



Byte Academy Tech Syllabus Timeline

Full-Stack Python

Week 1&2: Language Basics

- Programming basics: from Key Concepts: Names & Objects, data structure, Iteration, Recursion, Control flow, Class Organization, to Running Python
- Writing Python using Git & Github
- Practicing Best Practices: Keeping it simple, writing short tests and good documentation, logging
- In class Quizzes
- Start with Python mini project: Well-documented Python module; (those are one to two-hour mini project which could be finished in class so that the students can discuss with peers and get feedback from the teacher/TA.)
- Start with Standard Library (standard libraries typically include definitions for commonly used algorithms, data structures, and mechanisms for input and output.)

Week 3&4: Standard Library and Framework

- Explore Python Standard library like IPython, Virtualenv, and the Python ecosystem
- In class Quizzes
- Django framework (modeling relational data, building admin interfaces)
- Python development on Django (online Tutorial: [Start with Django](#), [How I use Git and Django \(and GitHub\)](#) and [Starting a Django Project the Right Way](#))
- Project: building and deploying a simple static website using Git and Github
- Google App Engine: Tools and Webapps
 - Using NDB and templates, MVC style. (quick start)
 - Data Modeling on a non-relational DB
 - Offline operations with Task Queues
 - Project: SQL or NoSQL Server Side Application

Week 5: Front-End Development

- [Chrome developer tools \(DevTools\)](#) -DevTools are a set of web authoring and debugging tools built into Google Chrome. The DevTools provide web developers deep access into the internals of the browser and their web application. Use the DevTools to efficiently track down layout issues, set JavaScript breakpoints, and get insights for code optimization.

- CSS preprocessing - a front-end technology which provides predefined functions for transformations, selector nesting, variables, and mix-ins. It allows you to transfer the CSS code into something more like real programming language: having structure, variable and other features. It makes the style code modularized, portable, reusable and be developed much faster.
- [Canvas Interface](#); (simple interface to most common graphical objects, and matplotlib functions)
- How to use [Google Pagespeed](#) - fast and optimized pages lead to higher visitor engagement, retention, and conversions. The PageSpeed family of tools is designed to help you optimize the performance of your website. PageSpeed Insights products will help you identify performance best practices that can be applied to your site, and PageSpeed optimization tools can help you automate the process.
- Project: Single Page App/Advanced Website

Week 6&7: Plugging into the Web Ecosphere

- From embedding to JavaScript APIs: Youtube, Google Analytics and Google maps. (This will teach you how to embed these popular website/tools as a module in your own through JavaScript APIs)
- REST: RSS, Flickr, Twitter, Google Data APIs - explore the REST APIs provided by all famous social networks. With them you could obtain the most recent hot topics in all the tweets or get the most liked pictures in Flickr
- Design and develop a Mini App: Using one from the JavaScript APIs and one of the REST protocol
- Security: Server (personal/sensitive data) and client (XSS, CSRF attacks)
- Analytic Papers (topic not decided yet): could be about all the famous sites & apps
- Project: Facebook/any social media alike app (It requires the knowledge and developing experience gained in the previous weeks)

Week 8: Useful Open Source Toolset

- Linux theory and practice: distributions, package repositories, bash, ssh and grep
- Automatic deployment with [Fabric](#) (Fabric is a Python (2.5-2.7) library and command-line tool for streamlining the use of SSH for application deployment or systems administration tasks. It provides a basic suite of operations for executing local or remote shell commands (normally or via sudo) and uploading/downloading files, as well as auxiliary functionality such as prompting the running user for input, or aborting execution.)
- Automating testing with [Selenium](#); Simple load and performance testing
- Final Project part 1/3: From ideas to mockups

Week 9&10: Going deep

- Explore advanced topics through guided self-study and practice, choose one

- Building jQuery plugins; Coffeescript or dart; AngularJS, d3.js or processing
- Optimizing, caching; scaling; MongoDB
- HTML5 mobile apps
- HTML video and audio
- Async servers: tornado (or node.js)
- Anything else, or just give some more love to an open source project
- Final Project part 2/3: Design, Code, peer-review

Week 11&12: Final Project Presentations and Mock Interviews

- Final Project part 3/3: Presentation and feedback collection
- Interview-focused Algorithm Lectures, guidance on job interviews
- Meet the industry: Guest Speakers/Presentations by startups and web firms

