



Advanced Concepts in Data Analytics

Lab: Introducing Python and CSV Files

This page was intentionally left blank.

Table of Contents

Introducing Python and CSV Files.....	1
Introduction	1
Equipment and Materials	1
Lab Activity 1: Installing Python and Visual Studio Code	2
Lab Activity 2: Hello World	5
Lab Activity 3: Working with CSV files	11
References	14

Introducing Python and CSV Files

Introduction

Python is a dynamically typed, general-purpose programming language widely used in applications such as web development, data analysis, machine learning and computer vision. Python is powerful and yet readable and easy to learn, which makes it one of most popular programming languages for people of all backgrounds, with its great and fast-growing ecosystem of libraries, frameworks and tools. You'll use Python exclusively for all examples and lab assignments throughout this course.

Visual Studio Code is an open source code editor developed by Microsoft. It was ranked the most popular developer environment tool according to Stack Overflow Developer Survey 2019. You may have other favorite IDEs or editors; however, you're strongly recommended to use Visual Studio Code for code editing to demonstrate your level of use of the software, which is an important marking criterion for all labs.

In this lab, we will practice on Python basic syntax, data types, control flows, input and output, modules and a few standard Python libraries. If you are new to Python, you may not be able to learn everything about Python in one lab. Please refer to the reading source list for more Python tutorials. If you speak fluent Python, use this lab as a refresher.

Equipment and Materials

- BYOD laptop
- Python installer (instructions below)
- Visual Studio Code System installer (instructions below)
- CSV input file

Lab Activity 1: Installing Python and Visual Studio Code

To Install Python

1. Before you start, ensure that you have internet connection to download the executable files below.
2. Open your web browser and go to the [Python 3.7.6 download page](https://www.python.org/downloads/release/python-376/) (<https://www.python.org/downloads/release/python-376/>).
3. Scroll to the “Files” section, and then select the appropriate installer for your device (for example, the **Windows x86-64 executable installer**).
4. Execute the installer.

The *Python Setup* window appears.

5. Make sure the **Add Python 3.7 to PATH** checkbox is selected. This ensures that Python is accessible from command line.



Figure 1: Python Installer Setup Window

©2001-2020 Python Software Foundation. All rights reserved. Used with permission.

6. Click **Install Now**.

When the installation completes, the *Setup was successful* window appears.

7. Select **Disable path length limit**, as shown.



Figure 2: Successful Setup Window

©2001-2020 Python Software Foundation. All rights reserved. Used with permission.

8. To verify that Python is working properly, open a command line program, such as Command Prompt or MS PowerShell on Windows, and then type:

```
python -c "print ('Hello World')"
```

The text "Hello World" should appear in the command line, as shown below.

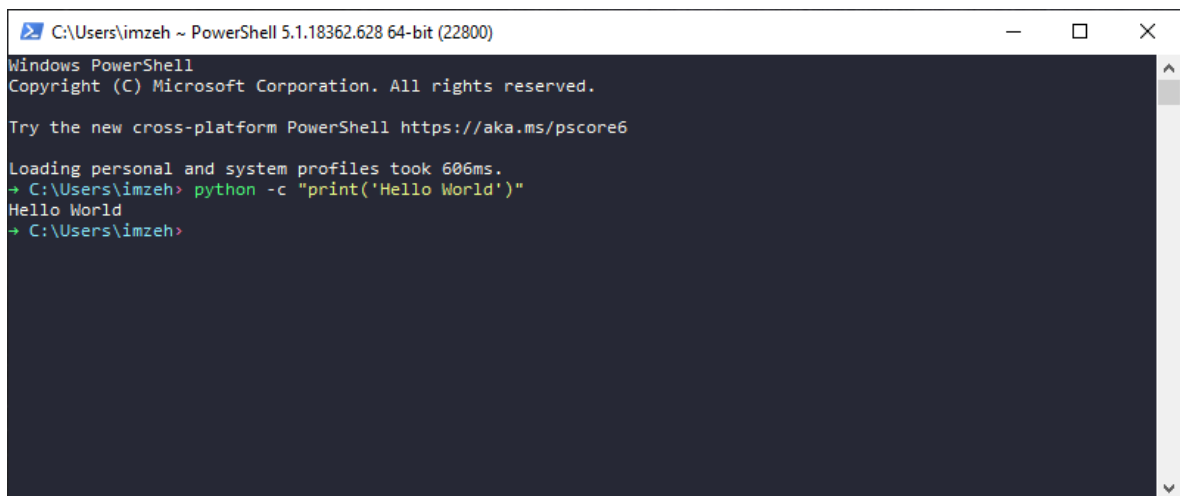


Figure 3: Testing Python from a Command Line

Used with permission from Microsoft.

To Install Visual Studio Code

1. Go to [Download Visual Studio](https://code.visualstudio.com/download) (<https://code.visualstudio.com/download>) and download the appropriate installer for your device.
2. Execute the installer.
The *Setup* window appears.
3. Accept the license agreement and all other default settings.
4. Once the installation is complete, install the Python extension for VS Code by following the instructions on the [Getting Started with Python in VS Code website](https://code.visualstudio.com/docs/python/python-tutorial) (<https://code.visualstudio.com/docs/python/python-tutorial>) for Python development.

The VS Code interface appears as follows.

