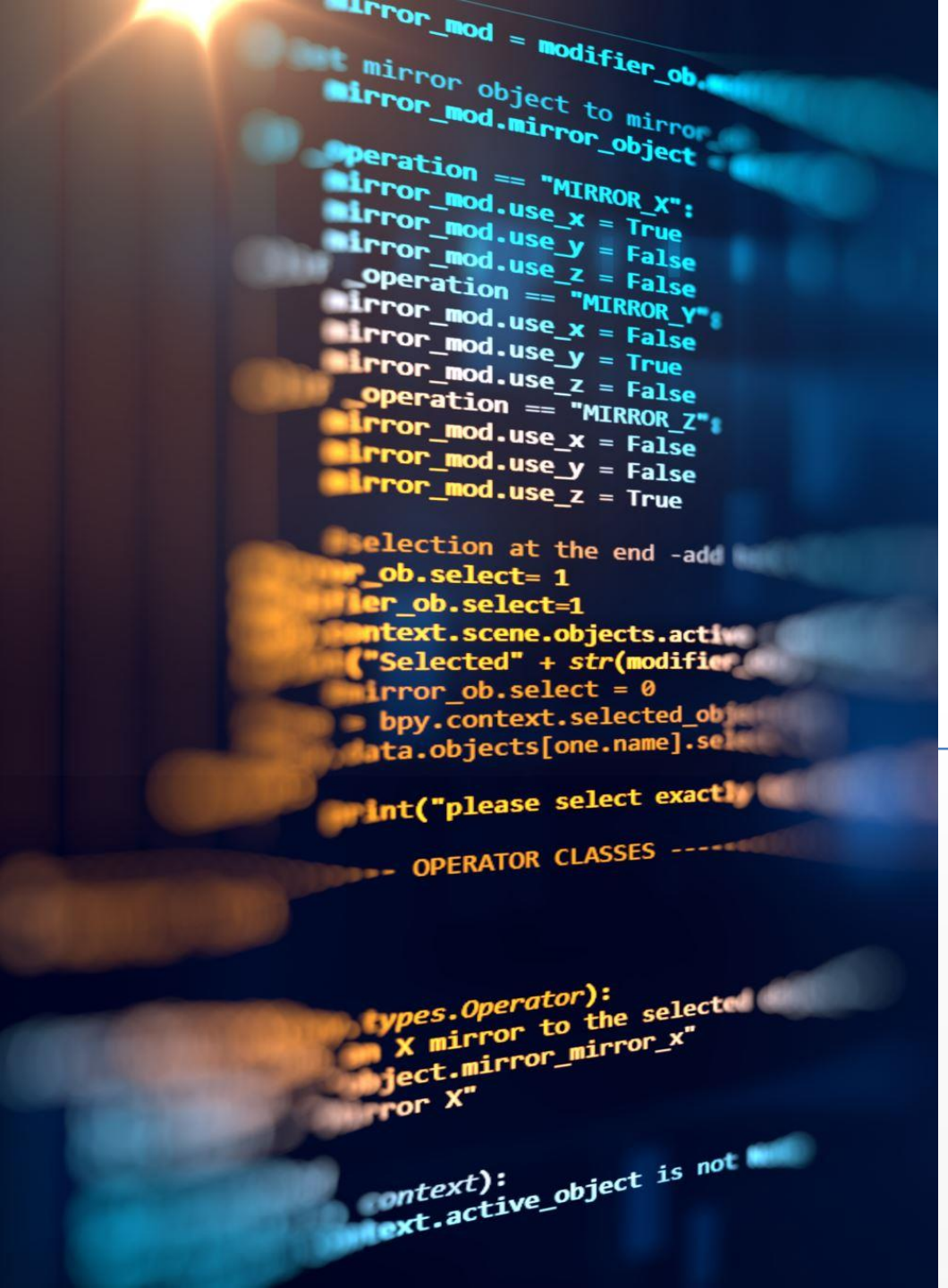


UMS
UNIVERSITI MALAYSIA SABAH

FACULTY OF COMPUTING & INFORMATICS
FAKULTI KOMPUTERAN DAN INFORMATIK

Introduction to PHP x PYTHON INTEGRATION

Module By
Dr. Mohd Shamrie Sainin
Faculty of Computing and Informatics, UMS

PHP x Python Integration

- Objectives of this workshop:
  - To introduce the web-based application technology
  - To explain the requirements of PHP x Python application integration
  - To provide samples and hands-on sample applications of PHP x Python
- This sharing session is to explain the hands-on integration of these two languages
  - PHP = Server-side processing (frontend)
  - Python = Backend application support

Technology

PHP + Python

- Implementation in WAMP, XAMPP based on Common Gateway Interface (CGI)
- Implementation
  - **Without Framework (we do this)**
  - With Framework: Laravel, Yii2, CodeIgniter (MVC framework)

