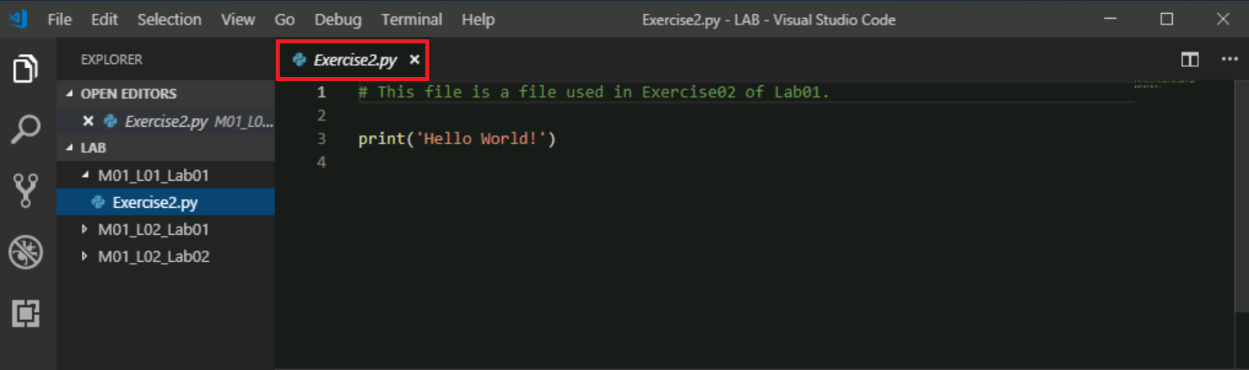
1. View Terminal
2. VS Code integrates PowerShell as a Terminal. The terminal can be used in a variety of ways depending on the language being used for development. You can view the terminal by selecting **Terminal** from the **View** menu.



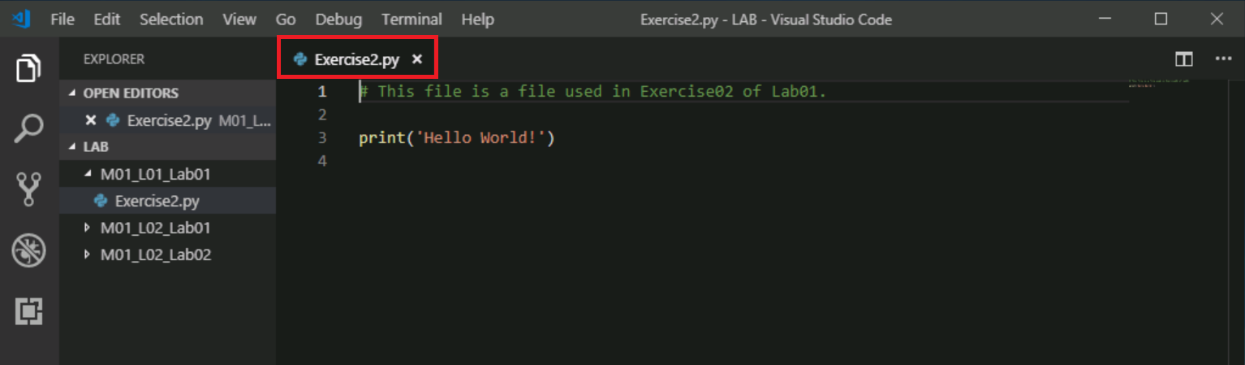
1. The terminal appears in the lower-right corner of the window.



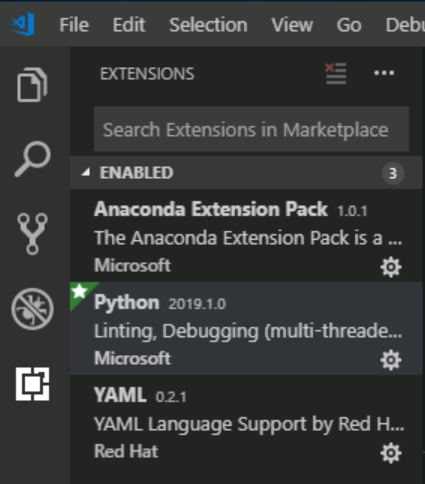
1. Open File
2. Click the Explorer in the **Side bar** and expand a folder in the Explorer. When you **single-click** a file name, VS code opens the file in **reading mode**. In reading mode, selecting a different file overwrites the tab, and the file name on the tab is displayed in italic format.