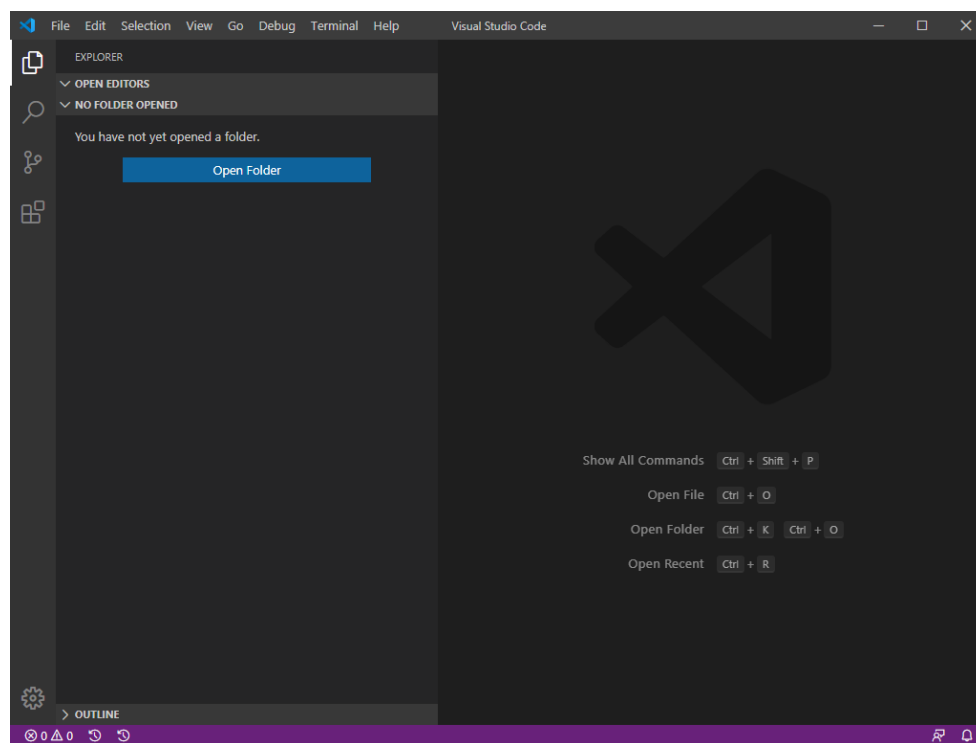


Figure 4: Visual Studio Code Interface

Used with permission from Microsoft.

Lab Activity 2: Hello World

In this section, you'll use Python to create a command line program that:

- Asks for your name
 - Asks for your year of birth
 - Greets you with "Hello"
 - Calculates and tells you your age
 - Says "Goodbye" at the end of program
1. Create a folder on your computer where you'll store the files for this lab (e.g., **lab1**).
 2. In VS Code, click **File > Open Folder** and open the folder you created.

Your screen appears as follows.

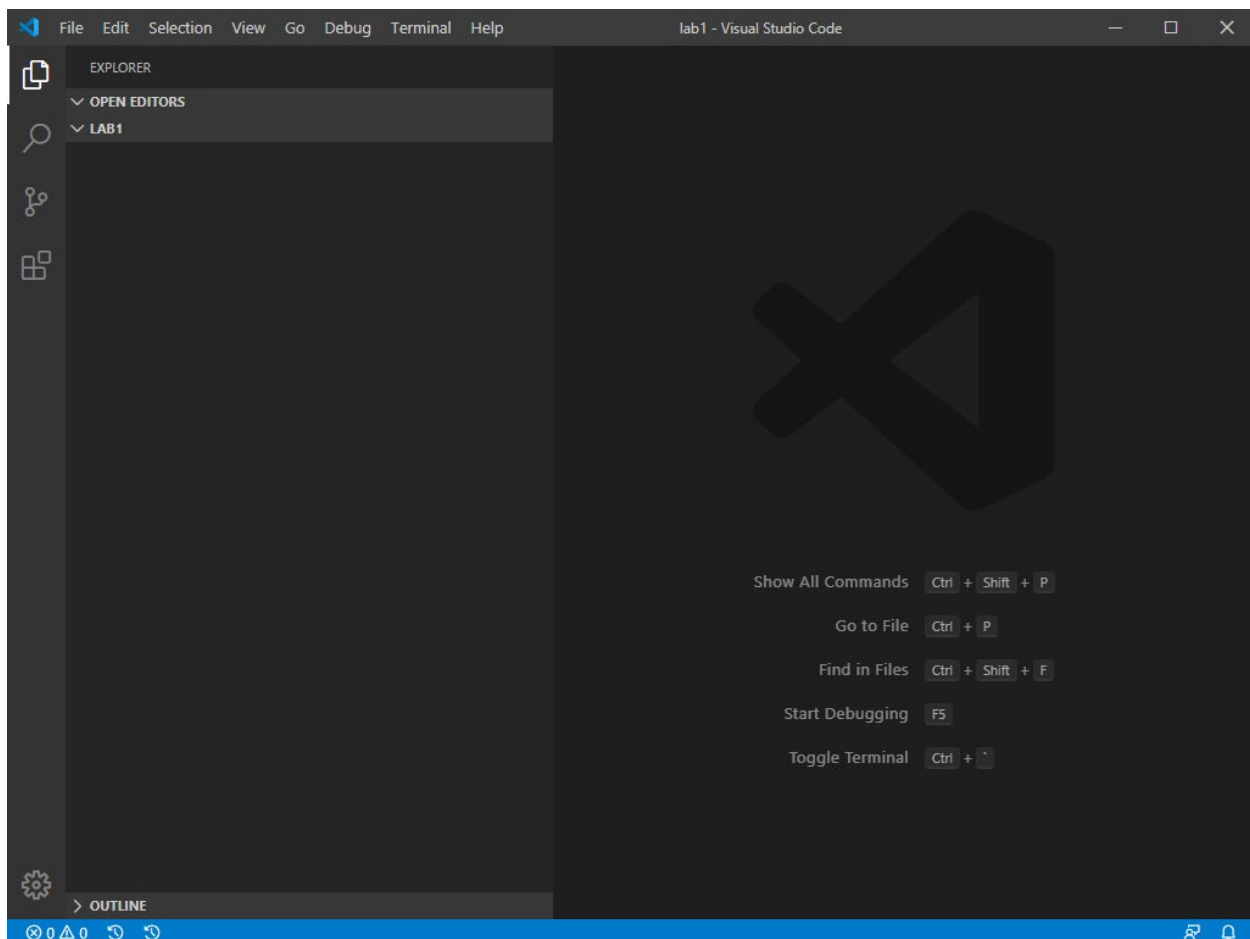


Figure 5: VS Code After Opening Folder

Used with permission from Microsoft.

