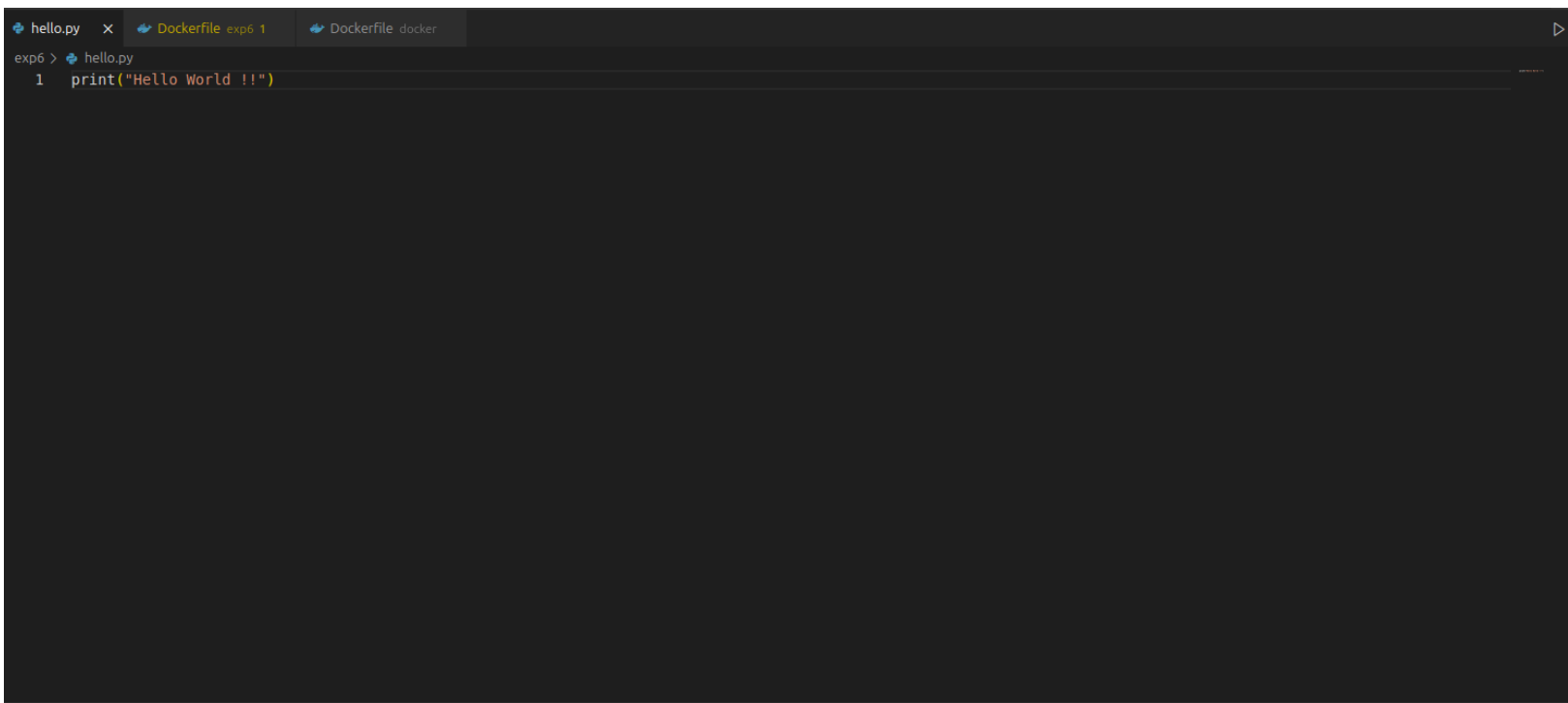


# Experiment No : 6

**Title:** Implementation of Containerization using Docker.

**Objective:** To implement Docker Containerization.

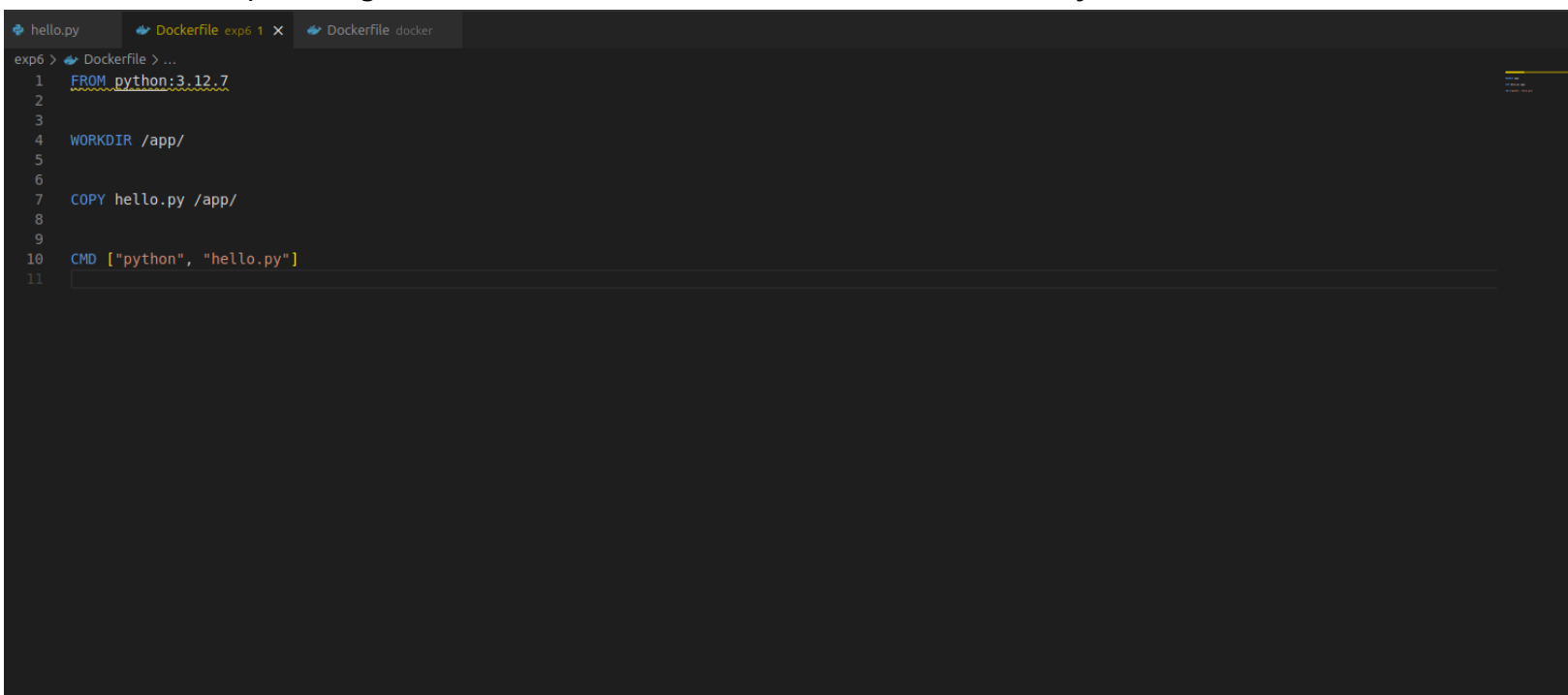
## 1) Creating a python script file named hello.py



The screenshot shows a code editor with three tabs: 'hello.py', 'Dockerfile exp6 1', and 'Dockerfile docker'. The 'hello.py' tab is active, displaying a single line of Python code: `print("Hello World !!")`. The editor's interface includes a dark theme, a sidebar on the left, and a top bar with the file name and a close button.

```
hello.py x Dockerfile exp6 1 Dockerfile docker
exp6 > hello.py
1 print("Hello World !!")
```

## 2) Writing a Docker file named Dockerfile in the same directory.



The screenshot shows a code editor with three tabs: 'hello.py', 'Dockerfile exp6 1', and 'Dockerfile docker'. The 'Dockerfile exp6 1' tab is active, displaying the following Dockerfile content: `FROM python:3.12.7`, `WORKDIR /app/`, `COPY hello.py /app/`, and `CMD ["python", "hello.py"]`. The editor's interface includes a dark theme, a sidebar on the left, and a top bar with the file name and a close button.

```
hello.py Dockerfile exp6 1 x Dockerfile docker
exp6 > Dockerfile > ...
1 FROM python:3.12.7
2
3
4 WORKDIR /app/
5
6
7 COPY hello.py /app/
8
9
10 CMD ["python", "hello.py"]
11
```