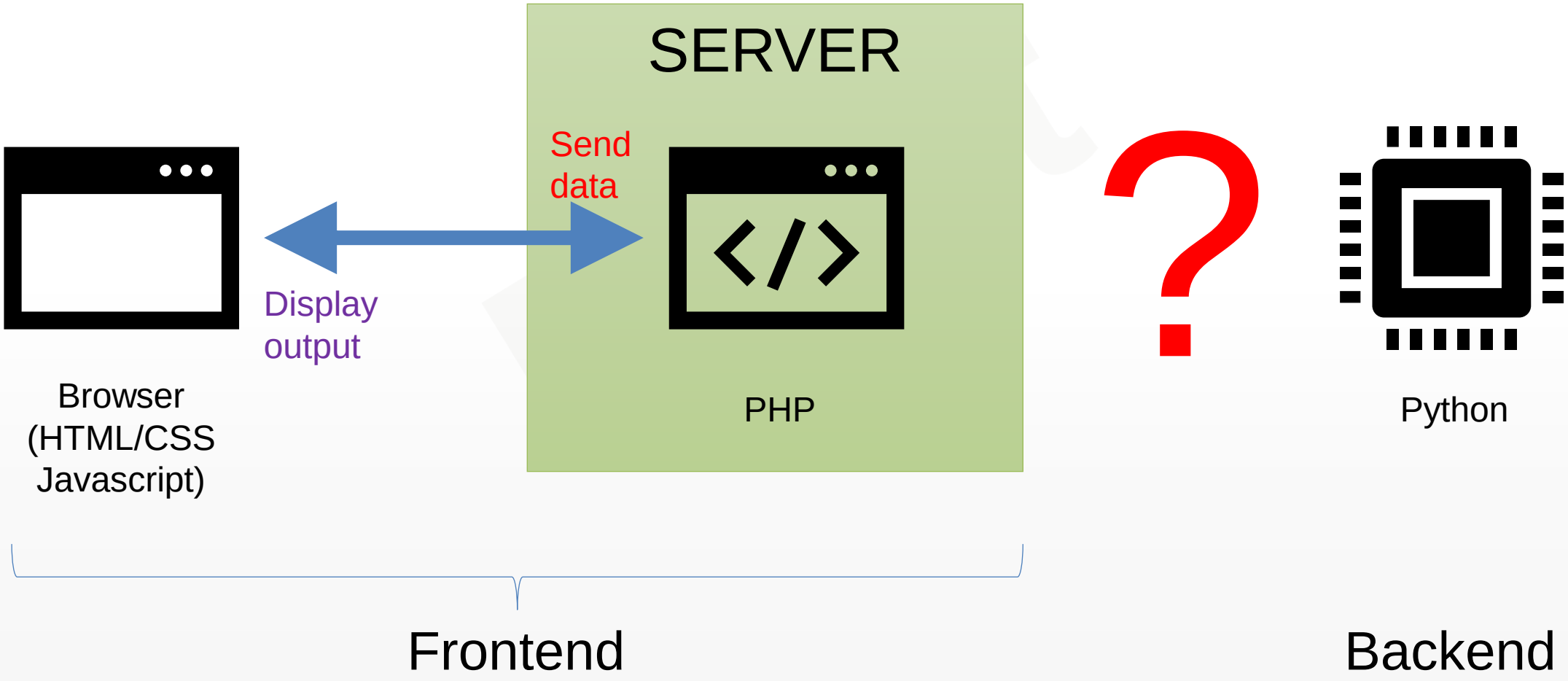
DJANGO Framework

- Django is a high-level Python framework which provides support for web apps.
- MVT framework
- Need to be installed

