**The PHP Language**

Naga Jyothi Madamanchi

***Abstract—*** The purpose of this project is to analyze the purpose and installation process of the PHP language. The compilation process and uses of this language will be compared to languages such as Javascript and Python.

***Index Terms—*** PHP, JavaScript, Homebrew, HTML, Htaccess, Apache, SQL, Python, Anaconda, Jupyter-notebook

**I. Introduction**

The PHP language has served as an essential tool for web developers since its conception in 1994. Although the language it is not as widely used as it once was, its versatility and straight forward syntax makes it an ideal language for new programmers.

Web developers often compare the PHP language to JavaScript, another highly popular language that serves a similar purpose to the aforementioned language. Both languages are used extensively for a variety of functions ranging from web user interfaces to the display of graphs [3].

**II. PHP Installation Windows/Linux**

The installation process for PHP is a relatively simple process on the Windows and the Linux operating system. The first step begins with the download of the file “php-8.1.12.tar.gz” from “php.net”. Once the tar.gz file is in the download folder. The next step involves the use of terminal. To install type the following command:

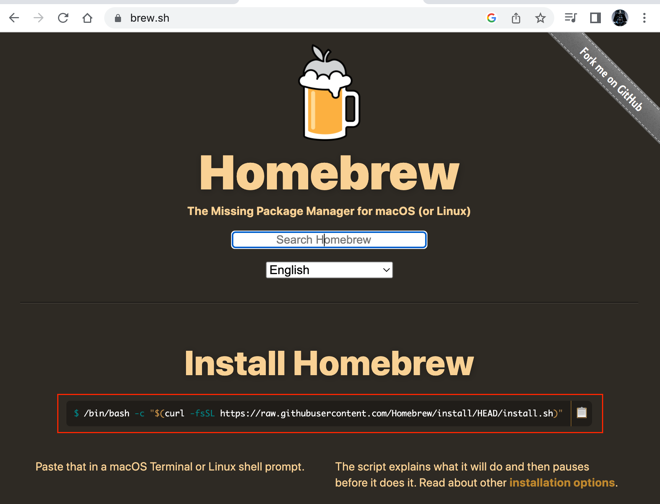
Windows PHP installation command:

sudo apt install php

Terminal will ask you if you want to proceed with the installation. Type ‘y’ to choose yes as your response. Upon successful installation, you will be able to verify where PHP was installed by typing “what php”. You also have the option to install a PHP add-on in Visual Studio Code [2].

**III. PHP Installation Mac**

In the MacOS operating system, the installation of the PHP language requires the use of the use of the program Homebrew to install the “php-8.1.12.tar.gz” file. The command highlighted by the red box in the image below must be copied into terminal to install Homebrew.



Homebrew installation command:

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

Once Homebrew has been installed, the next step is to type in the command below to install PHP with the .tar.gz file [2].

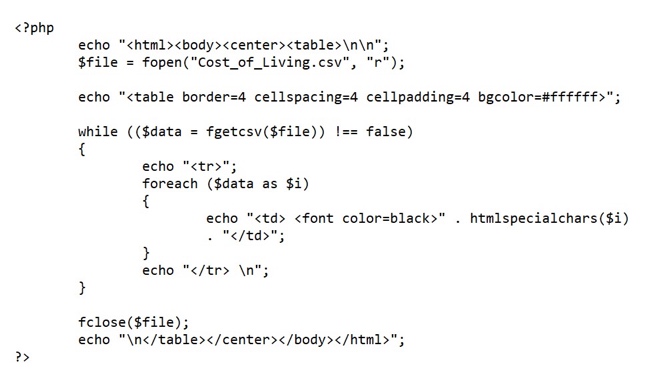
Mac PHP installation command:

brew install php

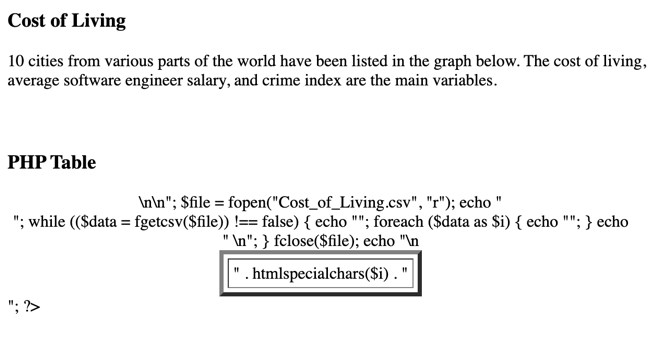
**IV. PHP Setup**

PHP language code does not run on HTML files by default. The webpage file “PHP\_Example.html” will not display the following table code on the browser.