3. Click **File > New File** and create a new file named **hello_world.py**.
4. Type your first line of code so that it appears as shown below, and then type similar line of code for birth year input.

Notes:

- Using third-party libraries is not allowed. Only use standard libraries.
- Because your program requires two pieces of information from users (their names and birth years), look for a built-in function on [The Python Standard Library website](https://docs.python.org/3.7/library/index.html) (<https://docs.python.org/3.7/library/index.html>) that reads from user input and stores it in the program. The built-in function is called [input\(\[prompt\]\)](https://docs.python.org/3.7/library/functions.html#input) (<https://docs.python.org/3.7/library/functions.html#input>).
- Also keep in mind that `input ([prompt])` returns user's input in a string.

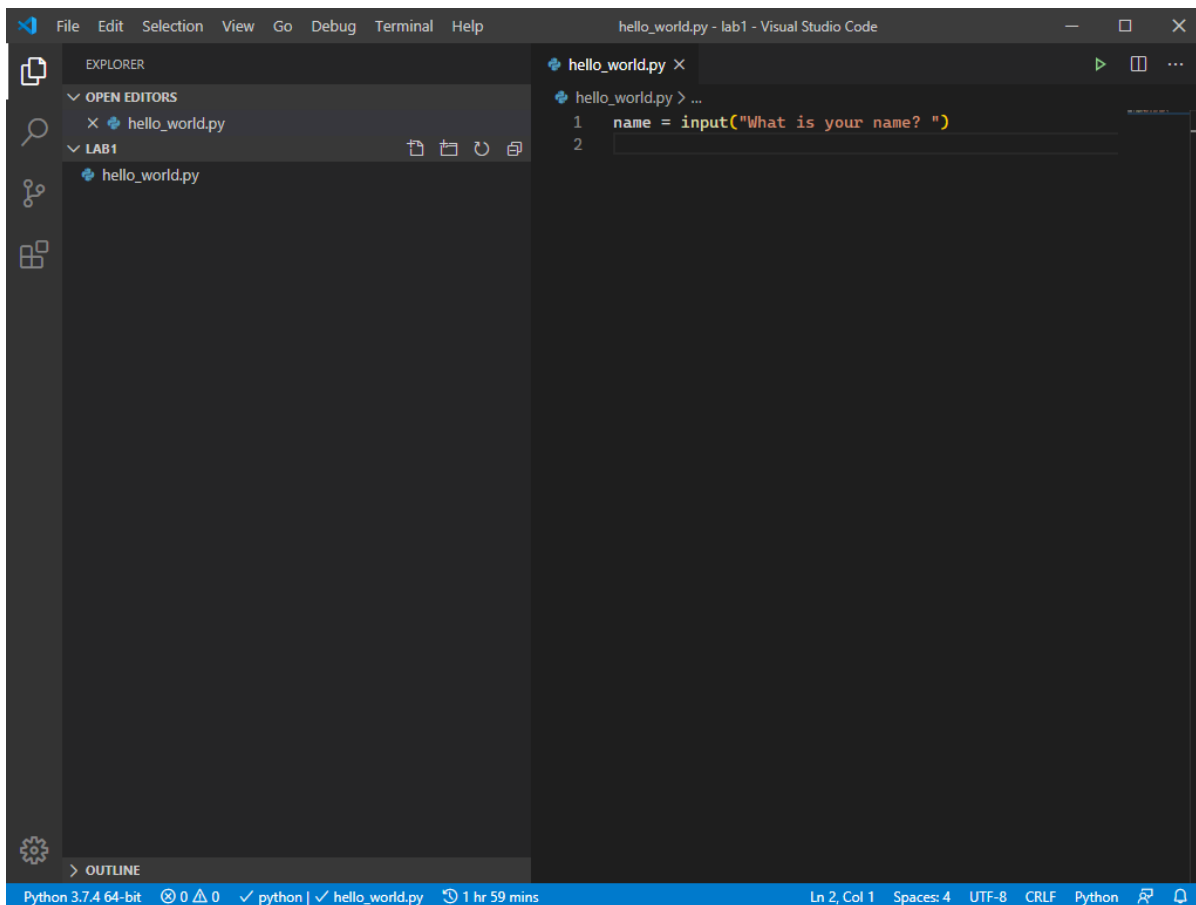
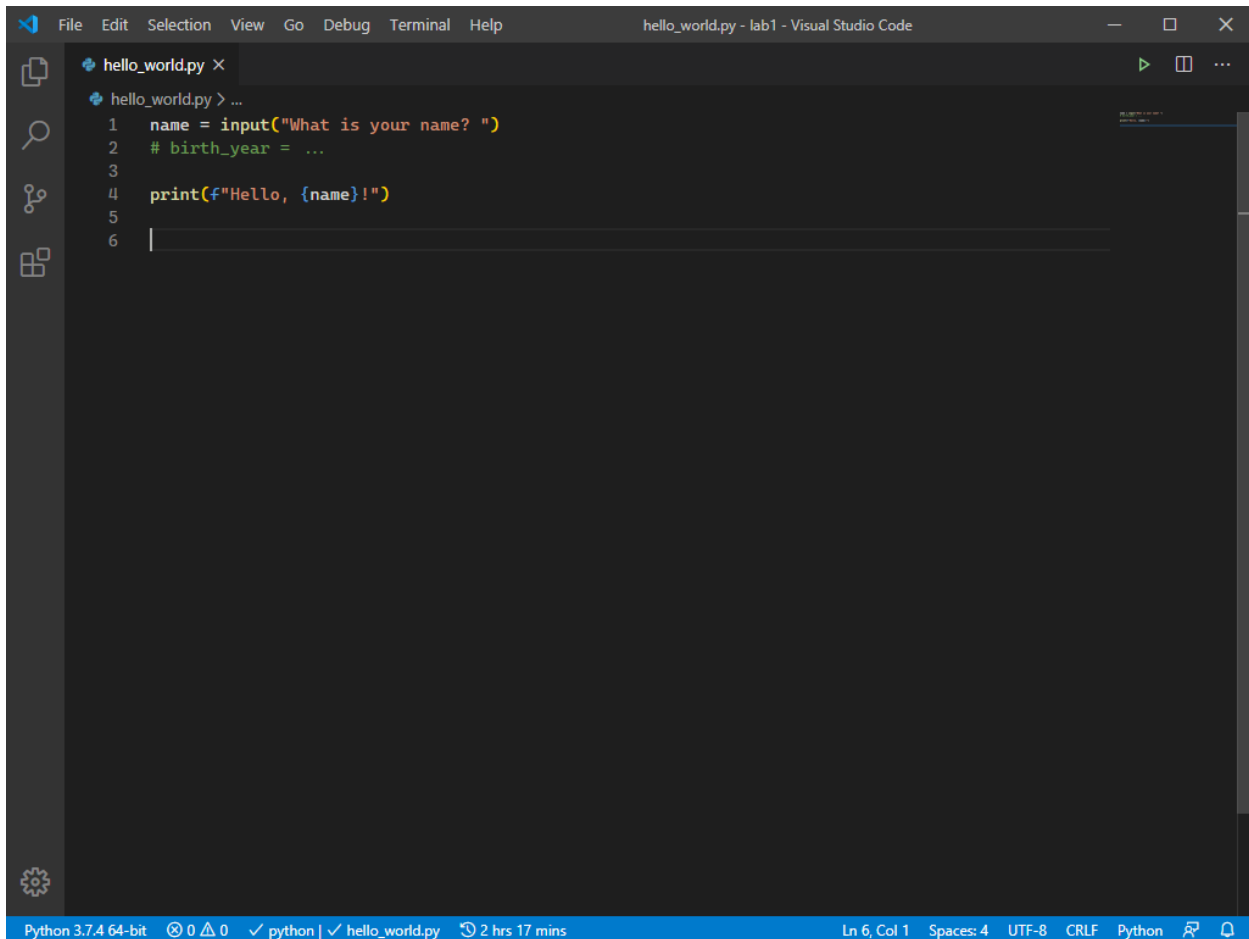