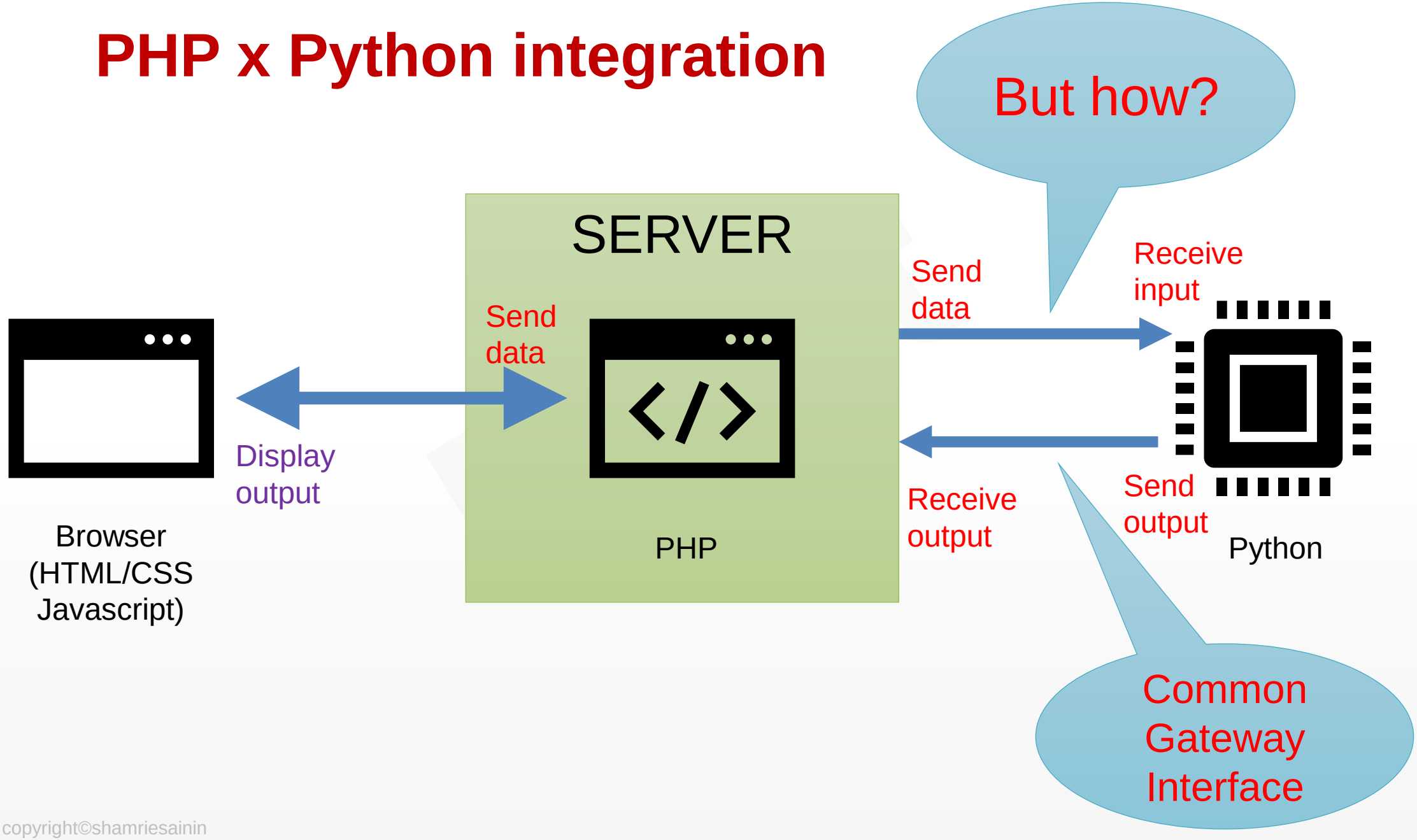
Flask

- Lightweight Web Server Gateway (WSGI) Interface web application framework with Python
- Flask is said to be more Pythonic than the Django

LOST? I don’t know how to integrate…



PHP x Python integration



REQUIREMENTS

Before we begin

- This workshop is NOT to learn in detail about HTML, CSS, JavaScript, PHP
- This workshop NOT to learn in detail about Python programming
- This workshop is to provide how to integrate PHP and Python using basic examples
- The examples in this workshop are running on WAMP 3.0.4, PHP Version 5.6.19 and Python 3.8
- Not implemented in PHP framework (such as Laravel, CodeIgniter, Yii, etc.)

Pre-requisite

- Web programming (HTML, PHP)
  - You are already familiar with web development
- Python
  - You have already had a basic Python programming

Requirements

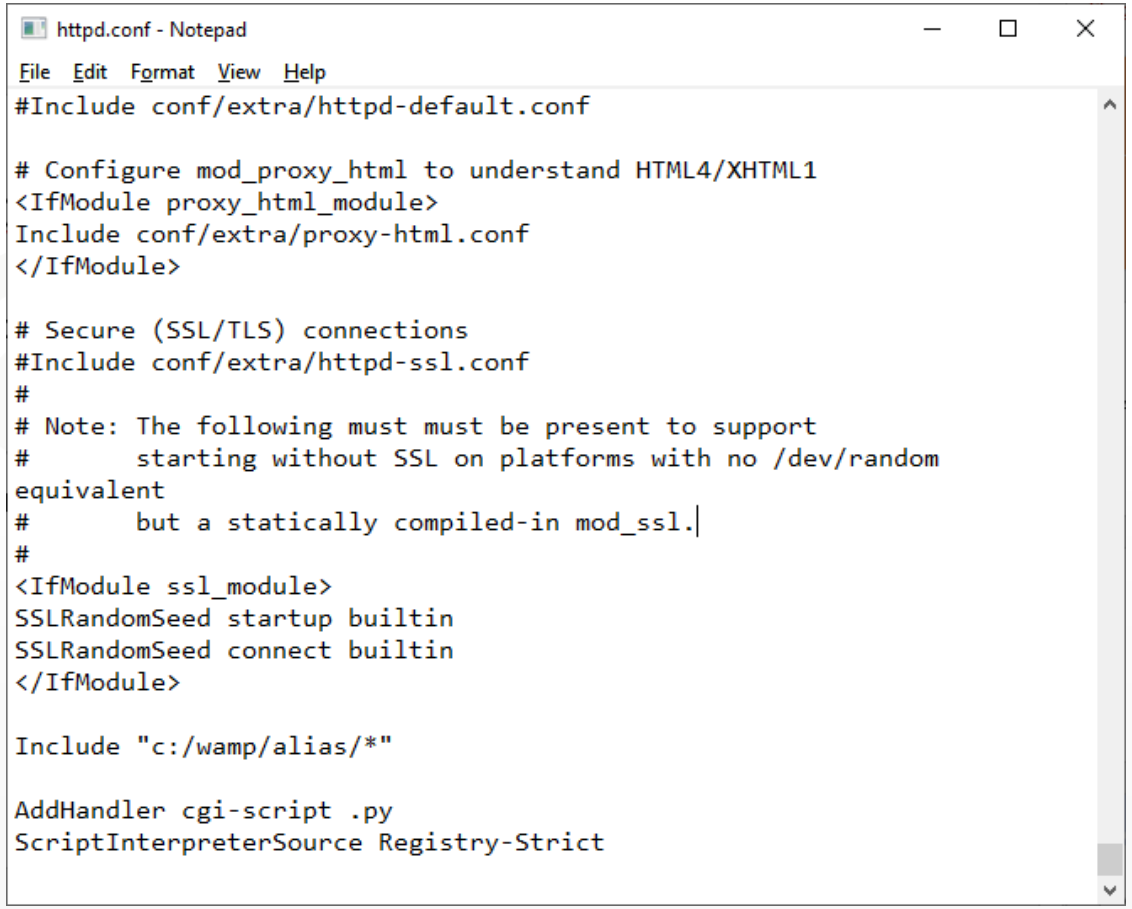
- WAMP/XAMPP local server
- Python 3.8
- Python IDE (preferably lightweight such as Thonny)
- Do not use other environments such as Anaconda. (but if you do, make sure you know the environment very well)
- Python Packages
  - sys, pandas, sklearn, numpy, pickle, json, vaderSentiment, etc.

