



# **LAMINA STUDIOS, LLC DATA ANALYTICS INTERNSHIP**

**H2O Wave: Hello World Tutorial Activity  
Documentation**

***Jewel Anne A. Reyes***  
*BS Computer Science*  
*Polytechnic University of the Philippines*

**September 1, 2023**



## *Table of Contents*

Table of Contents.....	2
Task Overview.....	3
Codes.....	4
Screenshots of Work.....	5

## *Task Overview*

H2O Wave is a software stack for building beautiful, low-latency, realtime, browser-based applications and dashboards entirely in Python without using HTML, Javascript, or CSS.

It excels at capturing data, visualizations, and graphics from multiple sources and broadcasting them live over the web.

H2O Wave gives your Python programs the ability to push content to connected clients as it happens, in realtime. In other words, it lets your program display up-to-date information without asking your users to hit their browser's reload button. You can use H2O Wave for:

- Dashboards and visualizations for live monitoring.
- Live information displays: news, tickers, health, or performance data.
- Apps that require instant notifications, updates, events, or alerts.
- Apps that involve messaging: chat, bots, etc.
- Collaborative apps: whiteboards, sharing, etc.

You can also use H2O Wave when you find yourself reaching for a web application framework - it can handle regular (non-realtime) apps just fine.

In this task, we are required to create a program using H2O Wave and Python to run a page with corresponding box attributes, title, and its content.



## *Codes*

The following are the steps with codes that was used for this task.

**Step 1:** Open a terminal and start the Wave Server

```
cd wave  
waved.exe
```

**Step 2:** Set up a virtual environment

```
python -m venv venv  
.\venv\Scripts\activate
```

**Step 3:** Write the program

```
from h2o_wave import site, ui  
  
# Grab a reference to the page at route '/hello'  
page = site['/hello']  
  
# Add a markdown card to the page.  
page['quote'] = ui.markdown_card(  
    box='1 1 2 2',  
    title='Hello World',  
    content='\"The Internet? Is that thing still around?\" - *Homer Simpson*',  
)  
  
# Finally, save the page.  
page.save()
```

**Step 4:** Run the program

```
python hello_world.py
```

## Screenshots of Work

Figure 1. Start the wave server using waved.exe command

```
C:\Users\reyes\wave>waved.exe
2023/09/01 12:04:46 #
2023/09/01 12:04:47 #
2023/09/01 12:04:47 # H2O Wave
2023/09/01 12:04:47 # 0.26.2 20230803101450
2023/09/01 12:04:47 # © 2021 H2O.ai, Inc.
2023/09/01 12:04:47 #
2023/09/01 12:04:47 # Running at http://localhost:10101/
2023/09/01 12:04:47 #
2023/09/01 12:04:47 # {"address":"10101","base-url":"/","t":"listen","web-dir":"C:\\Users\\reyes\\wave\\www"}
2023/09/01 12:04:47 # {"error":"listen tcp :10101: bind: Only one usage of each socket address (protocol/network address/port) is normally permitted"}
2023/09/01 12:04:47 # {"t":"listen_no_tls"}
C:\Users\reyes\wave>2023/09/01 12:05:00 # {"addr":["::1":64126],"route":"/","t":"ui_add"}
```

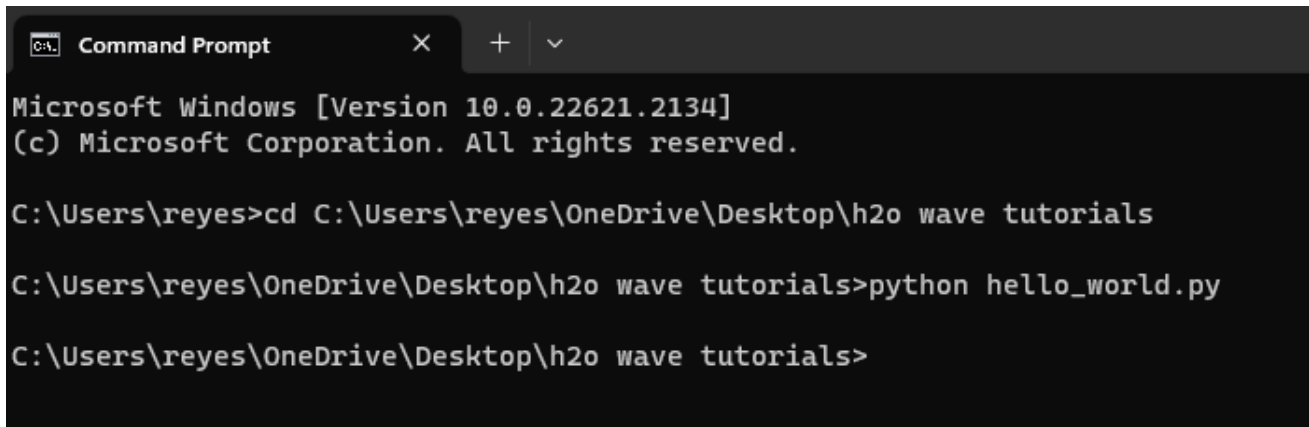
Figure 2. Setting the virtual environment

```
C:\Users\reyes\wave>python -m venv venv
C:\Users\reyes\wave>.\venv\Scripts\activate
(venv) C:\Users\reyes\wave>
```

Figure 3. Python code in Visual Studio Code

```
hello_world.py X
h2o wave tutorials > hello_world.py > ...
1 from h2o_wave import site, ui
2
3 # Grab a reference to the page at route '/hello'
4 page = site['/hello']
5
6 # Add a markdown card to the page.
7 page['quote'] = ui.markdown_card(
8     box='1 1 2 2',
9     title='Hello World',
10    content='\"The Internet? Is that thing still around?\" - *Homer Simpson*',
11 )
12
13 # Finally, save the page.
14 page.save()
```

Figure 4. Running the program in a terminal



```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\reyes>cd C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials
C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials>python hello_world.py
C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials>
```

Figure 5. Going to localhost:10101/hello and checking the output