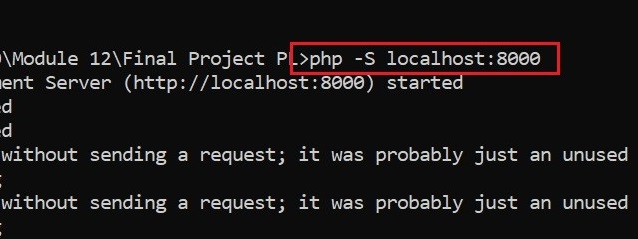
The PHP code: source - geeksforgeeks.org [6]



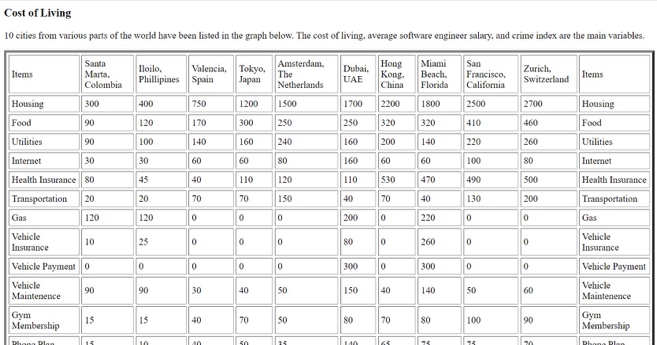
The HTML display:



As you can see, instead of a table, what you are seeing is non-sense error generated HTML gibberish. To mitigate this issue, one must first change the name of the “PHP\_Example.html” file to “PHP\_Example.php”. After that, the next step is to search for the “Final Project PL” folder in the Command Prompt. Once there, my next step was to type in the following command in terminal:



Once the localhost server has been setup, type in “localhost:800/PHP\_Example.php” link in the Chrome browser to view the PHP table code [4].

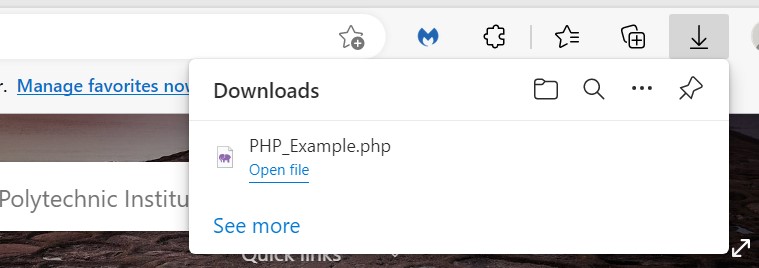


Be sure to run the localhost url in the Chrome browser. Any attempt to run the PHP file in the Microsoft Edge browser results in a file download instead of a proper webpage display.

Chrome PHP:



Edge PHP:



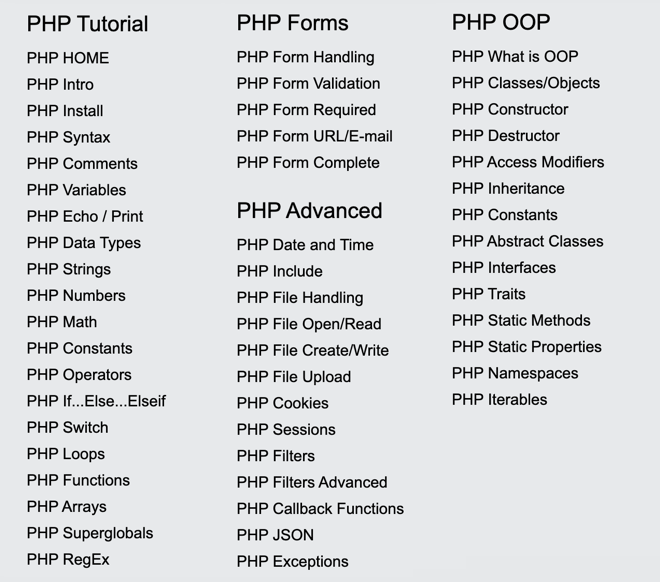
The previous iteration of this project had an “.htaccess” file. However, with the additional research and the revelation that an Apache server was not necessary. The “.htacess” was removed.

