

Environment Setting

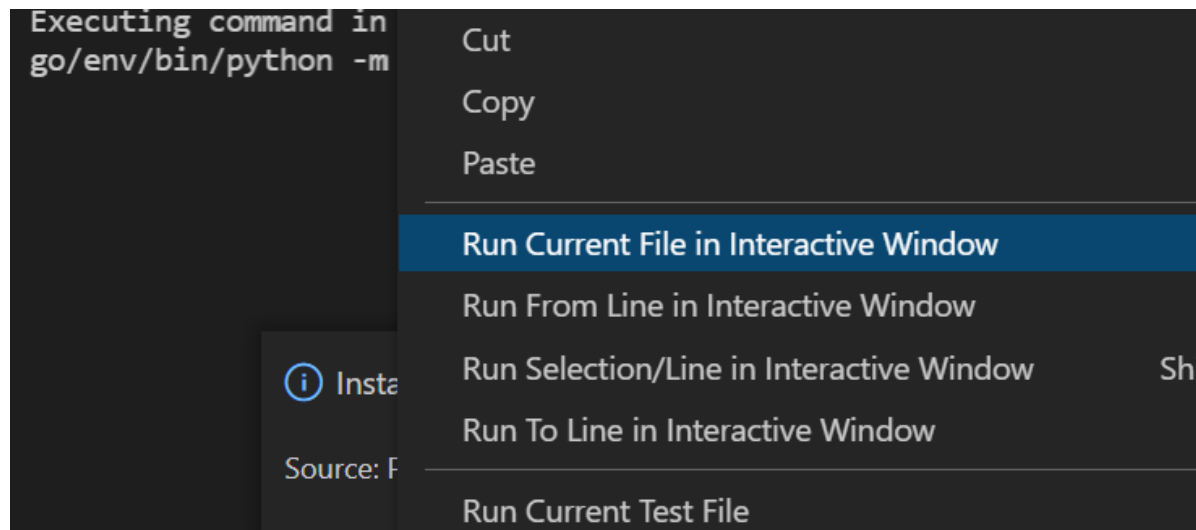
Create Folders for code

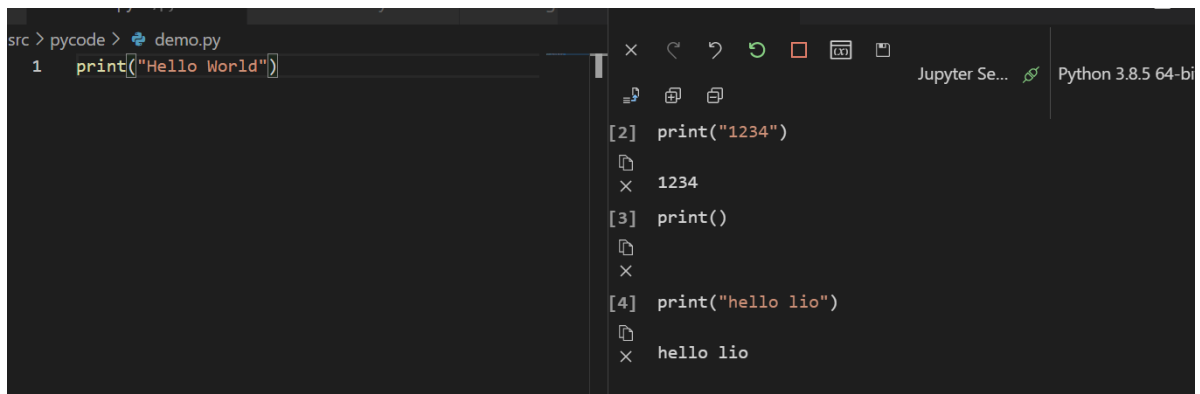
```
(env) melodie@ubuntu:~/Desktop$ cd Django
(env) melodie@ubuntu:~/Desktop/Django$ mkdir src
(env) melodie@ubuntu:~/Desktop/Django$ cd src
(env) melodie@ubuntu:~/Desktop/Django/src$ mkdir pycode
(env) melodie@ubuntu:~/Desktop/Django/src$ mkdir html
(env) melodie@ubuntu:~/Desktop/Django/src$
```

activate env (actually already activated)

```
(env) melodie@ubuntu:~/Desktop/Django$ source env/bin/activate
(env) melodie@ubuntu:~/Desktop/Django$ pip3 list
Package          Version
-----
asgiref           3.3.4
Django            3.2.4
pip               20.0.2
pkg-resources     0.0.0
pytz              2021.1
setuptools        44.0.0
sqlparse          0.4.1
```

Run current code in interactive window (and follow up install)





The image shows a code editor with a dark theme. On the left, a file named 'demo.py' is open, containing a single line of code: `print("Hello World")`. On the right, the output console shows the results of four separate code executions: `print("1234")` outputs '1234', `print()` outputs an empty line, and `print("hello lio")` outputs 'hello lio'.