Figure 6: VS Code First Line
Used with permission from Microsoft.

5. Once the user's name is stored in the program, greet the user by prompting a custom message with the user's name in it. For example, you can print out "Hello, John!" if the user enters the name as **John**, as shown below.

Notes:

- Python supports multiple formats for string literal.
- It's strongly recommended that you use f-strings, which were introduced in Python 3.6. For more detail, see [Literal String Interpolation](https://www.python.org/dev/peps/pep-0498/) (<https://www.python.org/dev/peps/pep-0498/>).



```
File Edit Selection View Go Debug Terminal Help hello_world.py - lab1 - Visual Studio Code
hello_world.py x
hello_world.py > ...
1 name = input("What is your name? ")
2 # birth_year = ...
3
4 print(f"Hello, {name}!")
5
6 |
```

Python 3.7.4 64-bit 0 0 python | hello_world.py 2 hrs 17 mins Ln 6, Col 1 Spaces: 4 UTF-8 CRLF Python

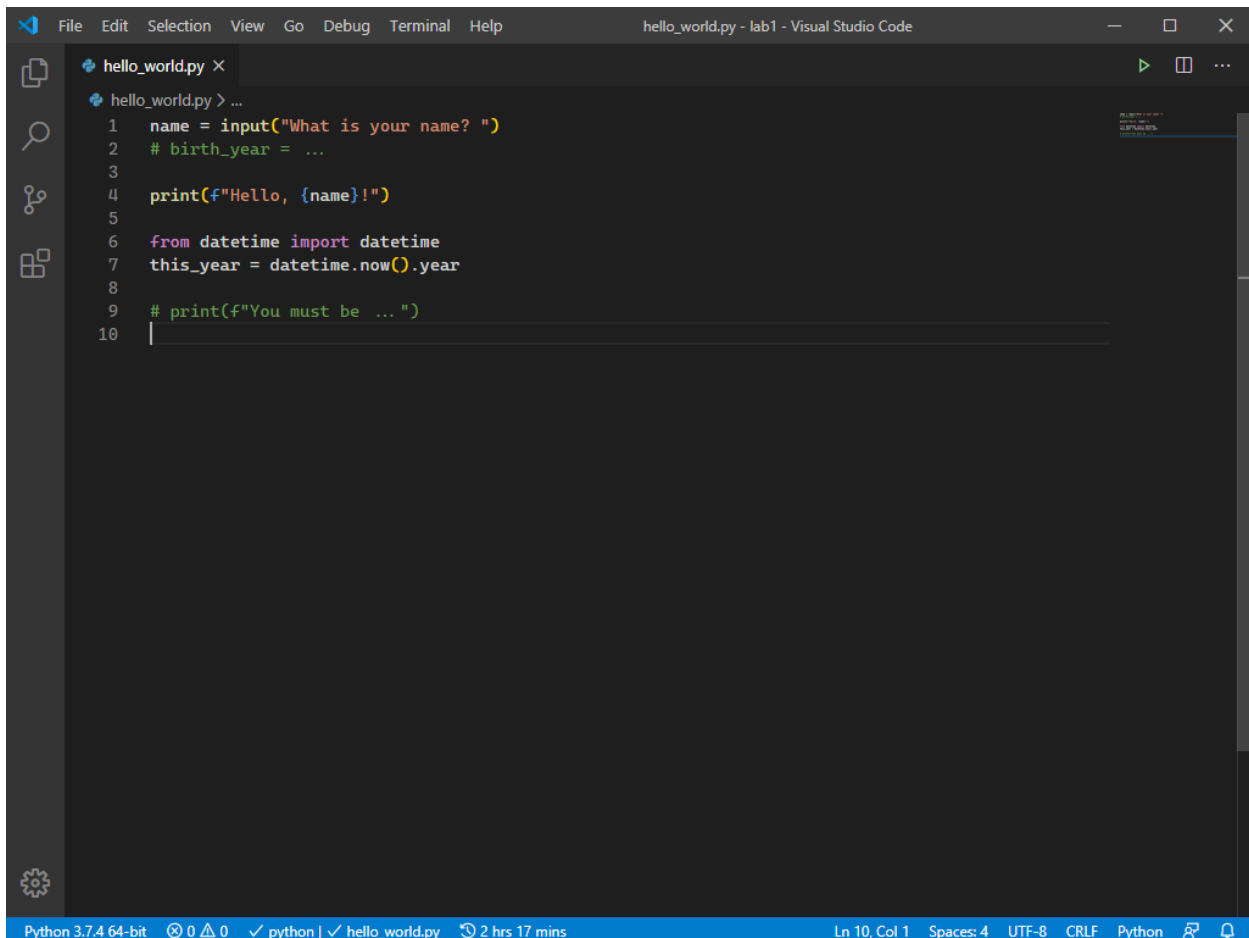
Figure 7: VS Code Print Greeting Message