**V. PHP Features**

The versatility of PHP extends beyond just the generation of tables from .csv files. The range of possible uses the language serves as one of the major reasons why nearly 80 percent of websites today use PHP. The strong community support the language has had since 1994 has allowed for the creation of a vast library of tutorials available for new users to learn from [3].

The PHP language comes with a wide array of possible uses that range from the use of math and operations, to the inheritance of classes. The image below lists a number of the features of PHP.

PHP Features: source – w3schools.com [14]



One relatively unknown but crucial function of PHP is its ability to interface with SQL databases. PHP offers a notable number of functions and operations that allow for the modification of the information inside SQL tables.

PHP SQL Functions: source – w3schools.com[14]

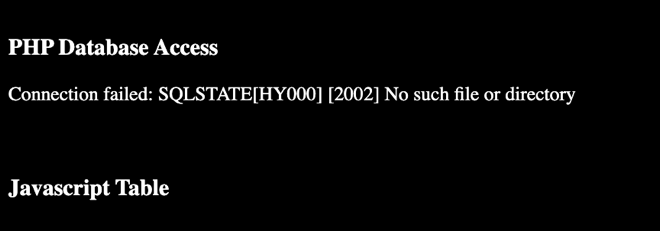


The code below, sourced from “w3schools.com”, utilizes a set of variables: server name, username, password, to connect to a specified SQL server. If one of the three variables does not correspond to a valid server, the PHP webpage prints an error message [7].

PHP Database Code: source – w3schools.com [7]

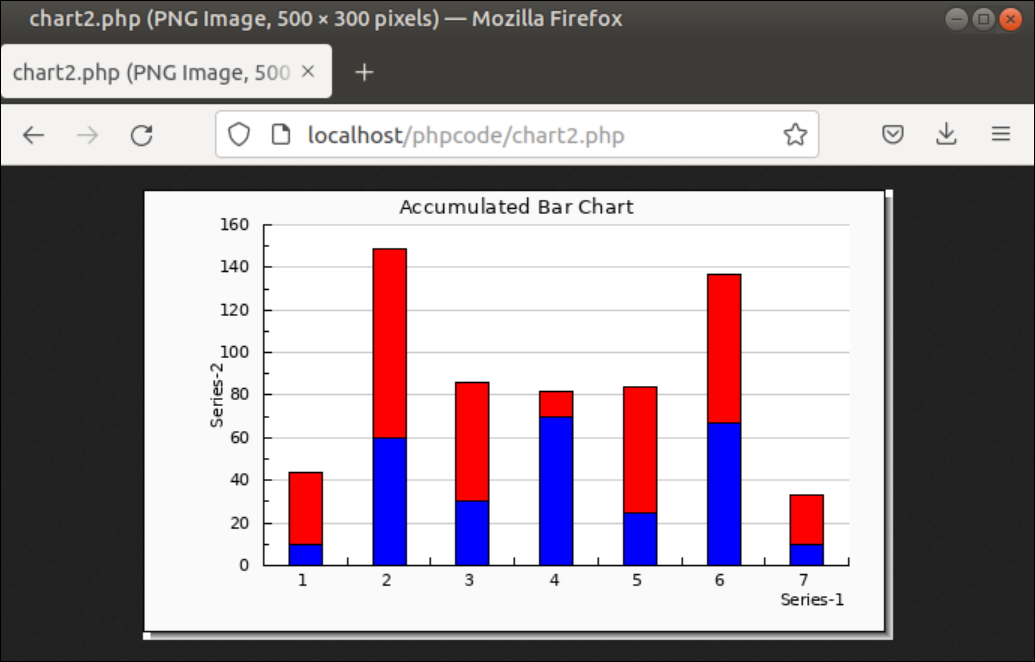


Error Message:



Another popular use for the PHP language is the generation of graphs. With the use of libraries such as JPGraph, one can create graphs from datasets imported from tables. The website “linuxhint.com” displays one example of a graph output of PHP code.

Source – linuxhint.com [6]



The implementation of libraries in PHP however is not as straight forward as it is in Python. In Python, I could download the tar.gz file of the library and install it with a “pip install name” command. In the case of PHP, the “php install jpgraph” does not work [8].

