Carli Spina

Learn computer programming and Web design
Choosing the best resources

Once of interest mostly to computer scientists, computer programming and Web design topics have started to percolate into the general consciousness in recent years. This newfound popularity was first noticed by many when Codecademy launched its Code Year initiative on January 1, 2012, blowing past 100,000 registrants in 48 hours and even drawing the attention of New York City's Mayor Michael Bloomberg.[1] But, this site is just one example in the rapidly developing landscape of online computer programming resources.

Now, a plethora of tools exist for teaching oneself a wide range of programming languages and Web design skills. Whether you want to design a Web site, create a game, visualize data, or build a mobile app, there is a resource out there to help you. In fact, so many resources exist that it can be difficult to determine which tools are best suited to your goals.

The first step to picking the best resource for your purpose is to decide how you hope to use your new skills. To do so, it is important to understand the difference between programming languages and HTML/CSS. HTML and CSS define the structure and style of online documents. They are important tools for designing usable and engaging Web sites, but they are not programming languages because they do not allow you to express an algorithm or tell a computer how to complete a task.[2]

This article will highlight a variety of resources that will help beginners to get started learning a range of computer programming languages and, in many cases, Web design skills. Whether you are interested in suggesting resources for students who are taking a computer programming class at your institution, offering a guide for those who want to teach themselves these skills outside of the classroom, or simply wanting to advance your own skills for professional development purposes, this guide will offer resources tailored to these needs. Most of the resources mentioned here are free; those that require a subscription for full access to all aspects of the service will be noted.

Organization Web sites

Nonprofit organizations that are devoted to supporting those interested in learning programming skills or maintaining the languages themselves can be some of the best sources of information for those interested in learning to write code.

• **Code.org.** This nonprofit organization is probably best known for the video it released showing famous figures, including Bill Gates and Mark Zuckerberg, talking about their experience learning to code and encouraging people to try computer programming. The organization also curates an impressive collection of resources and links for both teachers and others interested in

Carli Spina is emerging technologies and research librarian at Harvard Law School Library, e-mail: cspina@law.harvard.edu

getting started with computer programming and Web design. Resources are divided by topic and include tools to learn both programming languages and Web design skills. *Access:* http://code.org/.

• **Mozilla Developer Network.** This site is a wiki designed to allow anyone to access and contribute to the ever-growing collection of documentation, tutorials, and other educational materials for the programming language JavaScript, as well as HTML and CSS. For those who are just getting started with JavaScript or HTML/CSS, the site offers a good combination of introductory materials and links to tutorials. It is a good site for users at various levels of proficiency, and especially those who are interested in getting involved with the Mozilla Developer Network community. *Access:* https://developer.mozilla.org/en-US/.

• **Python.** The official site of the Python programming language is the best source for information about choosing the optimal version of Python for your needs, as well as how to install it in Windows, Mac, or other environments. It also offers access to the full documentation that is available for Python and additional introductory materials for those who want to move beyond the basics. In addition to these technical materials, the site is also a great source for connecting with the Python community by joining a mailing list, contributing to the wiki, or finding a local user group. *Access:* http://www.python.org/.

• **Ruby.** Much as with Python, the Ruby community has a Web site that offers access to the official documentation for the language, installation information, third-party libraries, and tutorials. This site also offers many ways to interact with and participate in the Ruby community through links to the Ruby IRC Channel, local user groups, blogs, e-mail lists, and information on conferences for those who want to meet in person with other Ruby users. Newbies will particularly appreciate the fact that the site links to several tutorials for beginners on each page, including one that helps those who are experienced in other programming languages to make the transition to Ruby more easily. As of the time of writing, the content had been translated into 15 different languages. *Access:* https://www.ruby-lang.org/en/.

Tools for visual learners

For those who are completely new to computer programming, the idea of writing code can be a bit intimidating. These tools can make the process more accessible through their use of drag-and-drop elements and a visual approach to teaching basic programming concepts.

• **Google Blockly.** Described as a "graphical programming editor," Google Blockly makes use of blocks that are similar to the pieces found in Scratch (see below) to create programs. What sets this apart from Scratch is the fact that programs that are created using this drag-and-drop technique can then be exported as JavaScript, Python, or XML code, which can help users to make the connection between the graphical interface and these languages. All of the code is open source, making it possible for educators to use it in their own projects. *Access:* https://code.google.com/p/blockly/.

• **MIT App Inventor.** Using a similar interface to that found in Google Blockly, the MIT App Inventor lets users create apps for Android devices using a drag-and-drop interface. The entire process from idea to functional app can be done without writing a single line of code, making it more approachable for beginners. The site also offers a number of tutorials and curricula for using the tool in courses at all levels, including undergraduate and graduate levels. All of the code for the MIT App Inventor is open source and available on the site. *Access:* http://appinventor.mit.edu/.

• **Scratch.** Scratch was created by the MIT Media Lab's Lifelong Kindergarten Group as a tool to introduce programming concepts to children. Rather than requiring users to write code, Scratch makes use of puzzle pieces that represent different computer programming concepts, such as loops and variables. Users

drag and drop the puzzle pieces to create interactive games and projects. Puzzle pieces fit together only in selected combinations to create a working program, teaching users the basic concepts that are central to programming in any language. Even though it was initially developed for children, Scratch can be, and is, used by people of all ages as a painless introduction to the basic concepts of computer programming. Scratch also offers a community for users and resources for educators. *Access:* http://scratch.mit.edu/.

