

# Productivity Tools

Creativity Through Code

2015-10-30

## Abstract

Before delving into the actual specifics of web development, we will be introducing various productivity tools as well as a “development environment” to make your life easier.

## 1 Programming Environment

**Brackets** is a modern, lightweight, and open source text editor. Some of the very convenient features that it supports include inline editors, live preview, and preprocessor support. Overall, Brackets is a very powerful editor out of the box and it is highly recommended for this course.

**Download Link:** <http://brackets.io/>

**Atom** is GitHub’s recent “hackable” editor. It supports cross-platform editing, built-in package manager, smart autocompletion, file system browser, and multiple panes. It is also easily customizable with thousands of open source packages. Definitely choose this editor if you want to adjust your editor to suit your liking.

**Download Link:** <https://atom.io/>

**Sublime Text** is a very sophisticated and beautiful editor. It can support additional packages with its package manager, but the salient aspect about Sublime Text is its slick user interface and amazing features. Sublime text supports Goto Anything, Multiple Selections, Command Palette, Distraction Free Mode, and Split Editing.

**Download Link:** <http://www.sublimetext.com/>

## 2 Meteor

**Meteor** is a complete platform for developing and building web and mobile apps in Javascript. We will be using Meteor throughout the course.

**Download Link:** <https://www.meteor.com/>

## 3 GitHub

**GitHub** is an essential tool for any programmer that is used for distributed version control, collaborative coding, and source code management. Projects can be uploaded to GitHub as repositories. We will be learning about both the desktop graphical interface and the command-line tool. For a list of commonly used git commands, please refer to Table 1: Commonly Used Git Commands.

Table 1: Commonly Used Git Commands

Git Task	Command	Meaning
Initialization	<code>git config --global user.name "Jeffrey Xiao"</code>	Configures the author name to be used with your commits
	<code>git config --global user.email j@x.com</code>	Configures the author name to be used with your commits
Adding a repository	<code>git init</code>	Creates a new local repository
	<code>git clone /path/to/respository</code>	Creates working copy of a local repository
	<code>git clone https://github.com/USER/REPO</code>	
Adding files	<code>git add -A</code>	Adds all files to stage
	<code>git add "filename"</code>	Adds "filename" to stage
Committing	<code>git commit -m "Commit Message"</code>	Commits changes to head (but not yet to the remote repository)
	<code>git commit -a</code>	Commits any files you have added with git add or changed
Pushing	<code>git push origin master</code>	Sends changes to the master branch of your remote repository
Branches	<code>git checkout -b "branchname"</code>	Creates a new branch and switches to it
	<code>git checkout "branchname"</code>	Switches from the current branch to branchname
	<code>git branch</code>	Tells you what branch you are in currently and lists all the branches in your repo
	<code>git branch -d "branchname"</code>	Deletes branchname
	<code>git branch push origin "branchname"</code>	Pushes branchname to your remote repo
	<code>git branch push --all origin</code>	Pushes all branches to your remote repo
	<code>git branch push origin :"branchname"</code>	Deletes branchname from your remote repo
Updating Local Repo	<code>git pull</code>	Fetches and merges changes from your remote repo to your working directory
	<code>git merge "branchname"</code>	Merges branchname into your active branch
	<code>git diff</code>	Views all merge conflicts