Requirements: Server Handler

- Handler in httpd.conf (WAMP/XAMPP), paste in the end of the file
- Save and restart WAMP/XAMPP server

AddHandler cgi-script .py

ScriptInterpreterSource Registry-Strict



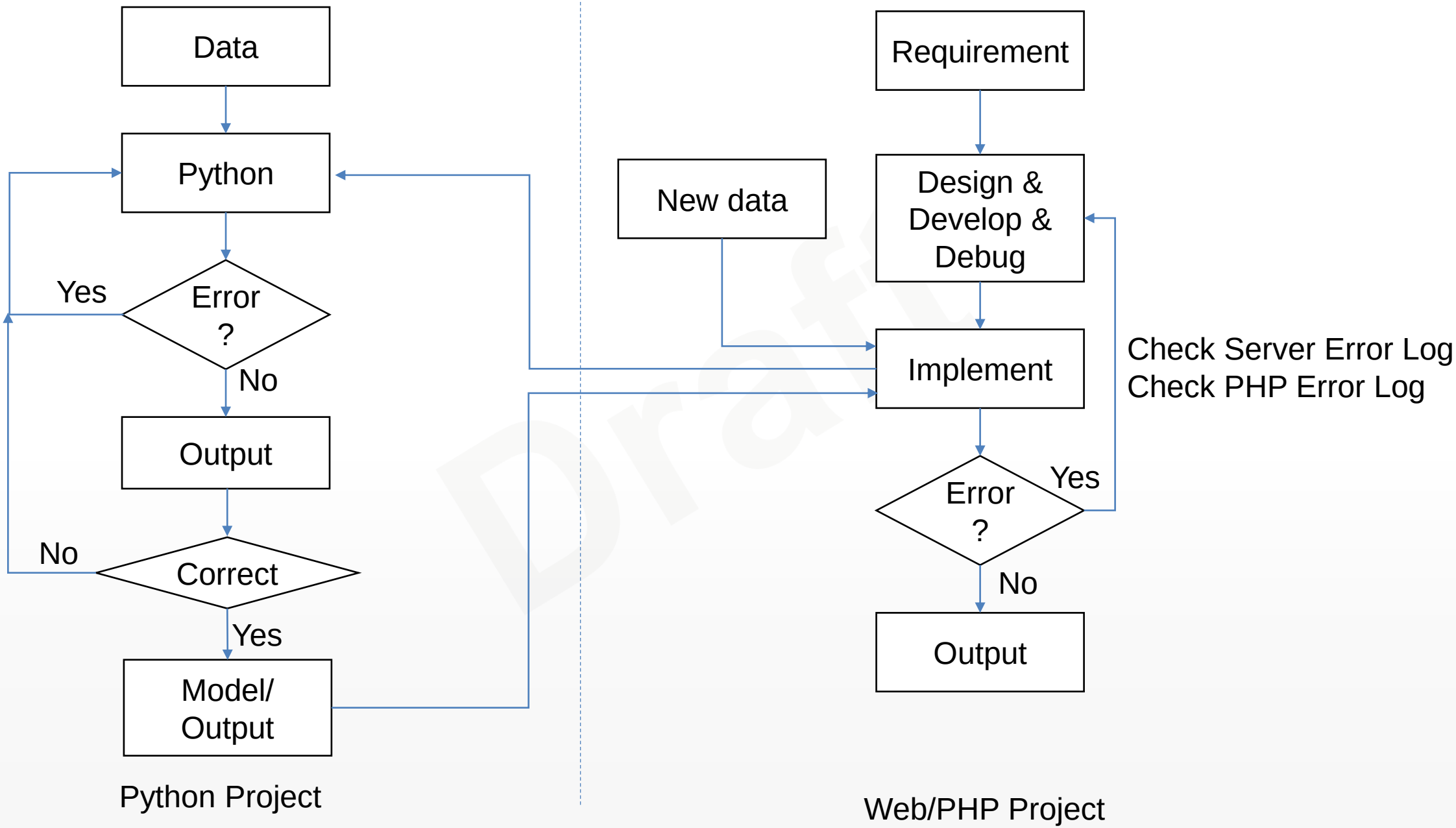
Requirements: Python Installation

- Recommended to Install PHP in **C:\users\...** folder.
- Install Python 3.8 if you plan to use tensorflow. Python 3.9 is not yet supported.