Used with permission from Microsoft.

- To calculate the user's age, the program needs to know what year it is when it runs. Search Python's official documentation for a **datetime** module, which handles the basic date and time types.

Note:

- The `datetime` class has a `now()` function that returns the local date and time. For more detail, see [Basic date and time types](https://docs.python.org/3.7/library/datetime.html#datetime.datetime.now) (<https://docs.python.org/3.7/library/datetime.html#datetime.datetime.now>).
- Access the `year` attribute on the `datetime` instance, and then apply simple math to calculate the user's age. The code to get the current year is shown below.



```
File Edit Selection View Go Debug Terminal Help hello_world.py - lab1 - Visual Studio Code
hello_world.py x
hello_world.py > ...
1 name = input("What is your name? ")
2 # birth_year = ...
3
4 print(f"Hello, {name}!")
5
6 from datetime import datetime
7 this_year = datetime.now().year
8
9 # print(f"You must be ...")
10 |
```

Python 3.7.4 64-bit 0 0 ✓ python | ✓ hello_world.py 2 hrs 17 mins Ln 10, Col 1 Spaces: 4 UTF-8 CRLF Python

Figure 8: VS Code Get Current Year
Used with permission from Microsoft.

8. Execute your Python program within Visual Studio Code by clicking **View > Terminal**.

The terminal window appears at the bottom of the screen.

