

Assignment – Web browsers, web servers, and web pages

Hi class

There are many options for front end development. RUBY is a nice clean and versatile language that is similar to Python but without the whitespace constraints. It also is object-oriented and is often used in web server applications using the Rails application. Python also offers webserver interfaces Django and Flask. Javascript offers Node.js which is increasingly popular for full stack development. This exercise will work with the most popular language for web development ...this is javascript. The coding studio below will lead you through several tutorials for javascript/CSS/HTML. You will then download a free web template for your resume and learn how to host it for free at GitHub pages. We won't build a full-fledged web server, but you will learn enough of the mechanics to adapt this exercise to a software landing site and/or web server in the future. When building a simple site, we can check how it behaves by simply opening index.html in a browser. With a more complex site that acts as a server, we would need to install a set of softwares called a server stack and open it via its URL. To build a web server, we would first typically build what is called a local server on your own machine so that we can interact with it in the same way a client would as we develop it. This requires a LAMP stack (Linux, Apache, MySQL, PHP) or XAMPP stack (for windows to be installed. Your files are put onto this local server (e.g. in Linux file system at /var/www/html/) and the site can be visited locally on your machine at the URL 127.0.0.1 For more information see this tutorial. <https://www.alphr.com/set-up-local-web-server/>

NOTE: all my starter codes are found at <https://github.com/gbabbitt/course-code-repo>

Another webserver in progress can be found here. This one uses JSmol viewers for protein structures. https://github.com/gbabbitt/MD_comparison_webserver

For this week's coding studio

1. Let us start with javascript for cats and learn how to open a console from a web browser and do very simple cat-like tasks. <http://jsforcats.com/>
2. Now lets go to a real sophisticated bioinformatics web server. <https://pbwww.che.sbg.ac.at/maestro/web>
3. Open and inspect the elements here as you did in the previous tutorial. Can you recognize the html that defines the layout and the javascript programs that are called from within the html. Note that you cannot access as much here as this is someone else's web site.

4. Go to the w3schools.com tutorials for HTML, CSS and javascript and refresh your skills so that you can read and index.html file for a website and understand how it functions. <https://www.w3schools.com/js/default.asp>
5. We will not write a website from scratch...ever! Why? It takes a lot of time. It also requires serious training in art and graphic design. There are plenty of templates created by web designers for us to use. Many are free. Go to the site below and find a free resume template to download. Be sure it is less than 25MB in size, so we can host it freely on GitHub
6. <https://templatemo.com/tag/resume>
7. Go to the index.html and open in both your web browser (i.e. Google chrome is nice) and an editor like Komodo or even one designed for webpages like MS Web Expression. You can now spend some time working to customize the template for your resume. Change text, delete html, css and javascript elements you don't need.
8. (optional) When you are ready, send your files to a GitHub repository. You will need to open a free account. This is visible to the world, so do not post anything that is 'owned' by another party. This is why a free template is best. Use a GPL 3.0 license for your repo. Send up your website files and under 'settings' you can activate GitHub pages and receive a url that you can use to see the final result and share with others. This can also be a good place to post examples of your own code and coding projects for prospective employers <https://pages.github.com/>
9. (optional) While you are here at RIT you can submit web pages to your account at banjo.rit.edu. A good program for file transfer and management is WinSCP. It can be found at <https://winscp.net/eng/index.php>. A Linux has many terminal based file managers and another GUI based one called 'konqueror'. With these programs you can move your files to a place to host them and you can also control permissions to control who can see them on the web as well.
10. Submit an index.html file along with all its support files and folders (i.e. css and js) to myCourses assignment dropbox as a .zip file. This is open ended and creative. Start with a template and edit it to make it look nice on your web browser. It need not utilize a web server, but let me know if it does need this. NOTE: this should only be if it is running code that is not part of the client-side web page. For example a JSmol viewer on your html page needs to run some java code on your computer when runs, so it will require a local server to be set up on your computer using LAMP or XAMPP as described above.
11. (advanced options) A new tool for js called model viewer can be used to easily embed protein data bank structures and other 3D images interactively. <https://modelviewer.dev/>

Also check out tools like ploty.js and d3.js for interactive web data dashboards

<https://plotly.com/javascript/>

<https://d3js.org/>