Python and packages path

- Check your Python executable complete path. May be different according to your Python installation and version
- Example
  - C:\Users**name**\AppData\Local\Programs\Python\Python38\python.exe
- Packages
  - Make sure you know where the packages installed
  - Check either in Local or Roaming
  - **pip install <packagename> --user** command will install in Roaming

Flow of work in PHP x Python



EXAMPLE 1

Testing PHP and Python Connection

Example 1: Test your connection

- Change to your (name) python.exe path!
- In Python file, make sure PATH to python.exe is in the first line.

test.php

```
<?php
$x = 1;
$output =
exec("C:\Users\name\AppData\Local\
Programs\Python\Python38\python.ex
e test.py $x");
var_dump($output);
echo $output;
?>
```

test.py

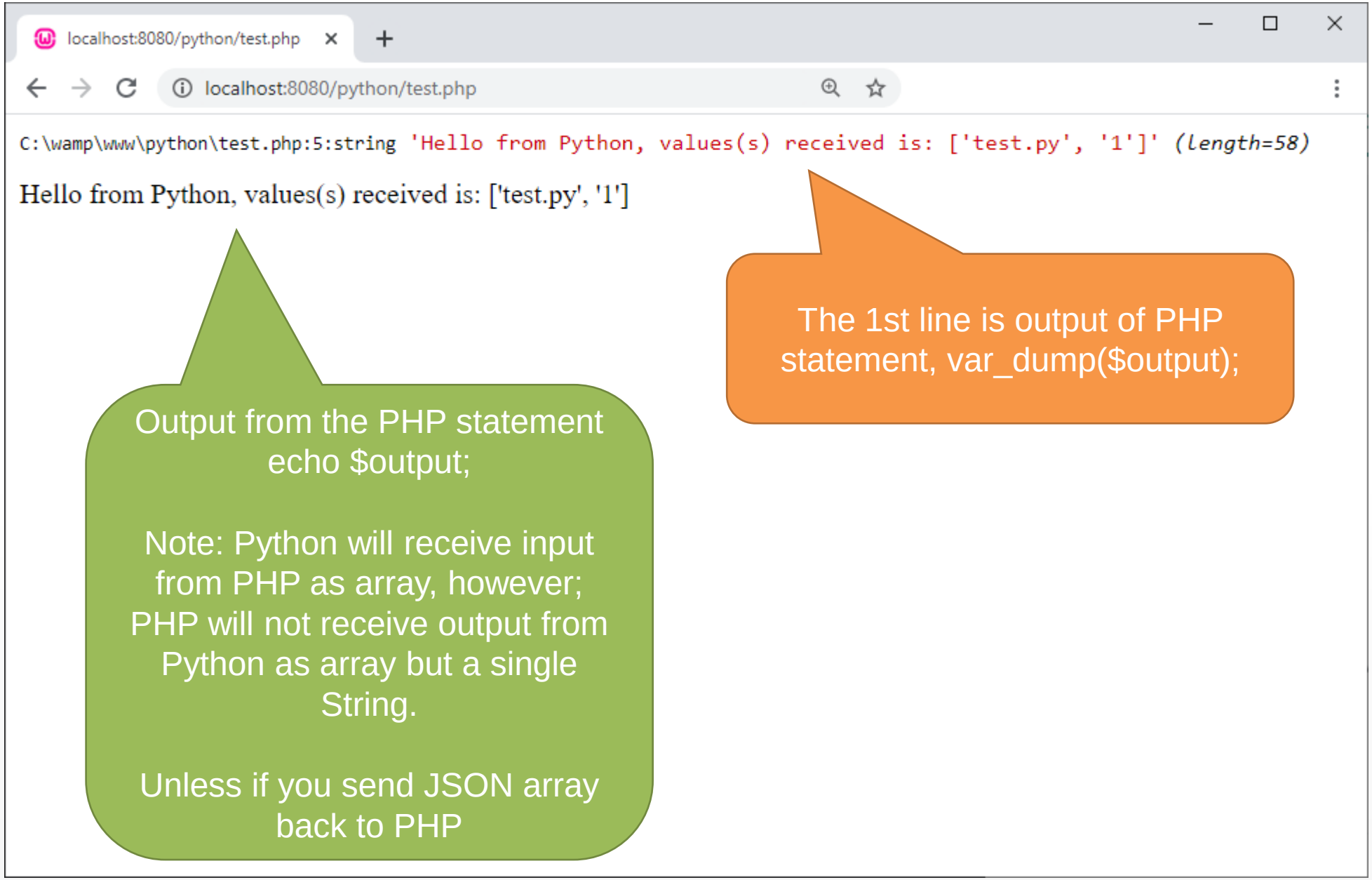
```
#C:\Users\name\AppData\Local\Progr
ams\Python\Python38\python.exe

import sys

var = sys.argv


print("Hello from Python, values(s)
received is: " + str(var))
```

Important part in PHP file

- $output = exec("C:\Users**name**\AppData\Local\Programs\Python\Python38\python.exe test.py $x");
  - C:\Users**name**\AppData\Local\Programs\Python\Python38\python.exe is your python.exe path
  - test.py is the python file to be executed
  - $x is the data/argument passed to Python

