

# Command Line Interface

If you have used command line, you can skip this section...

## What is Command Line Interface (CLI)?

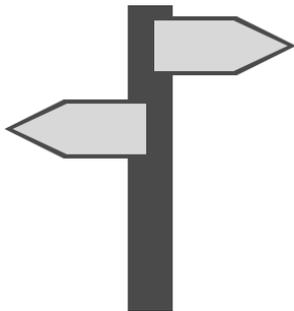
It "is a means of interacting with a computer program where the user (or client) issues commands to the program in the form of successive lines of text (command lines)."<sup>1</sup>

## How to use CLI?

You can use command line via the following, depending on your systems:

- **Windows:** cmd
- **Linux:** Terminal
- **MacOSX:** Terminal/iTerm

## 101 dip into CLI



### Know where you are

(Note: MacOSX & Linux Only; doesn't work on Windows)

Type the following in the terminal:

```
$ pwd
```

That prints the current directory you are at called **print working directory**.

(Windows only) Image below shows where you are already.

```
Command Prompt
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Users\Brennan>
```

<sup>1</sup> [http://en.wikipedia.org/wiki/Command\\_line](http://en.wikipedia.org/wiki/Command_line)



You can find out what is contained in the directory you are in by doing the following:-

(MacOS/Linux only)

```
$ ls
```

(Windows only)

```
c:\Users\Vicky> dir
```

(MacOS/Linux/Windows) You can change directory by typing the following:

```
$ cd a_directory
```

That should be enough for this workshop, if you want to learn more about CLI, Coding Grace has slides from a previous **Beginners CLI workshop**<sup>2</sup>

## If you want us to run a CLI Workshop, drop us an email

✉ [contact@codinggrace.com](mailto:contact@codinggrace.com)

---

# Introduction to Python



Open terminal and type “python” and you should see the following:

```
$ python
Python 3.4.3 (default, Apr 7 2015, 08:05:21)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

If an error appears, try using `python3` instead.

`>>>` means you are in the Python interpreter. You can type Python code and try commands out.

## Now let's write some Python

```
>>> print("Hello")
Hello
```

### Let's try some interaction:

```
>>> input("What's your name? > ")
```

---

<sup>2</sup> <http://bit.ly/1mVUjzG>



**What's your name? >**

It is waiting for your input. So type in your name, and hit **RETURN**:

```
>>> input("What's your name? > ")
What's your name? > Vicky
Vicky
```

## Let's exit the Python interpreter

To do this, type `exit()` or click **Ctrl-D** (i.e. EOF) to exit the interpreter.

It should bring you back to `$` or `>` prompt.

## Writing Python scripts

### Before we start, a couple of best coding practices

Things to note before writing Python code, best coding practices:

- Make sure you have set editor to **4 spaces**<sup>3</sup> as indentation is important in Python.
- Use *spaces* instead of *tabs*<sup>4</sup>.

You can find out more about the style guidelines for Python here:

<https://www.python.org/dev/peps/pep-0008>

### Your first Python script

In your editor, create a new Python script, and save it as `my_game.py`.

```
if __name__ == "__main__":
    main()
```

This allows the script to be run as a reusable modules, or as standalone programs.

To understand this more, let's add more code. Above the code just written, add the following:

```
def main():
    print(input("What's your name? >3"))
```

*(Continued next page)*

---

<sup>3</sup> <https://www.python.org/dev/peps/pep-0008/#indentation>

<sup>4</sup> <https://www.python.org/dev/peps/pep-0008/#tabs-or-spaces>



This is what the full code should look like:

```
def main():
    print(input("What's your name? > "))

if __name__ == "__main__":
    main()
```

Now remember to save the file. And let's go back to the terminal, make sure you are in the same location as your Python script by using **pwd** (if on Mac/Linux), **cd** and **ls**.

To run the script, you can type the following in the terminal:

```
$ python my_game.py
What's your name? > Vicky
Vicky
```

You can also run this code in the Python interpreter

```
$ python
>>> import my_game
>>> my_game.main()
What's your name? > Vicky
Vicky
>>>
```

Now we have the initial basics, let's continue with the rest of the workshop.

## Workshop files

You can find the workshop files here: <http://bit.ly/cg-py-workshop>

## Questions?

✉ [contact@codinggrace.com](mailto:contact@codinggrace.com)



# Resources

## Recommended Editors

- Visual Studio Code (All Platforms) - <https://code.visualstudio.com>
- PyCharm Editor (All Platforms) - <http://www.jetbrains.com/pycharm>

## References

- Python official website - <http://python.org>

## Tutorials

---

### Beginner

- Code Academy (Free) Python Course
    - <https://www.codecademy.com/learn/learn-python>
  - Exercism - <https://exercism.io>
  - Free Code Camp - <https://www.freecodecamp.org>
  - Improve Coding through games - <https://www.codingame.com>
- 

### Post-Beginners

- Self-driven Python tutorials - [newcoder.io](http://newcoder.io)
- Project Euler - [projecteuler.net](http://projecteuler.net)
- A Community that Masters Python through Code Challenges - [pybit.es](http://pybit.es)

## Local Python Events

- Python Ireland: 2nd Wed monthly Python Meetup - [python.ie](http://python.ie)
  - PyCon Ireland - Annual Python Conference
- PyLadies Dublin: 3rd Tue monthly Python Meetup - [dublin.pyladies.com](http://dublin.pyladies.com)