One potential work around I found was to unzip the tar.gz file and place the “jpgraph-4.4.1” library folder in my project directory. On the The “require\_once()” function was called just below the “Comparison Graph” section of the “PHP\_Example.php” file. The function imports the specific library files required for the creation of graphs.

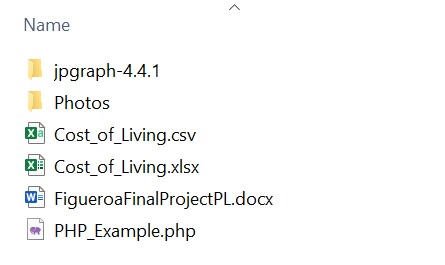
JPGraph Library import [6]:

require\_once (‘jpgraph-4.4.1/src/jpgraph.php’);

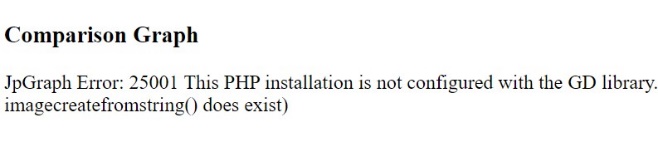
require\_once (‘jpgraph-4.4.1/src/jpgraph\_line.php’);

Despite the relocation of the JPGraph library to the project directory. This potential “work around” still did not achieve its intended purpose. The “require\_once()” function yielded an error despite the use of the correct directory address.

Project Directory:



JPGraph Error:



The nature of the error above suggests that the problem stems from a configuration issue. To make JPGraph work, the cache and the TTF directory of the PHP language must be adjusted to suit the installation. And to even be able to make such a change, the PHP needs to have the proper permissions to allow such changes [8].

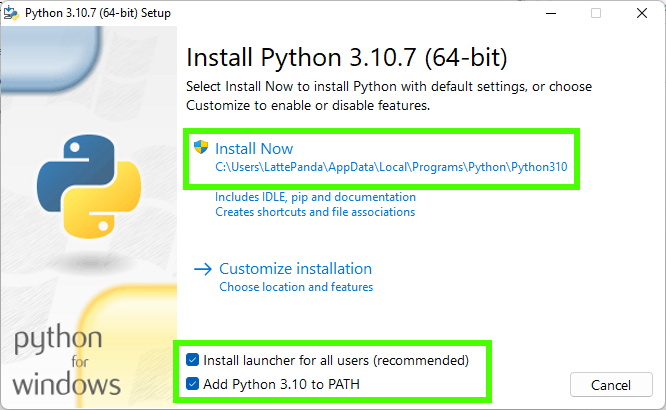
I am going to need to learn more about the intricacies of permissions and caches before I could explore the creation of PHP graphs further. This project unfortunately will not include functional PHP graph examples.

**VI. Python Comparison**

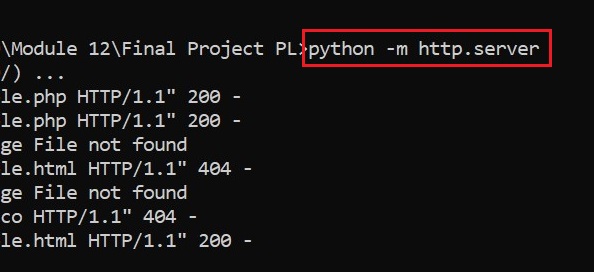
A number of notable similarities and differences exist between the PHP language and the Python language. Like PHP, Python has the capability to utilize classes. There are a number of other roles that Python can play that PHP also fulfils.

One similarity that Python shares with PHP is its ability to host a local client server. To run a server, one must install the Python language. Unlike PHP, Python comes with a downloadable installer which allows users with less exposure to command prompts easier access to the language.

