Kevin Chen

1/29/17

Notes

Getting Started

* Types of web developers
  + Front-end developers work with languages that render in the browser (or client). Languages: HTML, CSS, JavaScript
  + Back-end development: development on a server in the background.
  + Web stack: collection of tools (both front-end and back-end) used for web dev.
* The role of HTML
  + Defines the content and defines the structure within a webpage
  + Hypertext: documents on the web are not consumed in a linear fashion.
  + Information architecture: structural design of a website’s navigation
  + Markup language: set of tags added to the text of a webpage to define its structure. Roles: defines the role each piece of content has. It also generates the Document Object Model (DOM), which describes the structure of the webpage.
* The role of CSS
  + User Experience: deals with the look and feel of the site
  + User Interface: deals with the functionality of the site.
  + Design for a website deals with CSS. Made up of two parts: defining a set of rules (or style sheets) for how the DOM is translated into visual form AND the cascading style rules (the set of rules that describe the priority of how the styles are rendered on a page).
  + Involves learning to work with elements of motion and interaction.
* The role of JavaScript
  + For front-end and back-end. JavaScript is responsible for managing interactions within a browser.
  + Interacts with DOM handling events, managing media, and managing elements in a page.
  + Always a changing language. Deal with browser support.
  + On back end, it is used as a way to manage and control the development experience (e.g. compressing/combining code and using Node.js, WebPack, and Gulp.js).
* Back end essentials
  + Use of databases. Manage web page based on info from the database.
  + The set of tools each companies used (the stack) varies a lot.
  + Need to study at least one of the platforms for back end development and one for dealing with data. PHP and MySQL is a good choice. Then go Node.js, .NET, or Ruby on Rails
* Servers
  + Manage the interaction between data and how that data is presented.
  + Run software that manages the scheduling of content.
  + Some platforms are OS specific. Two main options: Linux (more popular and less expensive) and Windows (popular with large companies, great dev environments)
  + For servers, can use PHP and MySQL or use Node.js (as the server) and Express.js.
* Server-side scripting
  + Back end developers work with server-side scripting: running with code that runs on the server.
  + Main server-side web development languages: PHP, Node.js, ASP.NET, Ruby, Python. Start with PHP or Node.js. ASP.NET for Windows servers.
* Database languages
  + Most servers use an RDMBS called SQL. This language creates a series of tables with different fields, creating relationships between the data.
  + MySQL, MS SQL, Oracle, PostGresQL.
  + NoSQL: records with no specific structures. These languages are sometimes referred to as Document Stores. Languages include Firebase (from Google), MongoDB, Couch Base, and Couch DB.

Learning and Training

* Train for the industry
  + Instructor led training, self-study, working on projects, and learning that happens as a result of working with clients.
  + Learn to read documentation, examine others’ code, and explore new ideas
  + Have to build real projects. Learn to work with others.
* Traditional education
  + Having a college education does make a big difference.
  + But less than half of developers have a degree in CS. But higher degrees = more money.
  + Benefits: teaches students how to think and learn by themselves AND gives students a firm and wide foundation (e.g. math and physics)
* Coding bootcamps
  + Designed to quickly teach the practical skills needed in the industry.
  + Good results; often teach skills not taught in school.
  + But camps have low acceptance rate and are intensive. Only good if you know you want to be a software developer
* Other learning options
  + The web, online groups (e.g. Slack), Stack Overflow, books and online courses, book subscription services (e.g. O’Reilly Safari), E-books, online training, work on existing projects (e.g. Github) by forking, Codepen and JSbin (training sites)
* Social learning opportunities
  + Need to make contacts within the industry. Employee referrals are great.
  + Local meetings for developers: meetup.com.
  + Conferences: barcamp.
  + Hackathon
* Core developer tools
  + Master your editor (IDE or text editors), the terminal (many apps offer CLI’s, or command line interfaces – see Learn the Linux Command Line: The Basics video course), web browser (good for testing and debugging)
* Other essential tools
  + phpMyAdmin lets you manage databases through an interface
  + Version Control: Git and Github
  + Build tools: create workflows for common tasks (see Web Project Workflows with Gulp.js, Git , and Browserify)
  + Frameworks: libraries of pre-build code that help you build sites faster: Bootstrap, jQuery, JavaScript frameworks like Angular.js and React
  + WordPress: a content management system.
* Core concepts
  + Accessibility: create sites for everyone (e.g. disabled people). See Foundations of UX: Accessibility
  + Agile workflows: produce software in two-week sprints to respond to unpredictability of standards. See Agile Project Management
  + Test-driven development: create tests at the same time you create features. See Foundations of Programming: Test-Driven Development
* Business and leadership skills
  + Communication skills. Get different perspectives. Raise issues without being confrontational. See Connecting with Peers in the Workplace.
  + Present and defend ideas. See How to Present and Stay on Point AND Public Speaking Fundamentals.
  + Profile and LinkedIn and Github. See How to Rock Social Media.
  + See Negotiating Your Salary

Getting a Job

* Types of jobs
  + Remote workers have more job satisfaction, but are uncommon.
  + Want not just good salary, but also work/life balance and company culture.
* Maximize compensation
  + Working at a large company gives you more money.
  + Full-time employees earn more than self-employed or freelance developers.
  + More experience helps. Back-end earns a little more than front-end. Management and executives earn even more.
  + Education lets you earn more.
* Land a job
  + Best way to get a job is through a referral. Thus, interact with other developers.
  + Recruiters. See the course Making Recruiters Come to You
  + Seek the opportunity directly. Try to connect with people in the company (e.g. through social media). See Job Hunting Online.