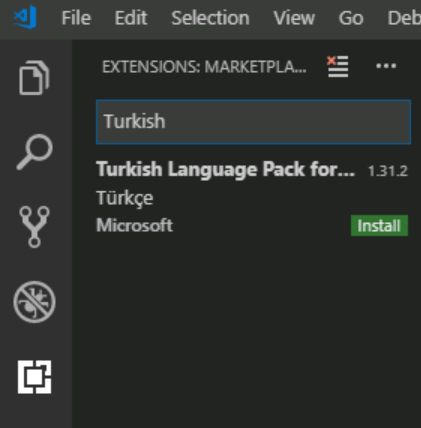
1. If you edit a file opened in reading mode or **double-click** a file name in Explorer or double-click the tab of entire file, the file is in **edit** mode. If you select another file, the tab will not be overwritten, and the file will be opened in a separate tab.



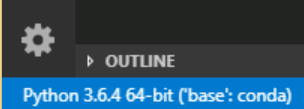
1. View Extension
2. VS Code is a development environment that supports a variety of languages. If you use a programming language on VS code, you can efficiently code it by installing an extension.
3. If you installed vs code at the same time as the Anaconda installer, you already have an extension for Python installed.

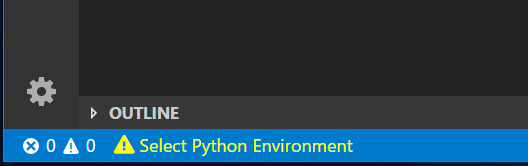


1. If you want to install new extensions, use the search bar to find them. For example, try searching for a Language Pack extension. As an example, we are searching for a Turkish language pack (but we also provide language packs for other languages).

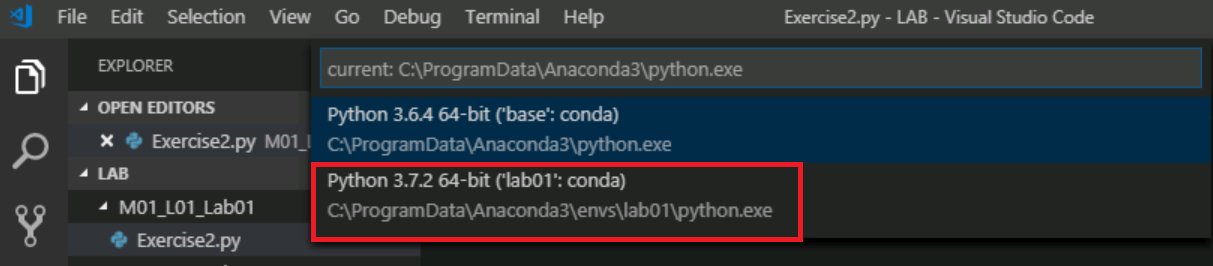


1. Select Runtime
2. VS Code allows you to switch between different python environments for development. Clicking on the python environment displayed in the status bar (or warning indicating that you need to select a Python environment) opens the command palette, and you can select a **Python environment**.

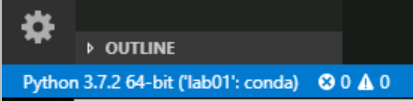




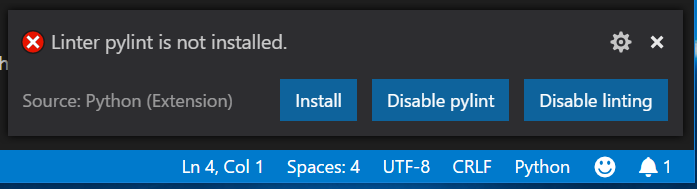
1. In this section, please select the Lab01 virtual environment that you created in Exercise 1.