Windows Python Installer: source – tomshardware.com [12]



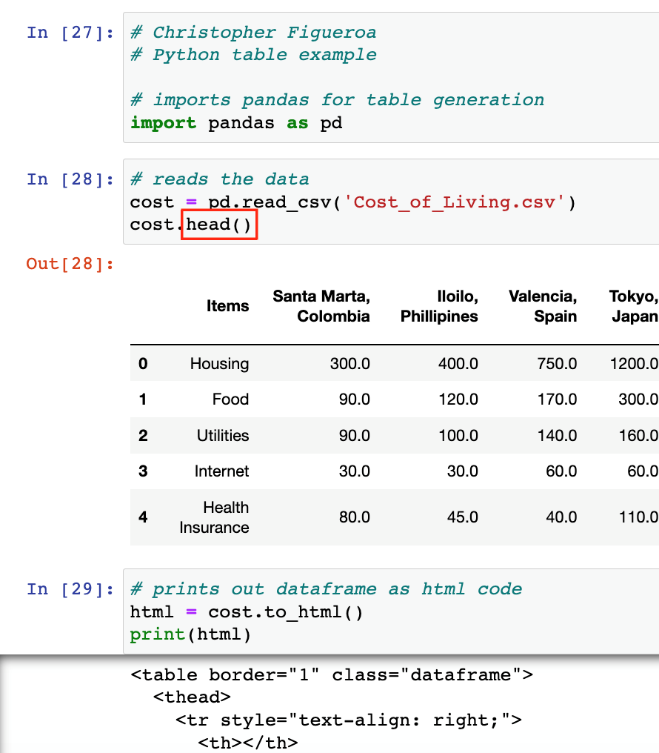
One the installer has completed its process; one must then verify that Python has been installed correctly by typing “python –version” into terminal. If the command prompt returns a valid version “Python 3.9.13”, then the next step would be to type in the command below to setup the server.



With the server setup, I can now display any code I make in Python onto a webpage. The one variant of Python that I used for this project is the Anaconda language, which comes with its own separate installer. The syntax is the same, the one notable difference is that Anaconda allows for the creation of Jupyter-notebook .ipynb files [1].

The code in the “PythonTable.ipynb” file imports the Pandas library which is essential for the generation of tables. The .ipynb file itself contains a number of code cells which can be run independently from one another. The “head()” function can be used to display the table from the “Cost\_of\_Living.csv” file if it is the last line in the cell.

Jupyter-notebook code: source – pythonprogramming.altervista.org [10]

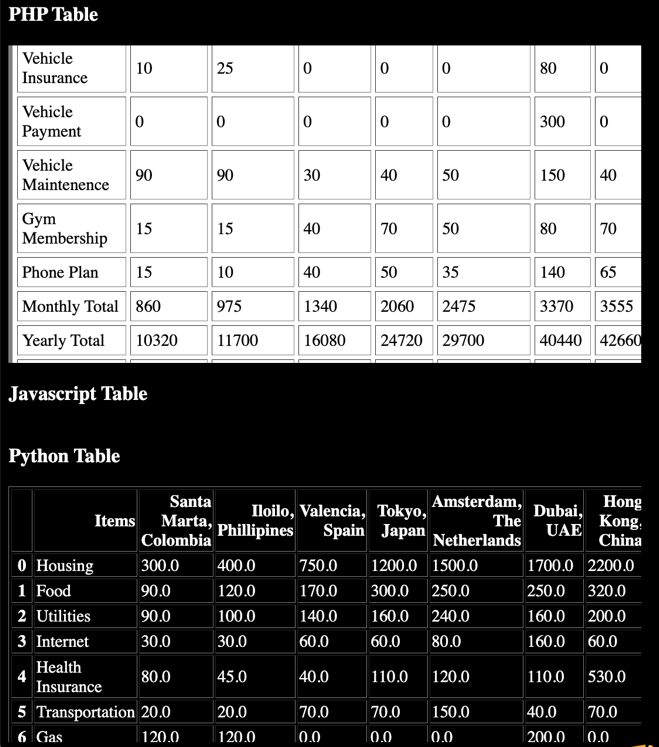


HTML Python Code:



Anaconda’s cell output feature also allows for the generation of HTML code from the “to\_html()” function. The output of that function serves as functional code that can be pasted into the “PHP\_Example.php” file [10]. A method to directly link the python to an HTML or PHP file may exist, however I will just have to paste the output of the Python code into the .php file due to time constraints.