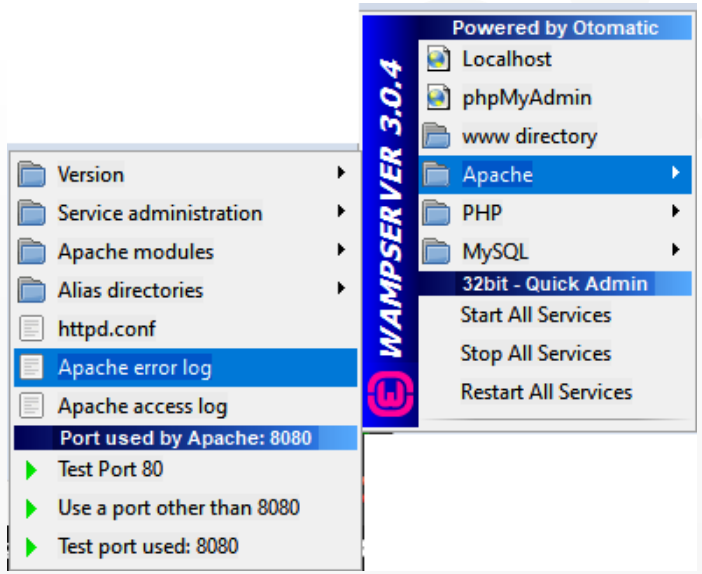
Important part in Python file

- #C:\Users**name**\AppData\Local\Programs\Python\Python38\python.exe
  - Must be the first line in every Python file for (PHPxPython)
  - Change to your path
- import sys
  - Required to enable argument reception from PHP
- var = sys.argv
  - Assign argument(s) received from PHP in array type
  - sys.argv[0] is the python file to be executed
  - sys.argv[1] is the 1st value (argument) passed from PHP
  - sys.argv[2] is the 2nd value (argument) passed from PHP if available.
  - If more arguments passed, then get based on the index number
- print("Hello from Python, values(s) received is: " + str(var))
  - Concatenating variable from sys.argv need to cast to str



None is displayed in the browser? Don’t panic

- Check your code (PHP/Python)
- Check Server Error Log



apache_error.log - Notepad

File Edit Format View Help

```
The system cannot find the path specified.
'PYTHON' is not recognized as an internal or external command,
operable program or batch file.
'PYTHON' is not recognized as an internal or external command,
operable program or batch file.
[Sat Nov 28 17:09:31.277177 2020] [:error] [pid 12804:tid 1252] [client
::1:56006] script 'C:/wamp/www/python/form_sentiment.php' not found or
unable to stat
[Sat Nov 28 18:43:34.334503 2020] [:error] [pid 12804:tid 1244] [client
::1:61998] script 'C:/wamp/www/python/form_textarea.php' not found or
unable to stat
Traceback (most recent call last):
  File "form_process.py", line 4, in <module>
    print("Hello " + sys.argv[1] + " " + sys.argv[2])
IndexError: list index out of range
Traceback (most recent call last):
  File "form_process.py", line 4, in <module>
    print("Hello " + sys.argv[1] + " " + sys.argv[2])
IndexError: list index out of range
Can't find a default Python.
Traceback (most recent call last):
  File "form_sentiment.py", line 13, in <module>
    print(str(analyser.polarity_scores(x)))
NameError: name 'analyser' is not defined
```

Caution

- Calling and submitting values from PHP to Python is very sensitive
- Example:
  - $output = exec(…) - Python path and (.py) file to be executed and variables must be in **one line**
  - Long text (with white space) input must be escaped with escapeshellarg($input)
- Output from Python must be single print command only to able to be submitted back to PHP
  - If you have many output to be submitted to PHP, use a JSON array variable to save all output and print the JSON variable.

Example: Submitting array (JSON) input to Python

test_array.php

```
<?php
$x = 1;
$y = 2;

$input = array($x,$y);
$input = json_encode($input);

$output =
exec("C:\Users\name\AppData\Local\Programs\Python\Python38
\python.exe test_array.py ". escapeshellarg($input));
var_dump($output);
echo "<p>Output from Python: <br>" . $output . "<p>";
echo "Dump the output from Python as JSON<br>";

var_dump(json_decode($output));
$data = json_decode($output);
echo "Accessing from JSON array:<br>";
foreach($data as $key=>$value) {
                echo "[".$key . "] => " . $value . "<br>";
}
?>
```