1. When the environment changes, the display of the status bar is also changed.

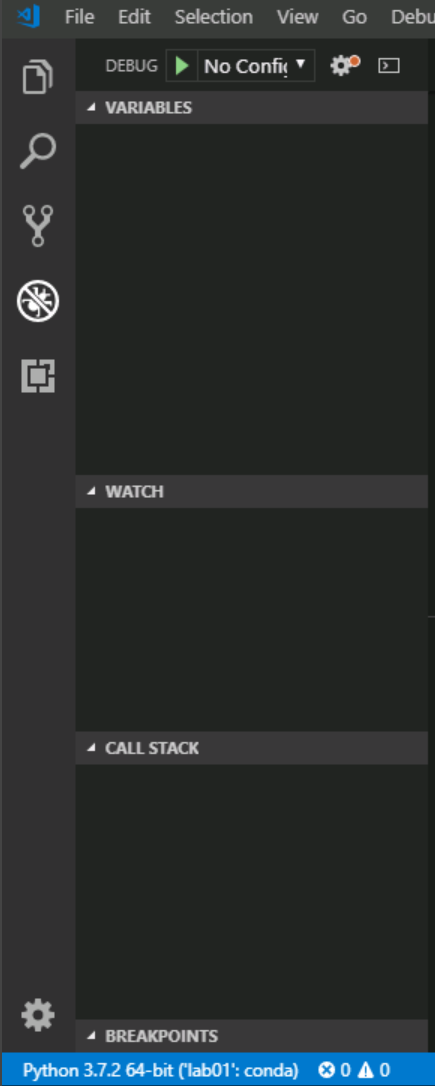


In switching the Python environment, you may receive a warning stating that linter is not installed. Please ignore the warning for this lab.



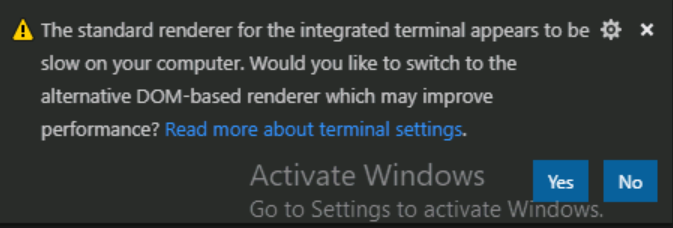
If you would like to get more information on linting: <https://code.visualstudio.com/docs/python/linting>

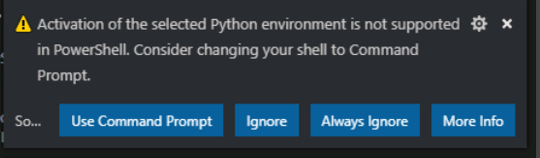
1. View debug settings
2. If you are debugging a coded script, select the **Debug** icon from the Activity bar. The Side bar displays a space for checking **local variables**, **watch expressions**, **call stack** and **break points**.



1. Warnings with VS Code

Safely ignore the warnings below if you encounter them during these labs or in the upcoming labs:





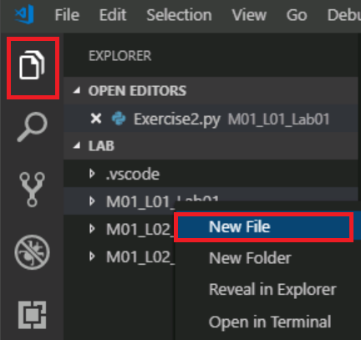
Exercise 2 has been completed.

Exercise 3: Running a code

In this exercise, you'll learn how to run a Python script on VS code.

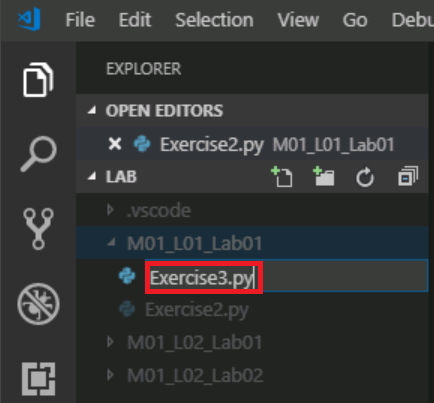
Tasks

1. Create File
2. Go back to the Explorer tab from the Side bar. **Right-click** the **M01\_L01\_Lab01** folder on the side bar and click on the **New file** button. (This exercise does not use an existing file and creates a new file.)



1. For the new file created, type the following file name:

Exercise3.py



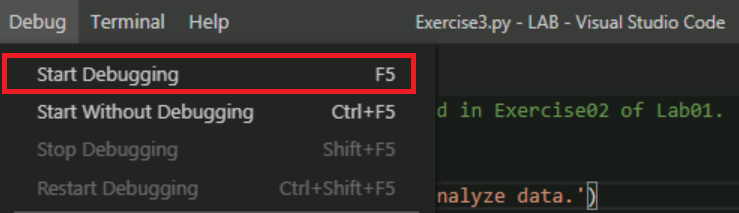
1. When the file is created, it will be opened automatically in the editor.
2. Coding

Let's enter the code to run. As much as possible, it's a good idea to type in your code to see IntelliSense support. Please enter the code below and save the file.