Python HTML Table:



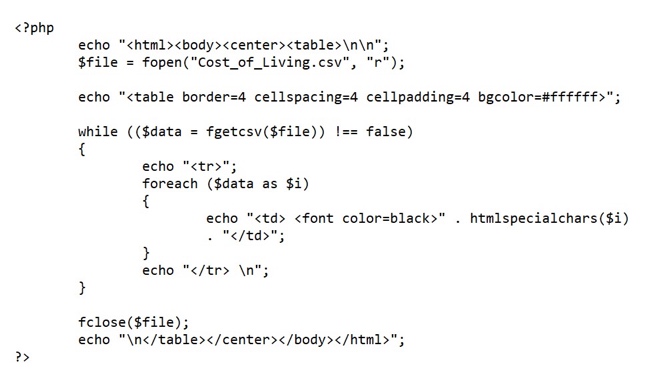
**VII. JavaScript Comparison**

When compared with PHP or Python, one can argue that the installation process for JavaScript is the simplest out of all the aforementioned languages. Many devices already come with support for JavaScript build in to their browsers. The wide range of support is among the reasons JavaScript is often favored over PHP by web developers. However, there are some other factors that make PHP more favorable in some instances [3].

Like PHP, and Python, JavaScript has the capability to perform operations and read and write databases. However, the feature that JavaScript lacks that is present in PHP and Python is the use of classes. Since JavaScript is not an object oriented language, it cannot make use of inheritance [14].

Another notable advantage that PHP has over JavaScript is that PHP is less prone to errors than JavaScript. The reason being, JavaScript code is more complex. The PHP code on the top image below is notable more simple than the more cumbersome JavaScript code.

The PHP code: source - geeksforgeeks.org [6]



JavaScript Code: source – code-boxx.com [9]

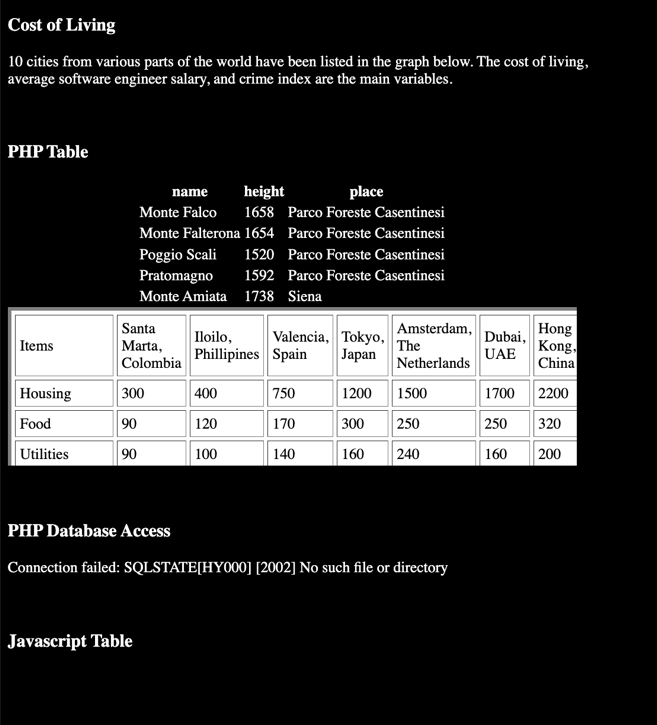


The JavaScript code in the image in the previous page does not even run. It displays a blank space instead of the table. The one example of code that does work does not read .csv files. It is instead a program comprised of lines that construct a simple table from scratch.

Working JavaScript Code: source – valentinog.com [13]



Glitchy JavaScript Table:



Although the code creates the table successfully, because of some difficult-to-find syntax error, the JavaScript table shows up in the same “<div>” space as the PHP table. The table was supposed to be below the “Javascript Table” title. The error in the code is correctable, but the purpose of this project is to compare the PHP language to Python and JavaScript, not to make the perfect PHP webpage.

**IIX. Challenges in Coding**

The original goal of this project was to create a PHP website with tables and graphs made from PHP code, Python, and JavaScript. Although I for the most part succeeded in my attempts to create tables using all three languages, I ran into a multitude of hurdles that prevented me from being able to successfully complete the graphs.

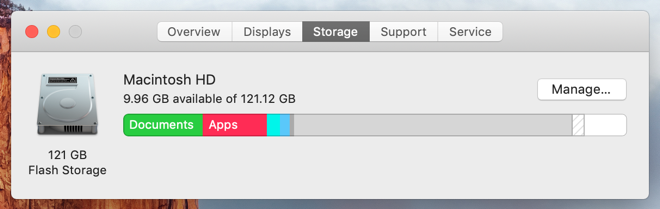
Earlier in this report, I mentioned the issues I encountered when I made my attempt to install the JPGraph library into PHP. I also described the trouble I encountered when I attempted to rectify the non-responsive JavaScript table generator code. But out of all the barriers I faced, the one that presented me with the most adversity was the computer owner’s greatest nightmare, “The Blue Screen of Death”.