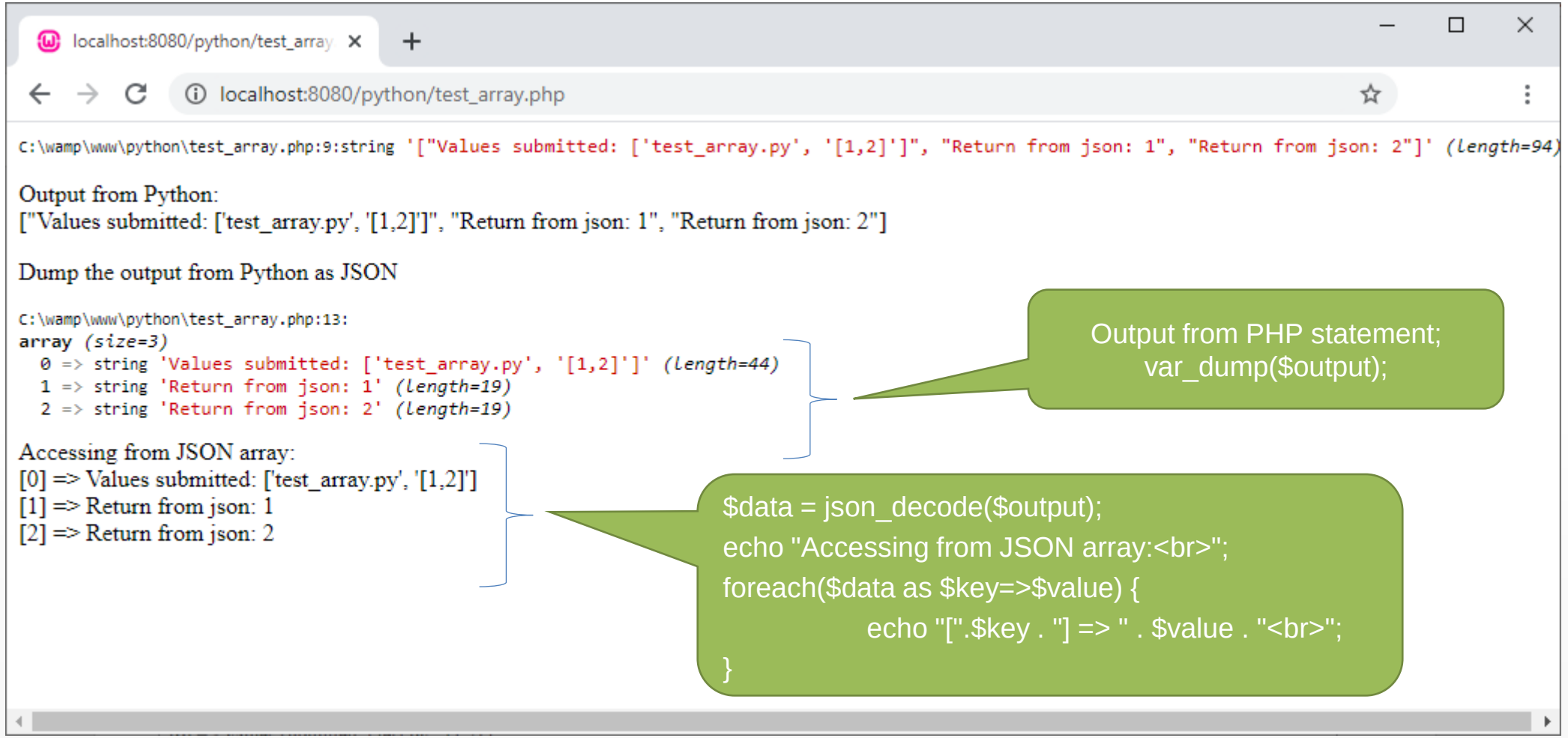
test_array.py

```
#C:\Users\name\AppData\Local\Programs\Python\Pytho
n38\python.exe

import sys
import json

#input = [1,2]
#input = json.dumps(input)
output = []
output.append("Values submitted: " + str(sys.argv))
#output.append("Values submitted: " + str(input))
j = json.loads(sys.argv[1])
#j = json.loads(input)

for k in j:
  output.append("Return from json: " + str(k))

print(json.dumps(output))
```

Test this first in Python if this file is providing output, e.g., uncomment
#input = [1,2]
#input = json.dumps(input)

Then, comment
j = json.loads(sys.argv[1]),
and uncomment
#j = json.loads(input)

