CS 29 Resources:

### Lambda School Complete CS Wiki:

<https://github.com/LambdaSchool/CS-Wiki/wiki>

### What to have done by Monday:

#### Install python 3 and pipenv (pipenv is the python way of NPM installing- it creates a virtual environment so you can have different versions of python and other things running on different projects) **HEADS UP: Windows Python installation can be on the more difficult side**

<https://github.com/LambdaSchool/CS-Wiki/wiki/Installing-Python-3-and-pipenv>

#### Review these links to be better prepared:

##### U.P.E.R. Problem solving framework

* + - <https://github.com/LambdaSchool/CS-Wiki/wiki/Lambda-Problem-Solving-Framework>

##### Computing Big-O notation

* + - <https://github.com/LambdaSchool/CS-Wiki/wiki/Computing-Big-O>

##### How to ask questions

* + - <https://zellwk.com/blog/asking-questions/>

##### Lambda School Python/Javascript Cheatsheet

* + - <https://github.com/LambdaSchool/CS-Wiki/wiki/Javascript-Python-cheatsheet>

#### Review Day 1 of Intro to Python as is can take a few hours to go through the material for Monday’s lecture

* + <https://learn.lambdaschool.com/cs/module/recay2erzDlYUPSeO>

#### Fork and clone Intro to Python

* + <https://github.com/LambdaSchool/Intro-Python-I>

### 

### Reading Resources:

#### Lambda School Optional CS Reading list

* + <https://github.com/LambdaSchool/CS-Wiki/wiki/Computer-Science-Reading-List>

#### Grokking Algorithms

* + <https://github.com/RbkGh/Free-Algorithm-Books/blob/master/book/Grokking%20Algorithms%20-%20An%20illustrated%20guide%20for%20programmers%20and%20other%20curious%20people.pdf>

#### Problem Solving with Algorithms and Data Structures

* + <https://www.cs.auckland.ac.nz/compsci105s1c/resources/ProblemSolvingwithAlgorithmsandDataStructures.pdf>

#### Data Structures and Algorithms Using Python

* + <http://home.ustc.edu.cn/~huang83/ds/Data%20Structures%20and%20Algorithms%20Using%20Python.pdf>

#### Automate the Boring Stuff

* + <https://1.droppdf.com/files/5iHzx/automate-the-boring-stuff-with-python-2015-.pdf>

#### Big O Notation

* + <https://rob-bell.net/2009/06/a-beginners-guide-to-big-o-notation/>

#### Pipenv Docs

* + <https://pipenv-fork.readthedocs.io/en/latest/index.html>

#### Python PEP8 style guide

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

#### Python environment manager Anaconda

* + <https://docs.anaconda.com/anaconda/install/>

#### Python Writing Gotchas

* + <https://docs.python-guide.org/writing/gotchas/>

#### Cracking the Code Interview

* + <http://ahmed-badawy.com/blog/wp-content/uploads/2018/10/Cracking-the-Coding-Interview-6th-Edition-189-Programming-Questions-and-Solutions.pdf>

#### Code: The Hidden Language of Computer Hardware and Software (for Computer Architecture)

* + <https://bobcarp.files.wordpress.com/2014/07/code-charles-petzold.pdf>

### Algorithm Resources:

#### Data Structure Visualizations

* + <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>

#### Sorting Visualization

* + <https://visualgo.net/bn/sorting?slide=1>

### Extra Learning Resources:

#### Kaggle

* + <https://www.kaggle.com/learn/overview>