Blue Screen of Death: source - h30434.ww3.hp.com[15]

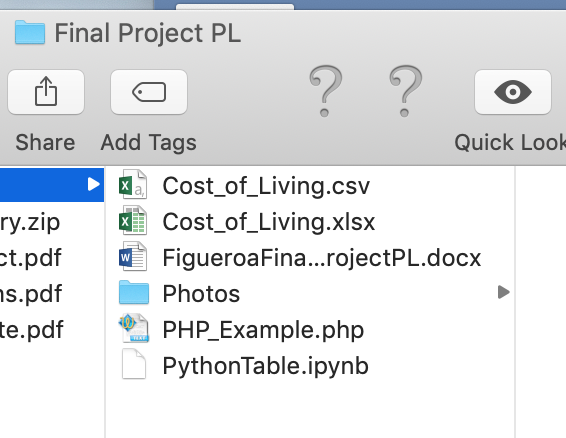


As I was about to begin the construction of my graphs, my computer inexplicably forced itself to restart after it prompted a message that indicated “a problem has been detected”. I ignored the error at first since I first believed an error caused by an excessive use of memory. I made sure not to have Chrome and Word open at the same time. But the problem still persisted.

No longer willing to lose work over crashes, I transferred every file and software I needed over from my faulty Windows computer to my more reliable MacBook pro. I spend hours reinstalling PHP, Python, and Anaconda onto my new device. I had to only install the essential programs since the space in my MacBook was very limited.



The Anaconda environment I reinstalled took over 20GB of storage space. The Python and PHP environments took roughly 1GB of storage each respectively. The installation of these programs and the limits on the number of new libraries I installed inhibited my ability to create the graphs I had planned.



The storage limit was another reason I opted to utilize the Jupyter-notebook environment of Anaconda instead of other traditional Python compilers such as PyCharm or Visual Studio Code.

**IX. Conclusion**

My research in the features, installation, and the uses of the PHP language has given me the knowledge necessary to understand this language thoroughly. Although I have not completed all of the objectives as I have hoped, I have gained a number of important insights on the similarities and differences between PHP, Python, and JavaScript.

With the information I have learned, I can conclude that PHP is as versatile and as simple to learn as the sources claim. All the necessary code can be contained by a few simple statements within the “<php” “?>” brackets themselves. And although the libraries are more difficult to install than those of Python or JavaScript, PHP’s usability of databases more than makes up for that shortcoming.

**References**

[1] B. Andriamanalimana, “Software”, *Module 1*, p. 66-70, 2022.

[2] I. Amabhijha, “How to Install PHP on Linux?”, *Geeksforgeeks*, p. 1-4, 2021.

[3] S. Mino, “8 Reasons Why PHP is Still so Important for Web Development”, *Jobsity*, p. 1-6, 2022.

[4] A. Ndugu, “How to run html on localhost”, *Devpractical.com*, p. 3-6, 2022.

[5] S. Kumar, “How to Display Data from CSV file using PHP>”, *Geeksforgeeks*, p. 1-3, 2020.

[6] F. Yesmin, “How to Create a Chart Using PHP”, *linuxhint.com*, p. 2-10, 2017.

[7] “PHP Connect to MySQL”, *w3schools*, p. 1, 2022.

[8] “3 Installation”, *digialliance*, p. 1-2, 2021.

[9] W. Toh, “Display CSV As Table In Javascript (Simple Examples)”, *code-boxx*, p. 5-9, 2022.

[10] P. Giovanni, “Python to make tables code”, *pythonprogramming*, p. 1-4, 2019.

[11] A. Lo Duca, “How to insert a Python graph into an HTML Web Site”, *towardsdatascience*, p. 2-7, 2021.

[12] L. Pounder, “How To Install Python on Windows 10 and 11”, *tomshardware*, p. 2-6, 2022.

[13] V. Gagliardi, “Back To The Basics: How To Generate a Table With JavaScript”, *valentinog*, p. 2-8, 2020.

[14] “PHP Introduction”, *w3schools*, p. 1-6, 2022.

[15] Q. Tedi, “Blue screen of death (BSOD) 116”, *www3.hp.com*, p. 1, 2019.