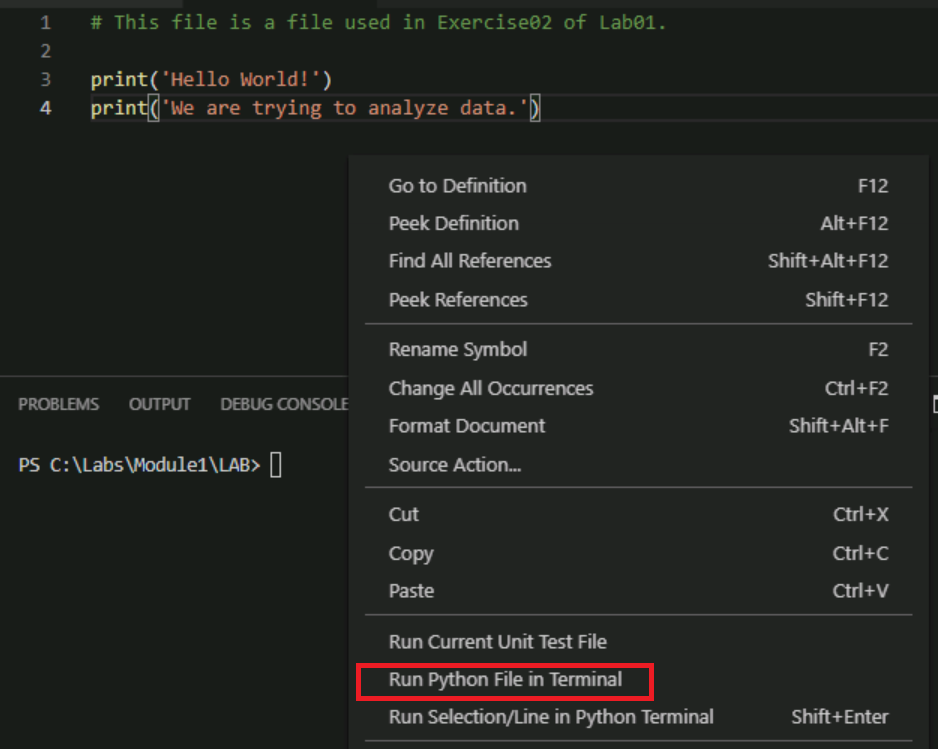
print('Hello World!')

print('We are trying to analyze data.')

1. Running the script file
2. There are two ways to perform a script file.
3. If debugging is required, choose **Start debugging** from the **Debug** menu.



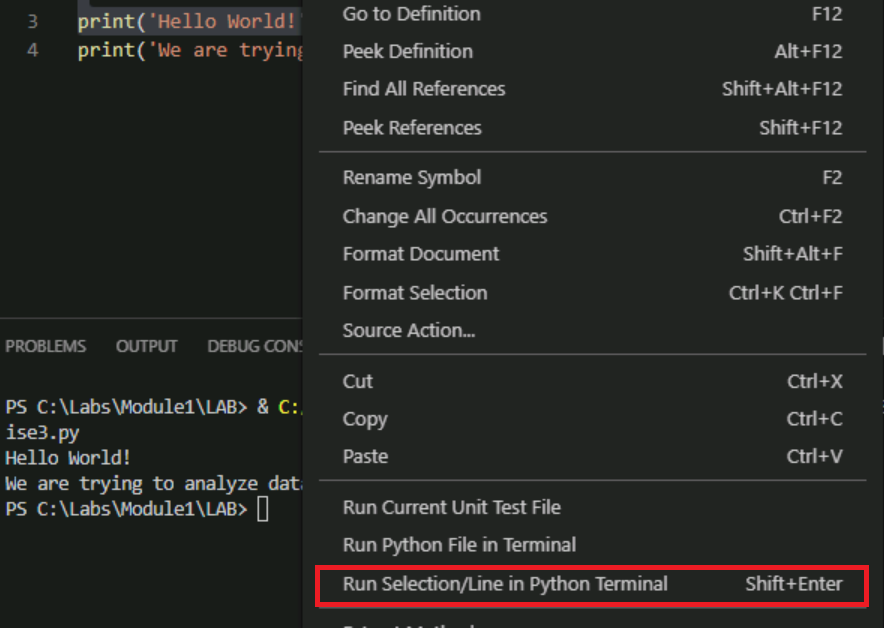
1. If you just want to run the entire script, right-click in the editor and select **Run Python File in Terminal**.



