

Debugging Flask in VS Code

Set a breakpoint – by clicking on the line number where you want the program to stop

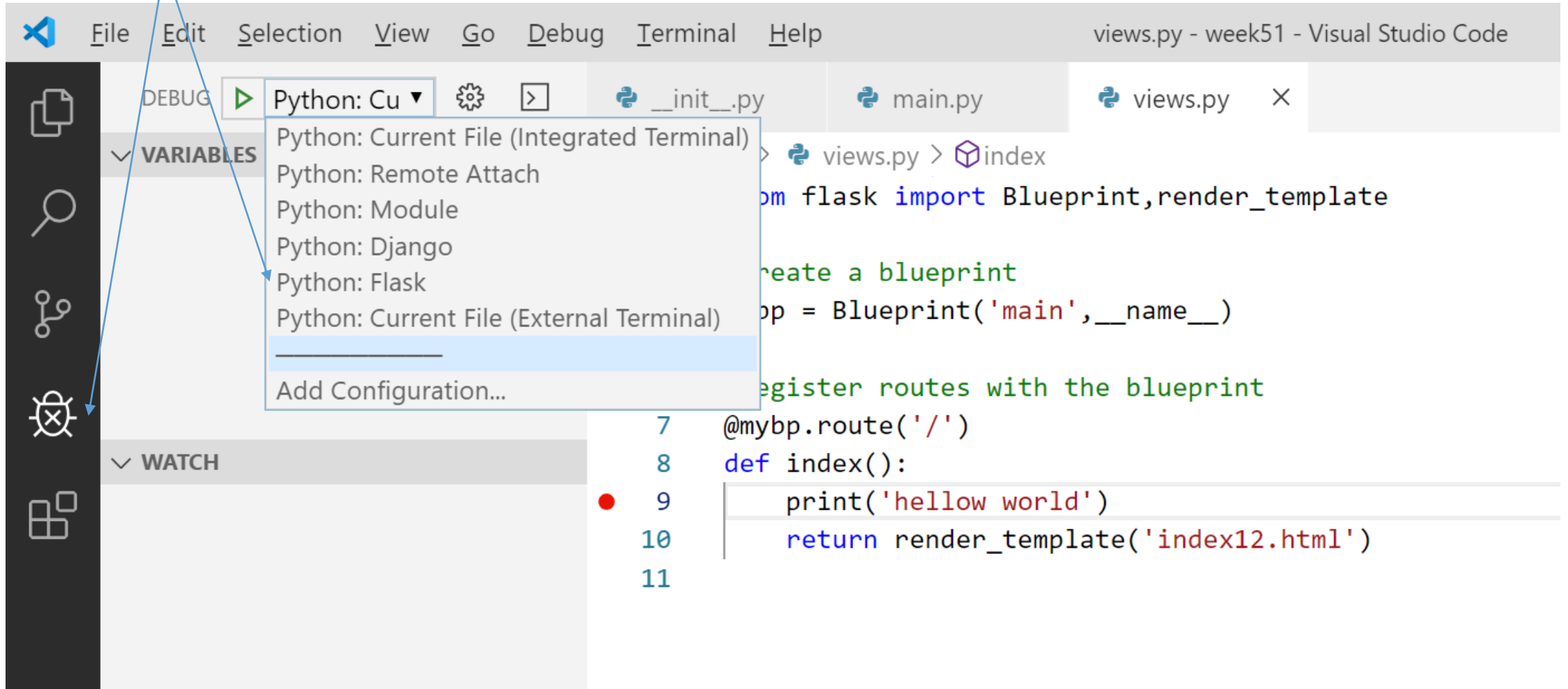
The screenshot shows the Visual Studio Code interface with the following components:

- Menu Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Explorer Panel:** Shows the file structure. Under 'WEEK51', there is a folder 'reservation' containing files like '__pycache__', '.vscode', '__init__.py', 'views.py', and 'main.py'. The 'views.py' file is currently selected.
- Editor Panel:** Displays the code in 'views.py'. The code is as follows:

```
1  from flask import Blueprint, render_template
2
3  #create a blueprint
4  mybp = Blueprint('main', __name__)
5
6  #register routes with the blueprint
7  @mybp.route('/')
8  def index():
9      print('hello world')
10     return render_template('index12.html')
11
```

Line 9 has a red dot breakpoint set next to the line number. The text 'reservation > views.py > index' is shown above the code.
- Bottom Panel:** Contains tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The 'TERMINAL' tab is active, showing a cursor and the number '2:'.

Go to the debug tab, click on Debug -> **Select Python: Flask** from the drop down list



Click the settings button and it would open up a launch.json file

The screenshot shows the Visual Studio Code interface with the `launch.json` file open. The file is located in the `.vscode` directory. The configuration is for a Python Flask application. The `env` section is highlighted with a blue box, and an arrow points to the `"FLASK_APP": "flask_demo.py"` line. The `DEBUG` button is also visible in the top left corner.

```
.vscode > {} launch.json > Launch Targets > {} Python: Flask
42         "--noreload",
43         "--nothreading"
44     ],
45     "django": true
46 },
47 {
48     "name": "Python: Flask",
49     "type": "python",
50     "request": "launch",
51     "module": "flask",
52     "env": {
53         "FLASK_APP": "flask_demo.py"
54     },
55 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

2: Python D

* Serving Flask app "flask_demo.py"
* Environment: production
WARNING: This is a development server. Do not use it in a production environment. Use a production WSGI server instead.
* Debug mode: off
Usage: python -m flask run [OPTIONS]

Set this to main.py as that is the file running the flask app in our case

Select **Debug -> Start Debugging** on the main menu, it would execute until the breakpoint and stop

The screenshot shows the Visual Studio Code interface with a Python Flask application being debugged. The code in `views.py` is as follows:

```
1 from flask import Blueprint, render_template
2
3 #create a blueprint
4 mybp = Blueprint('main', __name__)
5
6 #register routes with the blueprint
7 @mybp.route('/')
8 def index():
9     print('hello world')
10    return render_template('index12.html')
11
```

The debugger has paused at line 9, which is highlighted in yellow. The left sidebar shows the **VARIABLES** panel with **Locals** and **WATCH** sections, and the **CALL STACK** panel showing the current thread `index` in `views.py` at line 9:1. The bottom panel shows the **TERMINAL** output, which includes the command to run the application and the output of the Flask server.

Annotations on the image explain the debug controls:

- Stop debugging**: Points to the red square button in the debug toolbar.
- Step over each line of code**: Points to the blue right arrow button in the debug toolbar.
- Run till the next breakpoint**: Points to the blue right arrow button in the debug toolbar.

The terminal output shows the command to run the application and the output of the Flask server:

```
ions\ms-python.python-2019.8.30787\pythonFiles\ptvsd_launcher.py' '--de
alhost' '--port' '50807' '-m' 'flask' 'run' '--no-debugger' '--no-reloa
* Serving Flask app "main.py"
* Environment: production
WARNING: This is a development server. Do not use it in a production
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```