^ easier for education, separate from the script on the left

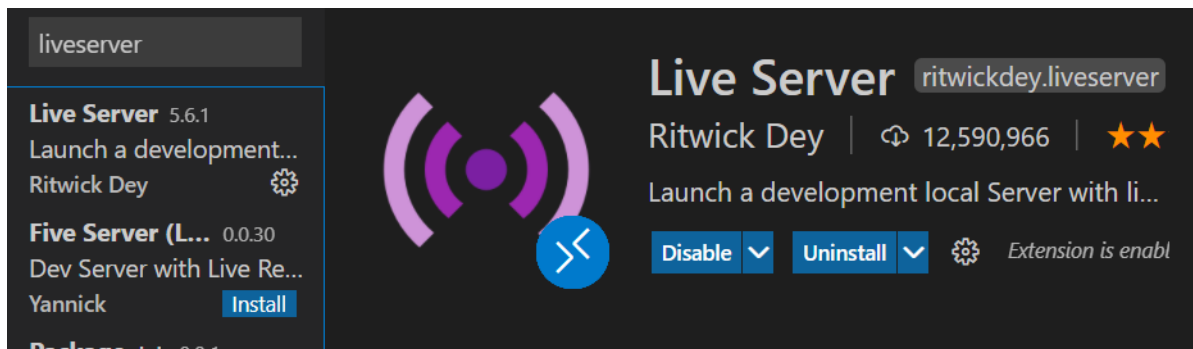
HTML knowledge

install extension: HTML

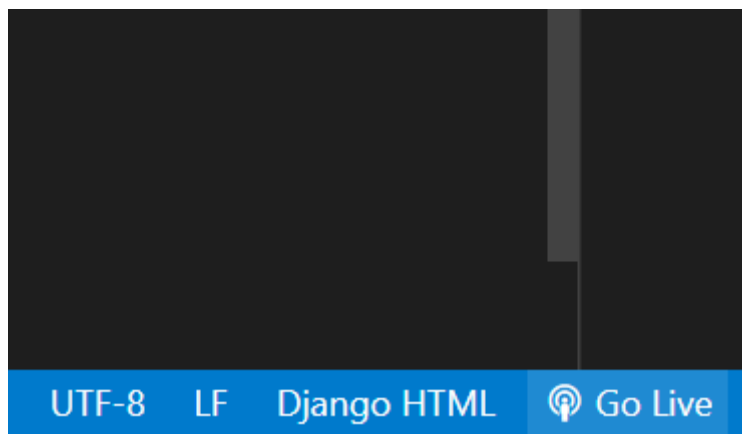
p: paragraph

table: `tr` (row). `td` (column)

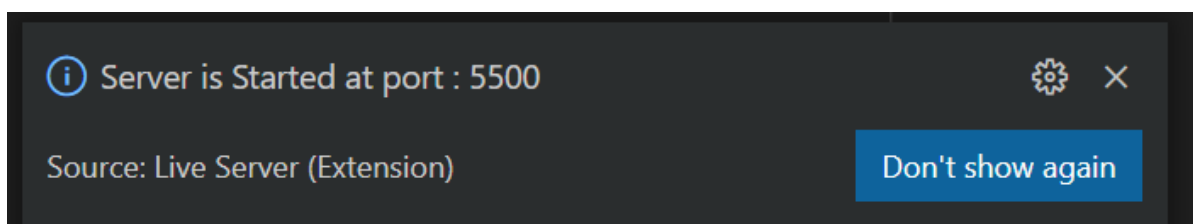
Install Live Server



click Go Live



and receive this:



Firewall Allow

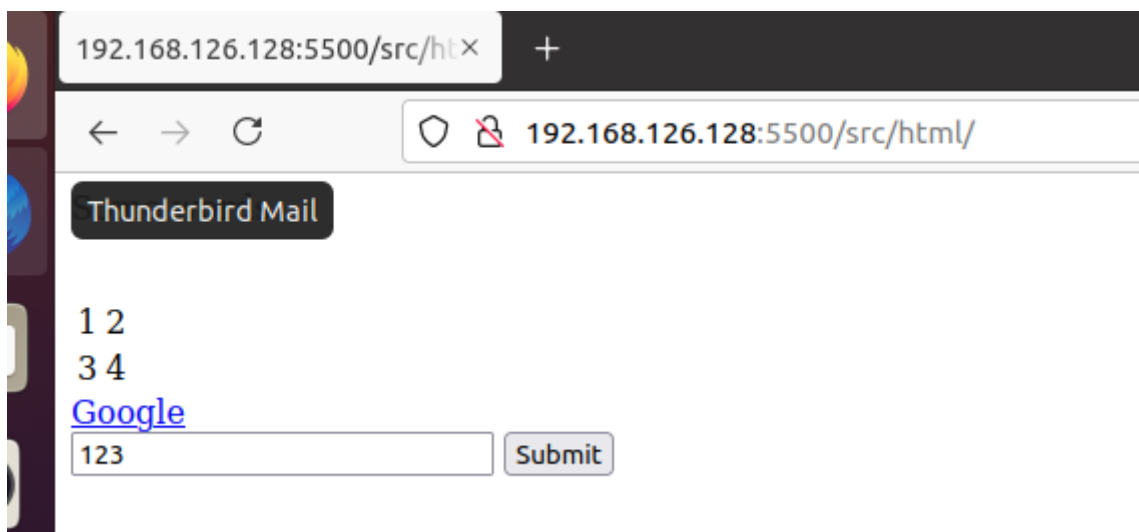
```
(env) melodie@ubuntu:~/Desktop/Django$ sudo ufw allow 5500
Rule added
Rule added (v6)
(env) melodie@ubuntu:~/Desktop/Django$ sudo ufw status
Status: active
```

To	Action	From
--	-----	----
22/tcp	ALLOW	Anywhere
20/tcp	ALLOW	Anywhere
21/tcp	ALLOW	Anywhere
80/tcp	ALLOW	Anywhere
888/tcp	ALLOW	Anywhere
8888/tcp	ALLOW	Anywhere
39000:40000/tcp	ALLOW	Anywhere
8000	ALLOW	Anywhere
5500	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)

running on:

<http://192.168.126.128:5500/>

(open this in Ubuntu or Mac/Win)



bravo

Basic CSS Example

Edit in CSS

```

4      <script>
5          function test(){
6              alert(1234);
7          }
8
9          function demo(){
10             alert(1);
11         }
12     </script>
13     <style>
14         p{
15             color: orange;
16             padding-left: 100px;
17         }
18     </style>
19 </head>
20
21 <body>
22     <p>Some words</p>
23     <br>
24     <table>

```

Reflect in HTML

Some words

1 2

3 4

[Google](#)

sudo bt

in BT environment, upload a template zip, unzip it

then go to <http://192.168.126.128:5500/src/html/site/index.html>

Web Scraping on Formatting

install bs4 and requests

```
src > pycode > getinfo.py > ...
1  import requests
2  from bs4 import BeautifulSoup
3  from time import sleep
4  req = requests.get("http://192.168.126.128:5500/src/html/site/in
5  #print(req.text)
6  print(req.status_code)
```

```
1  import requests
2  from bs4 import BeautifulSoup
3  from time import sleep
4
5  class robot():
6
7      def __init__(self):
8          self.url = "http://192.168.126.128:5500/src/html/site/index.html"
9
10     def getInfo(self):
11         try:
12             req = requests.get(self.url)
13             soup = BeautifulSoup(req.text, "html.parser")
14             layui_col_md21_list = soup.find_all("div", attrs={"class": "list-text"})
15             for info_tag in layui_col_md21_list[2:]:
16                 #content = info_tag.find("div", attrs = {"class": "content"})
17                 print(info_tag.text)
18                 print("--" * 100)
19         except:
20             pass
21
22 if __name__ == "__main__":
23     robot = robot()
24     robot.getInfo()
```