Interactive tutorials and practice exercises

Those who are ready to jump into writing their own code or are looking for additional practice to supplement coursework will be happy to find that there are many interactive tutorials and practice lessons available. Each of these tools allows users to practice programming skills in a Web browser without installing any programs on their computer.

• **Codecademy.** Perhaps best known for the Code Year initiative, which aims to teach general computer programming topics to beginners, Codecademy has since expanded to include much more. Courses are available for JavaScript, JQuery, PHP, Python, and Ruby, as well as lessons that walk users through Web design, APIs, and projects that combine multiple topics. As a bonus for those interested in teaching computer programming skills, Codecademy also offers the ability to create lessons either alone or in collaboration with other users. The Codecademy blog includes stories about how users have applied the skills they learned on Codecademy and brief lessons on additional topics. All content on the site is free. *Access:* http://www.codecademy.com/.

Codecademy

• **Code School.** Combining videos and downloadable slides with in-browser exercises, Code School offers courses on a wide range of skills related to Web design and development. Tutorials cover such diverse topics as Git, R, JavaScript, Ruby, Python, HTML, CSS, and Objective-C, to name just a few. In addition to these full courses, it also offers shorter screencasts on related topics. While courses can be completed in any order and combination, Code School also specifies curricula or "Paths" for those interested in a more guided curriculum. Full access to all of the courses and screencasts requires a subscription, but Code School also offers a number of free courses. At the time of writing, these free courses covered the basics of JavaScript, JQuery, Objective-C, R, Git, and Ruby. *Access:* https://www.codeschool.com/.

code school
learn by doing

• **CodingBat.** Created by Nick Parlante, lecturer in computer science at Stanford University, this bare-bones site offers practice exercises for anyone learning Java or Python. The site offers a limited number of training videos and teaching information, but it is not intended as a fully featured teaching tool. Rather, it focuses on offering additional practice exercises for those who already know the basics and want to complete drills to solidify their skills. For teachers, the site also offers an authoring feature that makes it possible to create customized exercises, the option to create collections of exercises tailored to a specific class, and the ability for students to share their completed exercises with their teachers. *Access:* http://codingbat.com/.

• **LearnStreet.** This site offers courses in JavaScript, Python, and Ruby with extensive practice exercises. But what particularly stands out about this service is the Code Garage Projects, a section devoted to more than 90 different projects that let users build games, tools, and more with their newly developed programming skills. The projects are

sortable by language and proficiency level, which makes this appealing even for those who have learned to program in class or on another site. LearnStreet also offers resources for teachers to integrate the interactive content into their courses. *Access:* http://www.learnstreet.com/.

Online courses and MOOCs

Some learners prefer a more traditional class structure complete with lectures, written materials, and assignments. Fortunately, there are many such options available for free online.

• **Google's Python Class.** Nick Parlante, creator of the CodingBat tool mentioned above, also created materials for a course that was offered at Google for employees who had some basic programming experience and were interested in learning Python. All materials for this course, including lecture videos, written materials, and practice exercises have been released for free online under a Creative Commons Attribution 2.5 license.[3] *Access:* https://developers.google.com/edu/python/.

• **MIT OpenCourseWare.** MIT maintains the MIT OpenCourseWare platform, a freely accessible repository of materials from most of MIT's classes. As of the writing of this article, this included materials from more than 2,000 courses[4] on a wide range of subjects from the humanities to the sciences, including computer science courses at both an undergraduate and graduate level, listed under the topic of engineering. Languages covered include C, C++, Java, and Python, among others. While the materials available for any given class will vary, many classes include at least lecture materials and assignments, and some also offer answer keys. *Access:* http://ocw.mit.edu/index.htm.

Getting help and connecting with others

Computer programming and Web design can often seem like solitary pursuits, but in reality there are many online communities for those looking for opportunities for collaboration, support, and community.

• **GitHub.** Whether you are interested in reading through code to improve your understanding, finding an open source project to contribute to, collaborating on a project with a friend, or simply backing up and sharing your own projects online, GitHub is a great resource. As a code repository, it is probably the best place to look for examples of open source code that you can reuse in your own projects. But beyond this, GitHub offers a good way for programmers to maintain a portfolio of their work and interact with other programmers. *Access:* https://github.com/.

GitHub

• **Stack Overflow.** If you search for the answer to a computer programming problem in any search engine, odds are the answer will be found in Stack Overflow. The site offers a way for programmers to ask and answer questions, whether they are beginners or experts. Best of all, other users vote on the answers, so that over time, the best answers are promoted to a place of prominence on the site. It is an ideal resource for new programmers, particularly since many questions have already been answered. *Access:* http://stackoverflow.com/.

Notes

1. Alyson Shontell, "Mike Bloomberg Pledges To Learn To Code On Codecademy This Year," *Business Insider,* www.businessinsider.com/mike-bloomberg-pledges-to-learn-to-code-on-codecademy-this-year-2012-1.

2. For more information on this distinction, see Thomas A. Powell, *HTML & CSS: The Complete Reference, 5th ed.* (New York: McGraw-Hill, 2010), 47.

3. Google's Python Class, https://developers.google.com/edu/python/.

4. "About OCW," MIT OpenCourseWare, http://ocw.mit.edu/about/. ℵ