**Python!**

**Tentative schedule:**

Week 1:

Fundamentals:

Spend 1-2 days learning syntax. Syntax simply means the vocabulary of the language.

Practice the fundamentals of programming that you've already been introduced to in your morning algorithms.

Intro to Flask:

Learn how to make a web server using the Flask framework. You'll create web pages that can accept data sent from a client and return a response. This quick intro will take about 2 days.

MySQL:

Spend 1-2 days to learn about MySQL, a commonly used relational database.

Learn to create visual representations of your database.

Formulate SQL queries using database creation and management software.

Week 2:

Flask + MySQL:

Extend your use of Flask to include data persistence using a database as storage.

Challenge yourself to complete The Wall, an assignment that brings together all the skills and tools you've learned so far.

Spend two days completing this section.

Object Oriented Programming:

Learn OOP, a style of organizing your code that is prevalent in web development.

Don't underestimate how important it is to develop a solid understanding of OOP, which is a pivotal skill. This is why you spend the rest of your time at the bootcamp building on your understanding.

Frameworks like Django, Rails, Angular, and iOS are built using an object oriented design pattern.

Django:

The framework you've all been waiting for!

You will learn and use code modularization to organize your code according to the purpose it serves.

Learn to build a project in Django.

Week 3:

Django:

Continue with Django for the rest of the week leading up to your exam.

Learn to store data in a SQLite database.

Learn to use a object relational mapping (ORM) to communicate with your database.

Learn to add a MySQL database to your project.

Deployment:

Learn to move your code (previously hosted on your computer) to the web so everyone can see the product of your hard work!

You will be required to deploy your belt exam.

Belt Exam:

Friday of week 3, you'll be ready to take your belt exam.

Your instructor will give you more details as the date approaches.

Try not to worry about this for now. The exam is low-pressure and can be retaken as many times as you need.

The exam is open-web – meaning you have all of your previous code and the entire internet at your disposal!

Week 4:

Retake your belt exam

You'll have this week to retake the belt exam as many times as you need

Do a project:

After passing your belt exam, you're ready to do a project!

Your project is your motivation to earn your belt. Complete your belt quickly in order to spend as much time as possible working on a fun project. Ask your for project ideas if you need help.

Create anything you want using your new skills. The sky's the limit!

Why Python?

* Readability
* Libraries
* Community
* Scope
* Ease

Core Philosophy:

* Beautiful is better than ugly
* Explicit is better than implicit
* Simple is better than complex
* Complex is better than complicated
* Readability counts

Why Python is Good for You:

* Popular
* Shallow learning curve
* Minimal setup
* Understandable
* Fast development cycle

Issues with running Python script in bash shell:

1. Python Shell hangs on initialization

Resolution: Adding an alias to your .bashrc file.

1. Print statements do not appear.

Workarounds: (1) flush the output buffer after each print statement (i.e. “sys.stdout.flush()”); (2) if running code from document using “python filename.py”, use “python –u filename.py”.

**When creating a variable in python, you don’t need to use “var” like in Javascript.**

**“Array” in Javascript == “List” in Python**

**“Console.log” in Javascript = “Print” in Python**

For comments:

* Single line - #
* Multiple lines – Triple quotations (“”” “”” or ‘’’ ‘’’)

Data Types:

Primitive data types:

* Boolean values
* Numbers
* Strings

Composite types:

* Tuples: type of data that is immutable and can hold a group of values. Tuples can contain mixed data types.
* Lists: A type of data that is mutable and can hold a group of values. Usually meant to store a collection of related data.
* Dictionaries: A group of key-value pairs. Dictionary elements are indexed by unique keys which are used to access values.

Strings:

Print inserts a space between elements separated by a comma.

Concatenate the contents into a new string with the help of +.

Lastly, you can use curly brackets - {} - and the string **.format()** method to inject variables into your string - this is known as **string interpolation**.

first\_name = "Zen"

last\_name = "Coder"

print "My name is {} {}".format(first\_name, last\_name)