9. Run your **hello_world.py** program by typing **python hello_world.py** in the terminal window, as shown below.

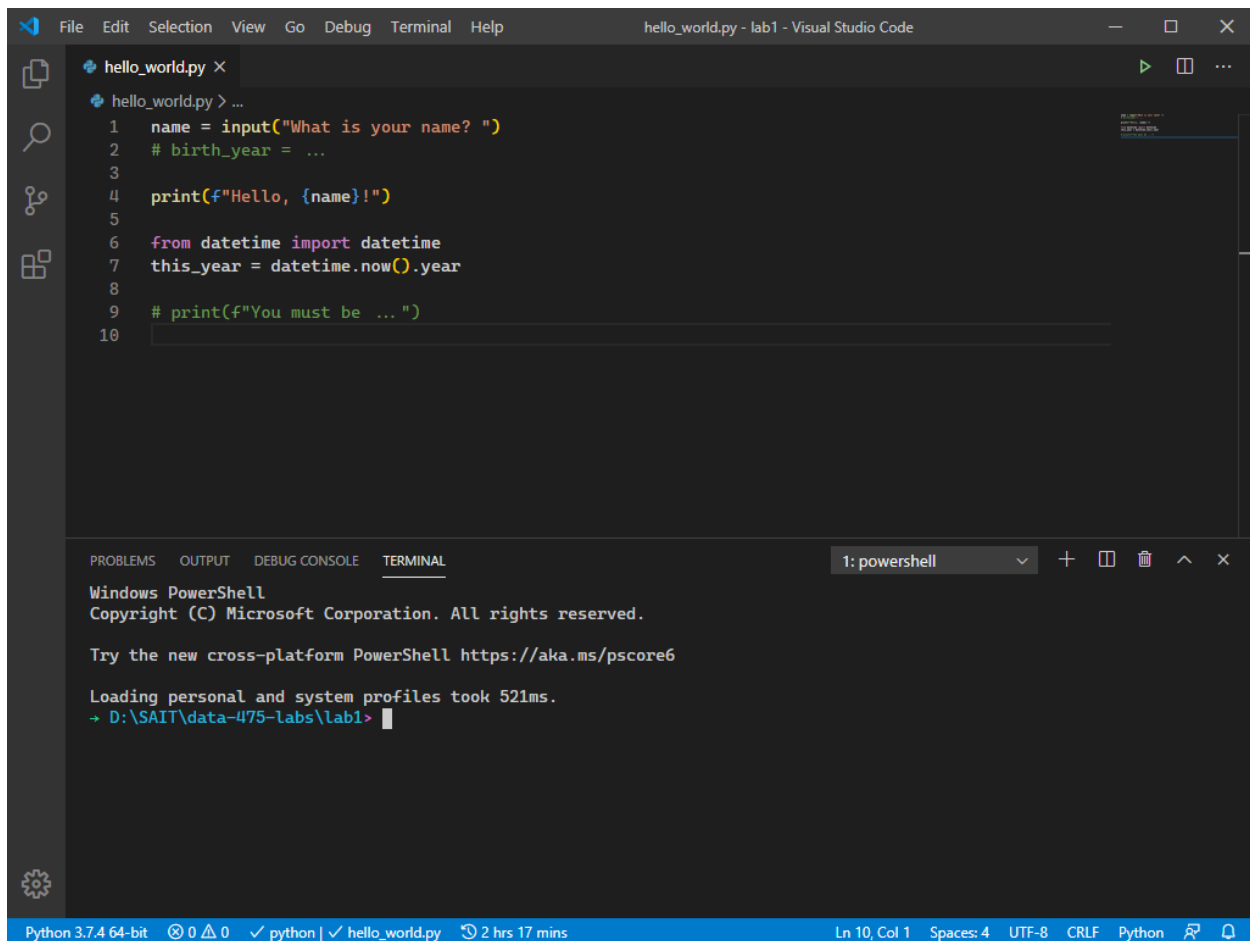
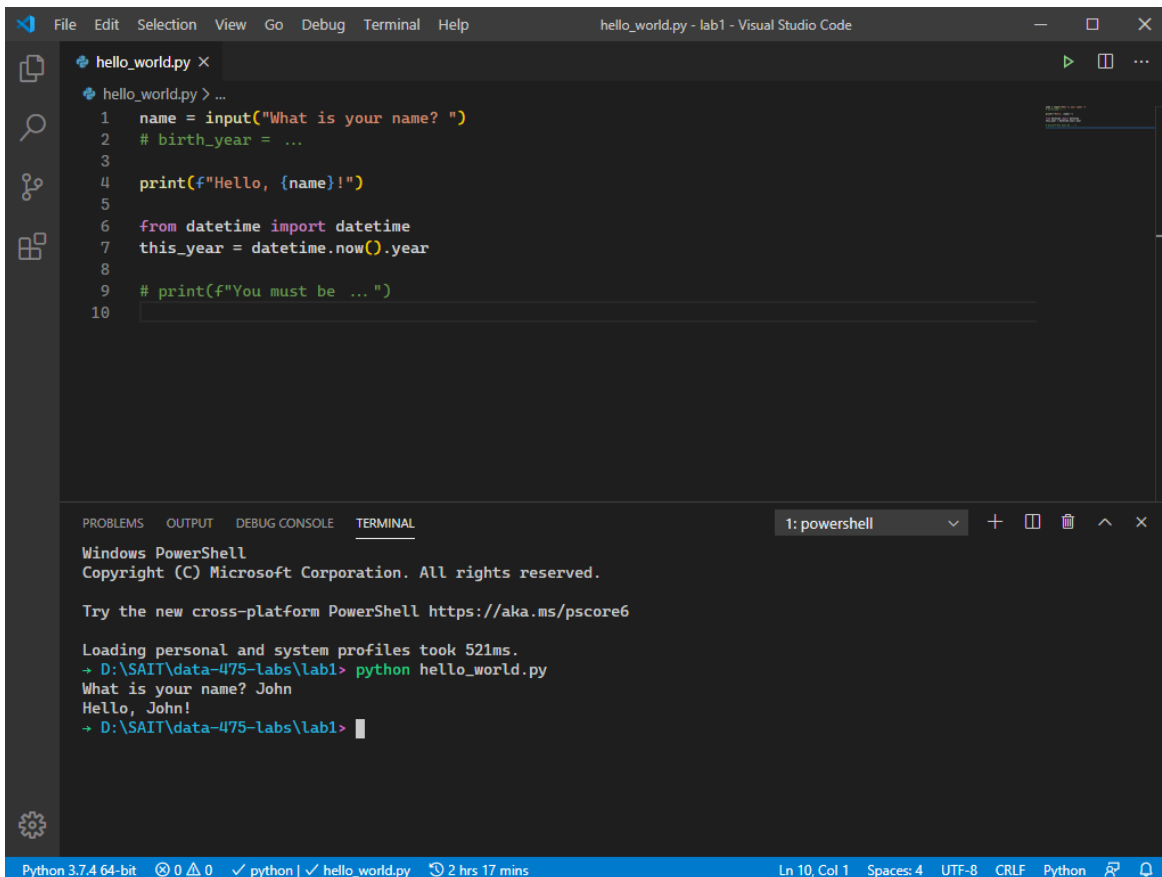


Figure 9: VS Code Terminal Opened

Used with permission from Microsoft.



```

File Edit Selection View Go Debug Terminal Help
hello_world.py - lab1 - Visual Studio Code

hello_world.py X
hello_world.py > ...
1 name = input("What is your name? ")
2 # birth_year = ...
3
4 print(f"Hello, {name}!")
5
6 from datetime import datetime
7 this_year = datetime.now().year
8
9 # print(f"You must be ...")
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

Loading personal and system profiles took 521ms.
+ D:\SAIT\data-475-labs\lab1> python hello_world.py
What is your name? John
Hello, John!
+ D:\SAIT\data-475-labs\lab1>
Python 3.7.4 64-bit 0 0 0 ✓ python | ✓ hello_world.py 2 hrs 17 mins Ln 10, Col 1 Spaces: 4 UTF-8 CRLF Python

```

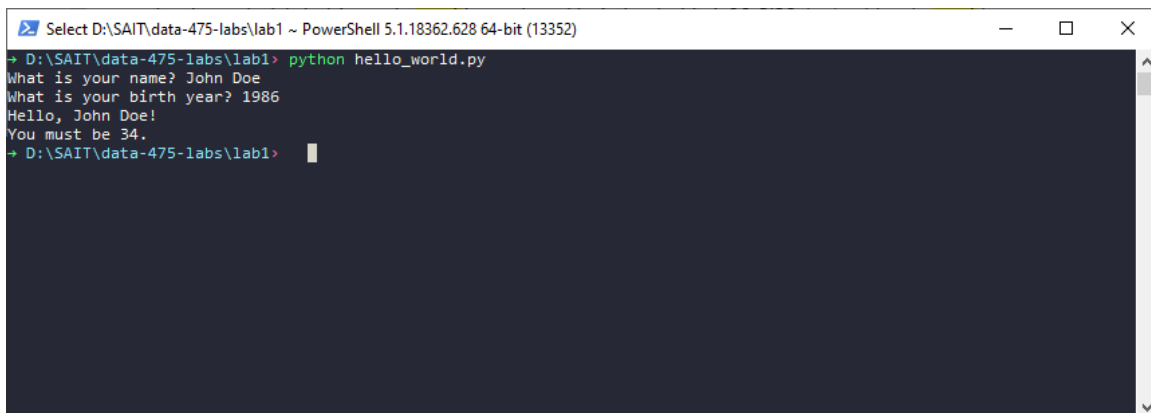
Figure 10: VS Code Hello World (Partial Code) Executed

Used with permission from Microsoft.

