

Alexandria Brumbaugh  
Professor Joel Ross, LIS 549  
Final Project: Draft 1

## **Introduction**

For this final project, I will be taking a closer look at the webpage of my workplace: Francis Martin Library, a neighborhood branch of the New York Public Library system. The website is located at <https://www.nypl.org/locations/francis-martin>. All branches in the system have similar formatting for their webpages, although I'm unsure whether it is a template built in-house or outsourced to web development contractors.

## **Code Review**

To begin, the HTML for this page is invalid according to the W3C Validator. The bulk of the error messages are for the bad-value inclusion of spaces in the opening script element, e.g. `<script type="text/javascript" src="./Francis Martin Library _ The New York Public Library_files/NRJS-f32db5da6c60de6a3b9">`. This is clearly a problem stemming from the file name "Francis Martin Library \_ The New York Public Library\_files", which could ostensibly be fixed by changing the name of the source file. There is also an orphan "aria-labelledby" attribute which does not point to any element in the document. Lastly, there are a number of unnecessary attributes and roles: seven instances of the "type" attribute in JavaScript code, and an instance each of the "navigation" role for the "nav" element, the "main" role for the "main" element, and the "contentinfo" role for the "footer" element. The page succeeds in using HTML semantically. Every list is written using `<ul>`, and all links are `<a>` elements.

Only two elements (outside of the `<script>` element, which we have not yet covered) were foreign to me: the `<meta>` element and the `<g>` element. Having looked them up on MDN, I now understand that `<g>` functions as a container element for graphic data, including further code such as `<svg>` and `<path>` elements, and that `<meta>` functions as a container for various metadata properties (ie.g. "site\_name", "url", and "title") which could not be otherwise represented in HTML. Otherwise, the code is very familiar - especially in the prevalence of nested `<div>` elements. Here is where I'll bring up my largest problem with the code, which is either needless space between lines or insufficient/no space between lines. It took me easily half an hour after importing the site's HTML to VSCode to reformat it into something easily parsable by human eyes, leading me to believe the site's code is generated via a hosting platform such as Squarespace or Wix. Further backing this hypothesis is a complete lack of comments anywhere within the HTML, leaving the purpose of individual lines subject to purely contextual interpretation. Frankly, I would not be surprised if I am the first person to lay eyes on the raw HTML of the website in recent years.

Three CSS files are linked in, with one plainly titled "styles.css" and the other two possessing long alphanumeric strings as titles, ostensibly modularly. No frameworks are readily apparent, but it appears that they are using JavaScript. The elements are styled almost entirely through CSS classes (although some ids are used as well), usually utilizing inline styling. These classes are semantic, including "header-globalAlerts", "header-globalAlerts-box", and "header-globalAlerts-box-item" in an early nest of `<div>` elements. Class naming appears to follow an internally consistent vocabulary, using a format of "[class]\_[subclass]", such as in "location-info\_address" and

"location-info\_phone". The most common selectors used are at-rules such as "@font-face" and class/pseudo-class selectors, such as ".nypl-icon-play-large:before". The class selectors are as complex as necessitated by the semantic naming conventions of the classes in the HTML, which at times can lead the CSS to seem overly specific. I was unfamiliar with at-rules, but have since learned that they're mainly used to define specific rules for CSS behavior - in this case, downloading and using externally hosted fonts. The biggest other question for me was the "content" property, which had values such as "\e801" and "\e614". From what I can gather, using the "content" property in this manner pulls and displays icons from an external library which are then not subject to further changes to the CSS. This is a really intriguing property, which frankly seems slightly over my head at my current level of experience.

### **Accessibility Audit**

According to the WAVE evaluation tool, the page is largely accessible. The only error brought up is the orphan ARIA reference "aria-labelledby", also addressed above. There are five other alerts: one long alternative text ("Red outline illustration of four icons: a dollar sign, a medical bag, a milk carton with an apple, and a house"), three redundant links, and one "suspicious link text" (reading, funnily enough, "click to learn about accessibility at the Library"). The redundant links all lead to "https://www.nypl.org/spotlight/welcome-back", the home page of the NYPL, with differing text preceding (one informing patrons that we are no longer charging late fines, and another referencing the current mask use policy). Given that these are the only major issues, I'm confident in saying that the page is well accessible; each image has

an "alt" attribute giving a written description, and all headings and sections are structured semantically.

### **Responsiveness Evaluation**

The page is responsive between mobile devices and desktop. The header fits to span the width of whichever screen is being used, and elements which are inline on desktop change to being stacked on mobile (for instance, the lists of upcoming programs under "For Adults"). The NYPL site header, containing hotlinks to major pages such as "Research" and "Connect" are moved to a sidebar, easily accessible by clicking a hamburger icon in the upper-right corner. The page continues to be well designed and easily navigable no matter which device is being used.