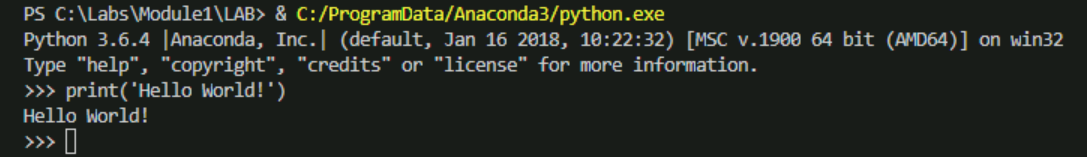
1. The output of the script will appear in the terminal.



1. Running chunk of code
2. You can transfer the code to the execution environment and keep the results in the execution environment's memory.
3. Try running only the first line of the code you just created. Select the first line with the mouse to highlight it. Then right-click the selection and select **Run Selection/Line in Python Terminal**.

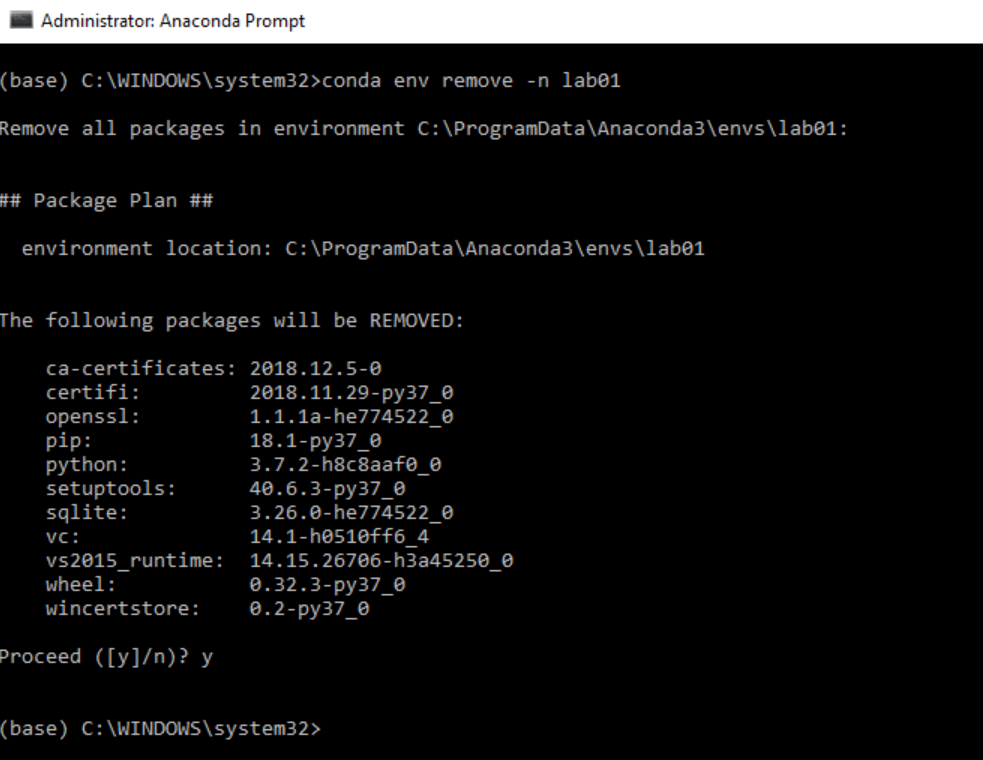


1. In the terminal, Python starts, the first line of code is transferred, and the results of the execution are displayed in the terminal.



1. In this state, if you run the second line of code in an equivalent operation, the second line will be forwarded to the terminal and the results will be displayed.
2. Clean up
3. Remove the lab01 conda environment with the following command and respond to Proceed y/n with **y**.

conda env remove -n lab01



Exercise 3 has been completed.