Complete the **hello_world.py** program by typing the missing two statements. The final output should resemble the screen below.

Note:

- The prompt message can be slightly different, as long as it contains the user's name and the correct age value.



```

Select D:\SAIT\data-475-labs\lab1 ~ PowerShell 5.1.18362.628 64-bit (13352)
+ D:\SAIT\data-475-labs\lab1> python hello_world.py
What is your name? John Doe
What is your birth year? 1986
Hello, John Doe!
You must be 34.
+ D:\SAIT\data-475-labs\lab1>

```

Figure 11: Final Output

Used with permission from Microsoft.

Lab Activity 3: Working with CSV files

There are far fewer example codes for this lab activity, and fewer step-by-step instructions, so use it as your first Python code challenge. However, let your instructor know if you are having difficulty. As before, using third-party libraries is not allowed. Only use standard libraries.

In this activity, you'll write a command line program to work with a comma-separated values (CSV) file, which stores data in a table. You'll create a program that:

- Reads the content of a CSV file
 - Counts and prints out the total count of rectangles
 - Calculates and prints out the total, average, maximum and minimum area of all rectangles
 - Stores the above information in another CSV file named **summary.csv**
1. Create a new file named **rectangle_summarizer.py** in the same lab folder you created earlier. Refer to Lab Activity 2, Step 3 for reference.
 2. Open the **rectangles.csv** file in Microsoft Excel. Notice that the CSV file has a heading, as well as three columns: **id**, **width**, and **length**. Each row in the CSV file stands for a rectangle. Both width and length units are in metres.
 3. For your program to work correctly, it must either skip the first row when reading the CSV file or delete the first row after reading the CSV file. There are many solutions you can choose.

Hint: Try using the built-in `next()` function to get the first line from the CSV file but not do anything with it.

Notes:

- Python comes with a `csv` module to handle CSV files. For more detail, see [CSV File Reading and Writing](https://docs.python.org/3.7/library/csv.html) (<https://docs.python.org/3.7/library/csv.html>), particularly the `reader()` and `writer()` functions.
- The Python `with` statement refers to context management when executing the block inside. It's used to handle desired entry and exit runtime. In the context of the file object `open()`, it ensures that the specific files can be read upon entry, and that the file stream is closed before exit.
- Use `newline=""` as the keyword argument for both `reader` and `writer` functions, since the `csv` module does its own universal newline handling (see the footnotes on the webpage above).

An example code snippet is shown below.

