

```
#!/bin/bash
```

This is the shebang for the bash script itself. It tells the system that this script should be executed using the Bash shell located at /bin/bash.

```
mkdir -p ~/altair_software_recruitment/executables/
```

As the question suggested me to create a python file I created a directory for the given file extension. So I created the directory path ~/altair\_software\_recruitment/executables/. As a result the py file will be placed in that specific place properly.

```
cat > ~/altair_software_recruitment/executables/hello_altair.py << 'EOF'
```

So I used the cat command where it uses a (<<) - 'here document' to feed multiple lines of text into the cat command which then writes them to the new file "hello\_altair.py". The 'EOF' will protect the bash script by removing any unnecessary scripts.

Then again in the terminal I wrote:

```
#!/usr/bin/env python3
```

This is the shebang for the Python file. It's the first line inside the new Python file.

"#!/usr/bin/env python3" is used to locate the interpreter in the system's PATH.

Then In the python script:

```
print("Hello Altair!")
```

Here's the required task! And to end the document I write : 'EOF'

```
sudo chown root:root  
~/altair_software_recruitment/executables/hello_altair.py
```

This linux command changes the owner and group of the file to root. As Linux permissions are 3 categories : the file's owner (user), the file's group and everyone else . But for the task and in reality for the secured case it is necessary to set the owner as root.

Then in the bash terminal :

```
sudo chmod 700 ~/altair_software_recruitment/executables/hello_altair.py
```

It changes the mode where a specific permission has been granted that is:

The first digit (7) sets permissions for the owner (root). 7 in binary is 111, which means read (4) + write (2) + execute (1) permissions.

The second digit (0) sets permissions for the group. 0 means no permissions.

The third digit (0) sets permissions for others . 0 means no permissions.

By this we can make the root as the owner and grant full permission to him. This is how one can read,write or execute a bash file! One may use other octal permission like 755,711 etc but 700 is better because at the same time root can modify, read and also has full access over it.