

# Defensive programming - Tools For Class

---

## The tools

---

In this class we will be using the following tools

- git (version control system)
- [python 2.7](#) (programming language)
- [pylint](#) (Python Linter)
- [Atom](#) (Text Editor) Recommended packages for atom and python
  - autocomplete-python
  - linter-pylint
  - platformio-ide-terminal
  - Atom-beautify

## Git

---

### What is Git?

"By far, the most widely used modern version control system in the world today is Git. Git is a mature, actively maintained open source project originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel. A staggering number of software projects rely on Git for version control, including commercial projects as well as open source. Developers who have worked with Git are well represented in the pool of available software development talent and it works well on a wide range of operating systems and IDEs (Integrated Development Environments).

Having a distributed architecture, Git is an example of a DVCS (hence Distributed Version Control System). Rather than have only one single place for the full version history of the software as is common in once-popular version control systems like CVS or Subversion (also known as SVN), in Git, every developer's working copy of the code is also a

repository that can contain the full history of all changes.

In addition to being distributed, Git has been designed with performance, security and flexibility in mind."

[more](#)

## Installation

Git can be install with relative easy on most every platform. So lets jump in.

### MacOS

MacOS by far has the most options for installing git. The two easiest ways to install git on Mac is with a binary package, or my favorite option the homebrew package manager.

Binaries: Download [here](#)

Homebrew ([get homebrew](#))

```
$ brew install git
```

More installion option can be found [here](#).

Be sure to check that your installion worked.

1. Open your terminal
2. Run the following command

```
$ git version  
git version 2.11.0
```

3. Configure your git client

```
$ git config --global user.name "abarrett"  
$ git config --global user.email "austin@stripedpurple.io"
```

### Windows

1. Download the latest [Git for Windows installer](#).
2. When you've successfully started the installer, you should see the **Git Setup** wizard screen. Follow the **Next** and **Finish** prompts to complete the installation. The default options are pretty sensible for most users.
3. Open a Command Prompt (or Git Bash if during installation you elected not to use Git from the Windows Command Prompt).
4. Run the following commands to configure your Git username and email using the following commands, replacing Emma's name with your own. These details will be associated with any commits that you create:

```
$ git config --global user.name "Emma Paris"
$ git config --global user.email "eparis@atlassian.com"
```

## Linux

### Debian / Ubuntu (apt-get)

Git packages are available via [apt](#):

1. From your shell, install Git using apt-get:

```
$ sudo apt-get update
$ sudo apt-get install git
```

3. Verify the installation was successful by typing `git --version`:

```
$ git --version
git version 2.9.2
```

4. Configure your Git username and email using the following commands, replacing Emma's name with your own. These details will be associated with any commits that you create:

```
$ git config --global user.name "Emma Paris"
$ git config --global user.email "eparis@atlassian.com"
```

---

## Fedora (dnf/yum)

Git packages are available via both [yum](#) and [dnf](#):

1. From your shell, install Git using dnf (or yum, on older versions of Fedora):

```
$ sudo dnf install git
```

or

```
$ sudo yum install git
```

2. Verify the installation was successful by typing `git --version`:

```
$ git --version
git version 2.9.2
```

3. Configure your Git username and email using the following commands, replacing Emma's name with your own. These details will be associated with any commits that you create

```
$ git config --global user.name "Emma Paris"
$ git config --global user.email "eparis@atlassian.com"
```

## From source

Git can also be compiled and install from source.

# Python

---

Python is a widely used high-level programming language used for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy which emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly braces or keywords), and a syntax

which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale.

Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library.

Python is widely used and interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

Example Code:

```
#!/usr/bin/env python

import random
from time import ctime

def random_int(a, b=None):
    random.seed(time.ctime())
    if b is None:
        number = random.randint(a)
    else:
        number = random.randint(a,b)

    return number

print random_int(10, 100)
```

## Installation

Binary packages for python can be downloaded from <https://www.python.org/downloads/> and installed by following the installer.

## MacOS

## Homebrew

1. Open a terminal
2. Run `$ brew install python`.
3. Test the installation by running the following

```
$ python --version  
Python 2.7.10
```

**Note:** you should be running version 2.x

4. Run `$ python`
5. Once in the console run

```
print "Hello world!"
```

you should see the phrase **Hello world!** print on the console

6. Type `exit()` to leave the python console

## Windows

1. Simply download the binary from <https://www.python.org/downloads/> and follow the prompts
2. Test the installation by opening **IDLE** and running the following

```
print "Hello world!"
```

you should see the phrase **Hello world!** print on the console

3. Type `exit()` to leave the python console

## Linux

Python comes with most linux distros including Debian, Ubuntu, and Fedora. You should check to make sure the default version of python is 2.x

**NOT** 3.x. This can be accomplished by running the following from the terminal `$ python --version`. Please consult the internet for help with switching your default version.

# Pylint

---

## What is pylint?

[Pylint](#) is a full feature python code analysis tool. Pylint has many features including

- Coding Standard
- Error detection
- Refactoring help
- Fully customizable
- Editor/IDE integration
- UML diagrams
- Etc.

## Installation

Full list of installation steps for most platforms can be found [here](#).

# Atom Text Editor

## What is Atom?

Atom is a text editor that's modern, approachable, yet hackable to the core —a tool you can customize to do anything but also use productively without ever touching a config file. [Download](#)

## Installation

Simply download and install the binaries from [atom.io](#).

Once installed I would recommend install some additional packages. Press `Cmd+,` on Mac or `Ctrl+,` on all other platforms to open the setting tab. Locate the **Install** section in the sidebar. From the search box find the following packages and click install.

- autocomplete-python (gives suggestions for what you are typing)
- linter-pylint (intergrates pylint into atom)
- platformio-ide-terminal (provides a terminal window inside atom)
- Atom-beautify (Cleans up code)

## Addition Settings and Packages to Consider

### Settings

#### Autosave

1. From the Settings tab (Ctrl+, on PC or Cmd+, on MacOS) locate the **Packages** pane.
2. In the search box type **autosave** and press enter
3. Select the setting button in the autosave box
4. locate checkbox labeled **Enable** and check it

#### Indenting and tabs

1. From the Settings tab (Ctrl+, on PC or Cmd+, on MacOS) locate the **Editor** pane.
2. Locate the checkbox labeled **Show Indent Guides** and check it
3. Locate the the input field labeled **Tab Length**
4. In the input box replace the content with the number **4**

#### Scroll past (This is just one of my preferences)

1. From the Settings tab (Ctrl+, on PC or Cmd+, on MacOS) locate the **Editor** pane.
2. Locate the checkbox labeled **Scroll Past End** and check it

### Packages

#### Atom-Beautify

1. From the Settings tab (Ctrl+, on PC or Cmd+, on MacOS) locate the packages pane.
2. Locate the **Install** pane
3. In the search box type **atom-beautify** and press enter
4. Click install in the package box labeled **atom-beautify**



## Auto-Runner

1. From the Settings tab (Ctrl+, on PC or Cmd+, on MacOS) locate the packages pane.
2. Locate the **Install** pane
3. In the search box type **auto-runner** and press enter
4. Click install in the package box labeled **auto-runner**

## Done

---

That's it! Have fun