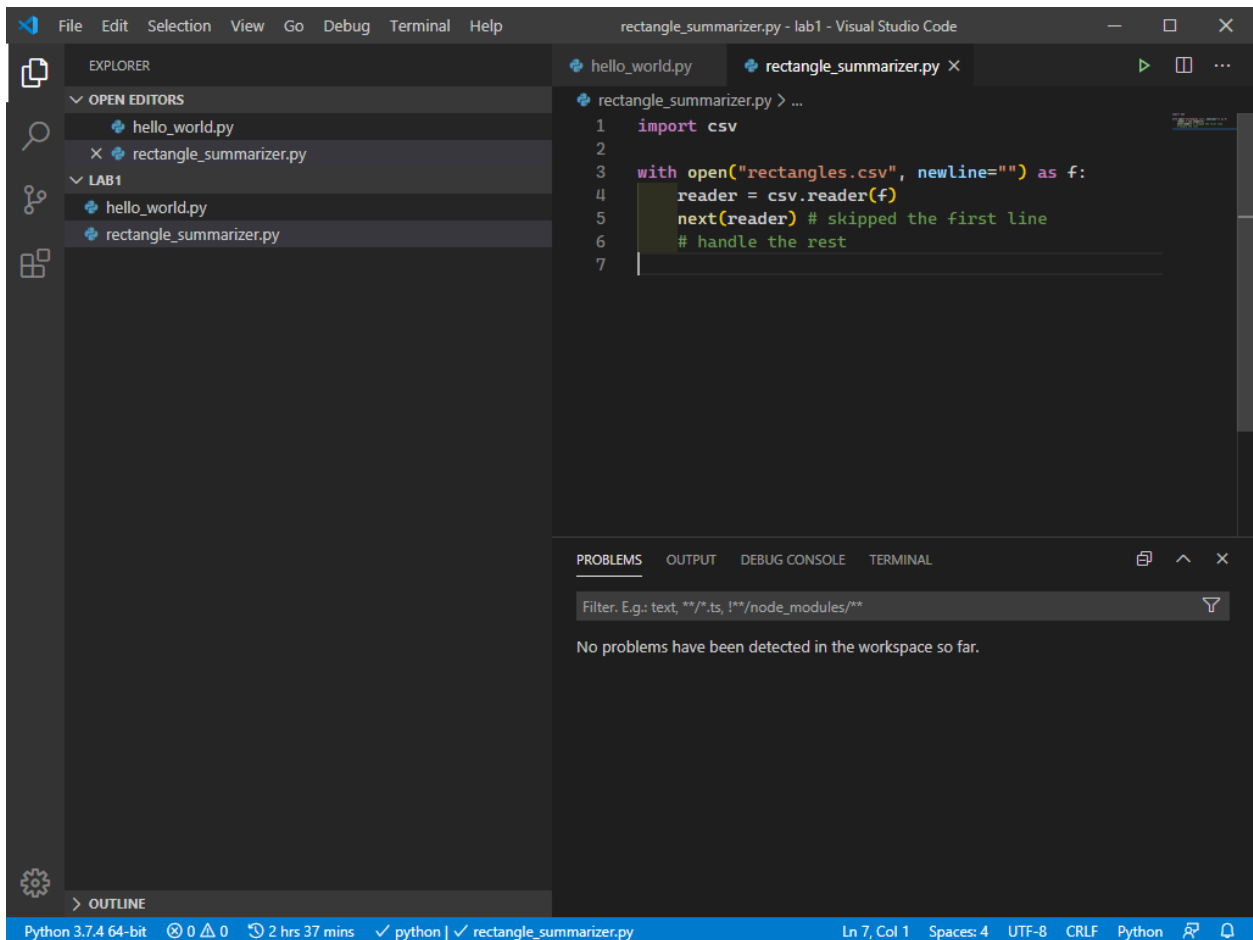


Figure 12: Visual Studio Code First Lines of Working with CSV Files

Used with permission from Microsoft.

4. Iterate the whole CSV file, compute the area of the rectangles using the width and length values, and store the area of each rectangle in a Python list variable.
5. Get the desired statistics from the list variable you just created.

Notes:

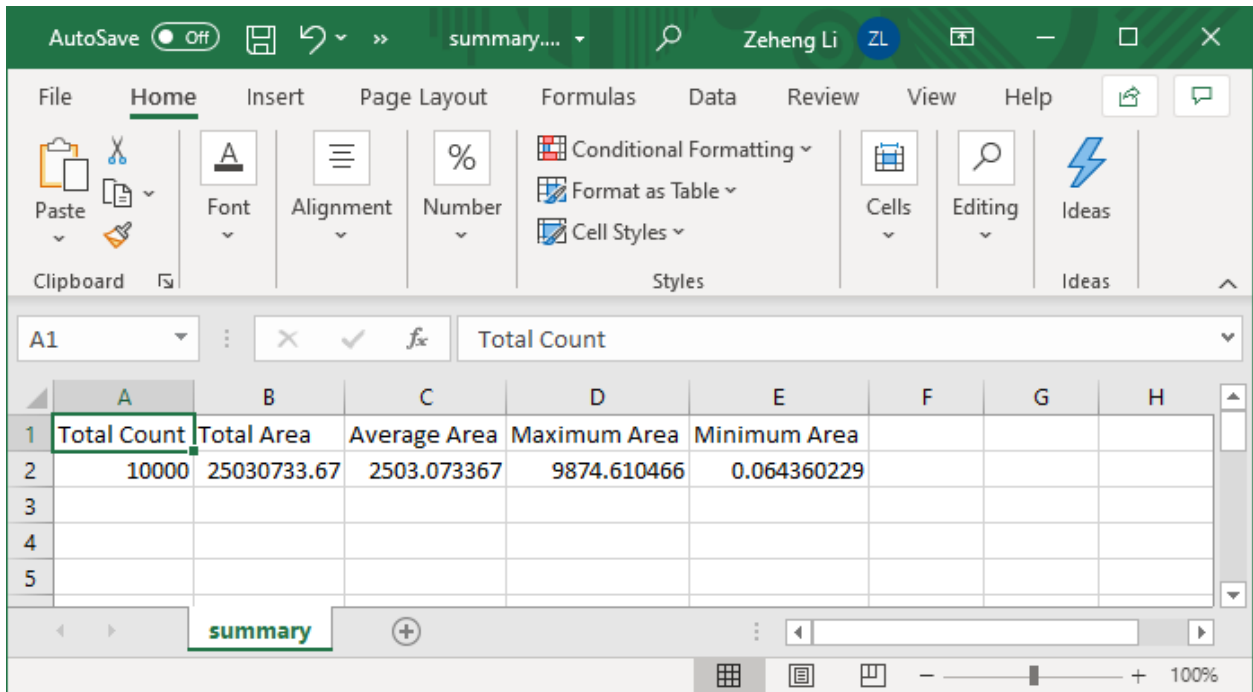
- You can use `min()` and `max()` to get the minimum and maximum value from a list. But there is no built-in function to get the average value from a list.
 - Python comes with a `statistics` module where you can use `mean()` to get the expected result. For more detail, see the entry at [Mathematical statistics functions](https://docs.python.org/3.7/library/statistics.html) (<https://docs.python.org/3.7/library/statistics.html>). Or you can compute it manually.
6. Use the `with` statement to open a file named **summary.csv** and write the statistics in it. The **summary.csv** file should have one row with five columns, namely, **total count**, **total area**, **average area**, **maximum area**, and **minimum area**. The final output should resemble the screen below.

```

D:\SAIT\data-475-labs\lab1 ~ PowerShell 5.1.18362.628 64-bit (21464)
→ D:\SAIT\data-475-labs\lab1> python .\rectangle_summarizer.py
Total Count: 10000
Total Area: 25030733.66941007
Average Area: 2503.0733669409874
Maximum Area: 9874.610465755648
Minimum Area: 0.0643602289153509
→ D:\SAIT\data-475-labs\lab1>

```

Figure 13: Final Output (Command Line)
Used with permission from Microsoft.



	A	B	C	D	E	F	G	H
1	Total Count	Total Area	Average Area	Maximum Area	Minimum Area			
2	10000	25030733.67	2503.073367	9874.610466	0.064360229			
3								
4								
5								

Figure 14: Final Output (CSV File)
Used with permission from Microsoft.

References

- Stack Overflow Developer Survey 2019. (n.d.). Retrieved from <https://insights.stackoverflow.com/survey/2019#development-environments-and-tools>
- Microsoft (2020). Visual Studio Code [Computer Software]. Retrieved from <https://code.visualstudio.com/download>
- Python Software Foundation (2020). Python (Version 3.7.6) [Computer Software]. Retrieved from <https://www.python.org/downloads/release/python-376>