

More about DNS: <https://www.cloudflare.com/learning/dns/what-is-dns/>

More on HTTP: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>

Pip installation

Ubuntu: `sudo apt install python3-pip`

Others: <https://pip.pypa.io/en/stable/installing/>

Open terminal (in Linux) or command-prompt or powershell(in windows)

[Note] In windows python or python3 command is generally abbreviated and made py  
So use

`py ...`

Instead of

`python3 ...`

## Virtual environment setup

### **Update pip:**

```
python3 -m pip install --upgrade pip
```

### **Install virtualenv**

Mac/linux

```
python3 -m pip install --user virtualenv
```

Windows:

```
py -m pip install --user virtualenv
```

### **Creating**

Linux: `python3 -m venv env`

Windows:

```
py -m venv env
```

## Activating:

linux/mac

```
source env/bin/activate
```

windows

```
.\env\Scripts\activate
```

## Deactivating

Deactivate

## Hello world App

Step1: Install dependencies

```
pip install flask
```

Step 2: Check installed packages

```
pip freeze
```

Step 3: Open a code editor (VScode, pycharm, atom) and create a file named app.py

Step 4: Follow with me or paste the following code:

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return "Hello World"

if __name__ == "__main__":
    app.run()
```

Step 5: Go to the terminal and run the code using the following command

```
python app.py
```

Step 6: Go to the url provide by flask i.e: <http://127.0.0.1:5000/> and verify it is working.

Now we are equipped with concept of working of web app we go further to improve our web app.

## TEMPLATES and LAYOUT

Separation of business logic and presentation logic.

Steps are self explanatory in starting files.

## REQUEST and FORMS

Request captures HTTP messages and provide us in easy format.

Methods are the verbs to describe how those HTTP messages are obtained.

Steps are described in starting files.

## Static Files

These are Javascript, CSS, images, pdf files that are to be accessed from your templates.

Jinja Syntax to include css would be:

```
<link rel="stylesheet" href="{{ url_for('static',filename='style.css')
}} ">
```

Note: *style.css* should be inside *static/* folder in the same directory of *app.py*

Flask Official Documentation: [https://flask.palletsprojects.com/\\_downloads/en/1.1.x/pdf/](https://flask.palletsprojects.com/_downloads/en/1.1.x/pdf/)