# Experiment No : 6

## 3) Building Docker Image using Docker file

- `docker build -t imagename  
directory_where_to_look_configuration_files`

```
(base) anuj@anuj-Bravo-15-C7UDXK:~/Desktop/dockerexp/exp6$ docker build -t hello .
[+] Building 19.6s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 123B
=> [internal] load metadata for docker.io/library/python:3.12.7
=> [internal] load .dockerignore
=> transferring context: 2B
=> [1/3] FROM docker.io/library/python:3.12.7@sha256:8d97a0ab83113f8984517f14492f57186d943db7815682cab9d820c9c0a8c998
=> resolve docker.io/library/python:3.12.7@sha256:8d97a0ab83113f8984517f14492f57186d943db7815682cab9d820c9c0a8c998
=> sha256:eacf19d9b99d130825bc1acb2e42ceee2c3e3aac856984c22d16c549b2337413 248B / 248B
=> sha256:74b0327f38a0efff3bfb2a203b1a4de2e995c6a8da9e9514d0de833aeca583e6 25.57MB / 25.57MB
=> sha256:9d77b800517785b29285f81676acea0c6cc49c4b14df9438b88dab95bf573510 6.16MB / 6.16MB
=> sha256:ce82e98d553dd62ca6a12bebf83992ae9f9ae2748275e74b66a68cc094f868b 211.31MB / 211.31MB
=> sha256:551df7f94f9c131f2fec0e8063142411365f0a1c88b935b9fac22be91af227e0 64.39MB / 64.39MB
=> sha256:5bd71677db44bb63b94de61b6f1f95d5540b4ba2d6a8a6bc4d19f422b25e0c2b 22.02MB / 23.87MB
=> sha256:fd894e782a221820acf469d425b802be26aadb5e5d26ea80a650ff6a974d488 48.50MB / 48.50MB
=> extracting sha256:fd894e782a221820acf469d425b802be26aadb5e5d26ea80a650ff6a974d488
=> extracting sha256:5bd71677db44bb63b94de61b6f1f95d5540b4ba2d6a8a6bc4d19f422b25e0c2b
=> extracting sha256:551df7f94f9c131f2fec0e8063142411365f0a1c88b935b9fac22be91af227e0
=> extracting sha256:ce82e98d553dd62ca6a12bebf83992ae9f9ae2748275e74b66a68cc094f868b
=> extracting sha256:9d77b800517785b29285f81676acea0c6cc49c4b14df9438b88dab95bf573510
=> extracting sha256:74b0327f38a0efff3bfb2a203b1a4de2e995c6a8da9e9514d0de833aeca583e6
=> extracting sha256:eacf19d9b99d130825bc1acb2e42ceee2c3e3aac856984c22d16c549b2337413
=> [internal] load build context
=> transferring context: 58B
=> [2/3] WORKDIR /app/
=> [3/3] COPY hello.py /app/
=> exporting to image
=> exporting layers
=> exporting manifest sha256:82a478e56add8851fa6f44d3d9e50e1f516bc4350540d5b254dc85c9015c7fe9
=> exporting config sha256:efe8f547c86ae433037a2e2cd2f7a1ca40bd60ee2f1119318f92741ae931987b
=> exporting attestation manifest sha256:76d65ef656a710216630eb39efca9eae9b9921627d3fb48618da2d7e071e75b0
=> exporting manifest list sha256:ad09cd0a0dc10b4c940bf14b114e54e788add7e1d5be7329841df057bb209b7
=> naming to docker.io/library/hello:latest
=> unpacking to docker.io/library/hello:latest

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/mvlyixs81jey87c9wkgb7gqk
(base) anuj@anuj-Bravo-15-C7UDXK:~/Desktop/dockerexp/exp6$
```

## 4) Running docker Image :

- `docker run --name container_name image_name.`

```
(base) anuj@anuj-Bravo-15-C7UDXK:~/Desktop/dockerexp/exp6$ docker run --name hellocontainer hello
Hello World !!
(base) anuj@anuj-Bravo-15-C7UDXK:~/Desktop/dockerexp/exp6$
```

## 5) Display the logs of container :

- `docker logs container_name`

```
(base) anuj@anuj-Bravo-15-C7UDXK:~/Desktop/dockerexp/exp6$ docker logs hellocontainer
Hello World !!
(base) anuj@anuj-Bravo-15-C7UDXK:~/Desktop/dockerexp/exp6$
```