Output: Submitting array (JSON) input to Python

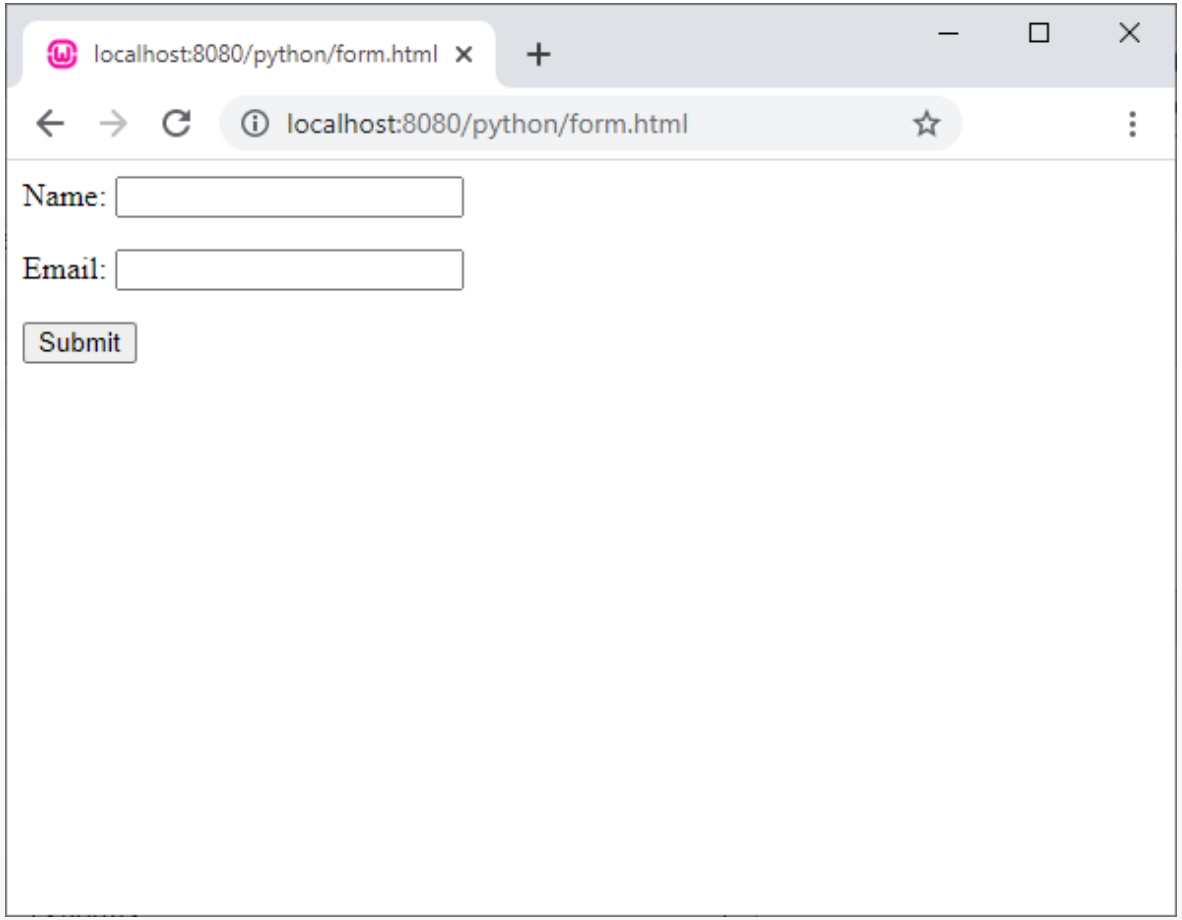


EXAMPLE 2

Working with form input

Example 2: form.html

```
<html>

<body>

<form method="post" action="form_action.php">

<p>Name: <input type="text"
name="name"></p>

<p>Email: <input type="email"
name="email"></p>

<input type="submit"  value="Submit">

</form>

</body>

</html>
```

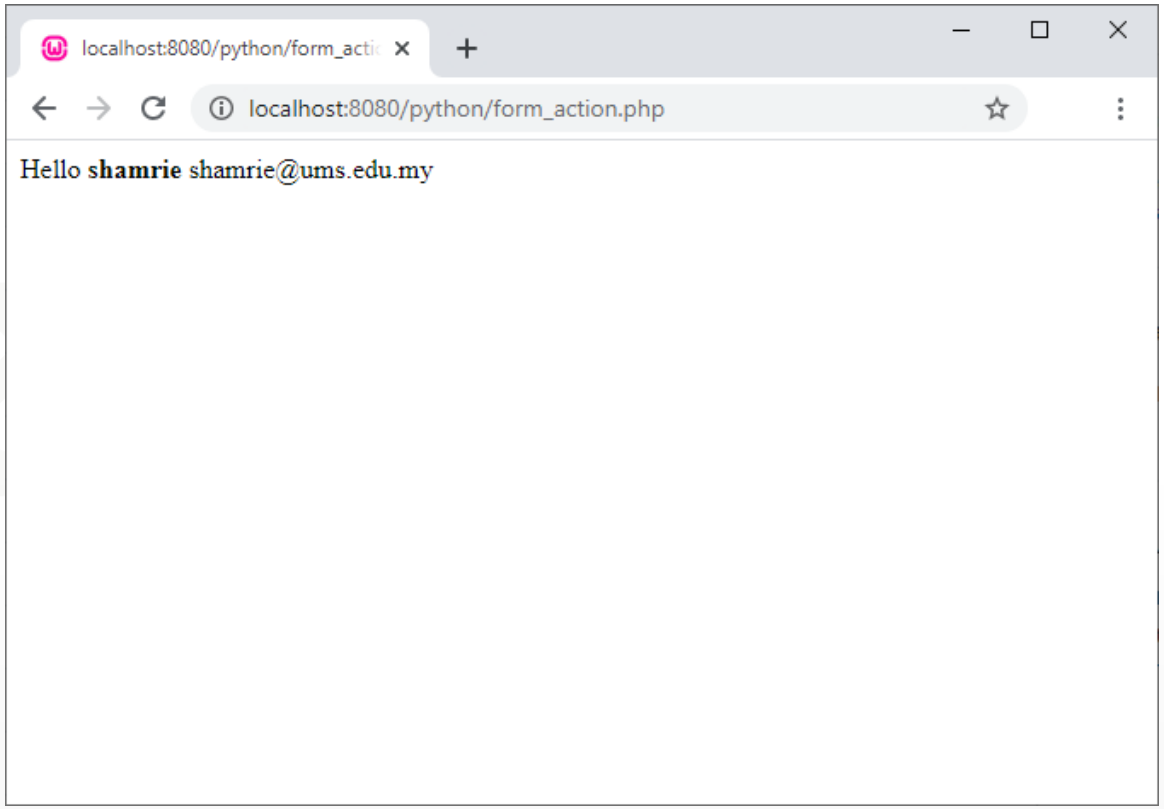


Example 2: form_action.php

```
<?php

$name = escapeshellarg($_POST["name"]);

$email = escapeshellarg($_POST["email"]);



$output =
exec("C:\Users\name\AppData\Local\Programs\Python\
Python38\python.exe form.py $name $email");

echo $output;

?>
```

Example 2: form.py

```
#C:\Users\name\AppData\Local\Programs\Python\Python38\python.exe


import sys

print("Hello " + sys.argv[1] + " " + sys.argv[2])
```



THANK YOU