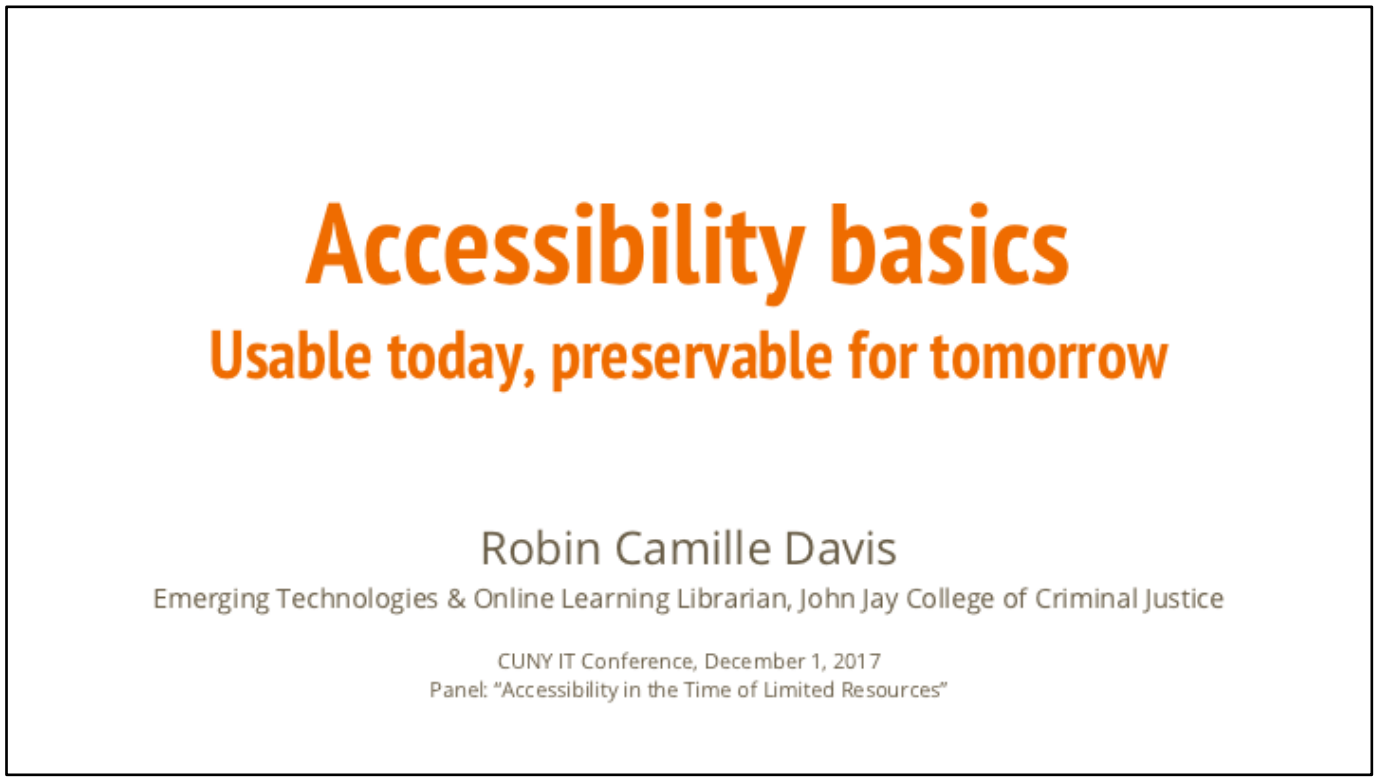


I'm Robin Camille Davis, I'm a librarian here at John Jay. I lead web projects in our library, including the main library website.

This opening presentation is a quick, practical review of accessibility. I want to emphasize that accessibility and usability go hand-in-hand. Designing sites to be accessible doesn't mean designing for edge cases or designing just for "disabled people." Designing an accessible website improves usability for every user. And that includes future users, too.

If you want to follow along, the URL at the top will take you to my Google Slides presentation. It will include my speaker notes as well.

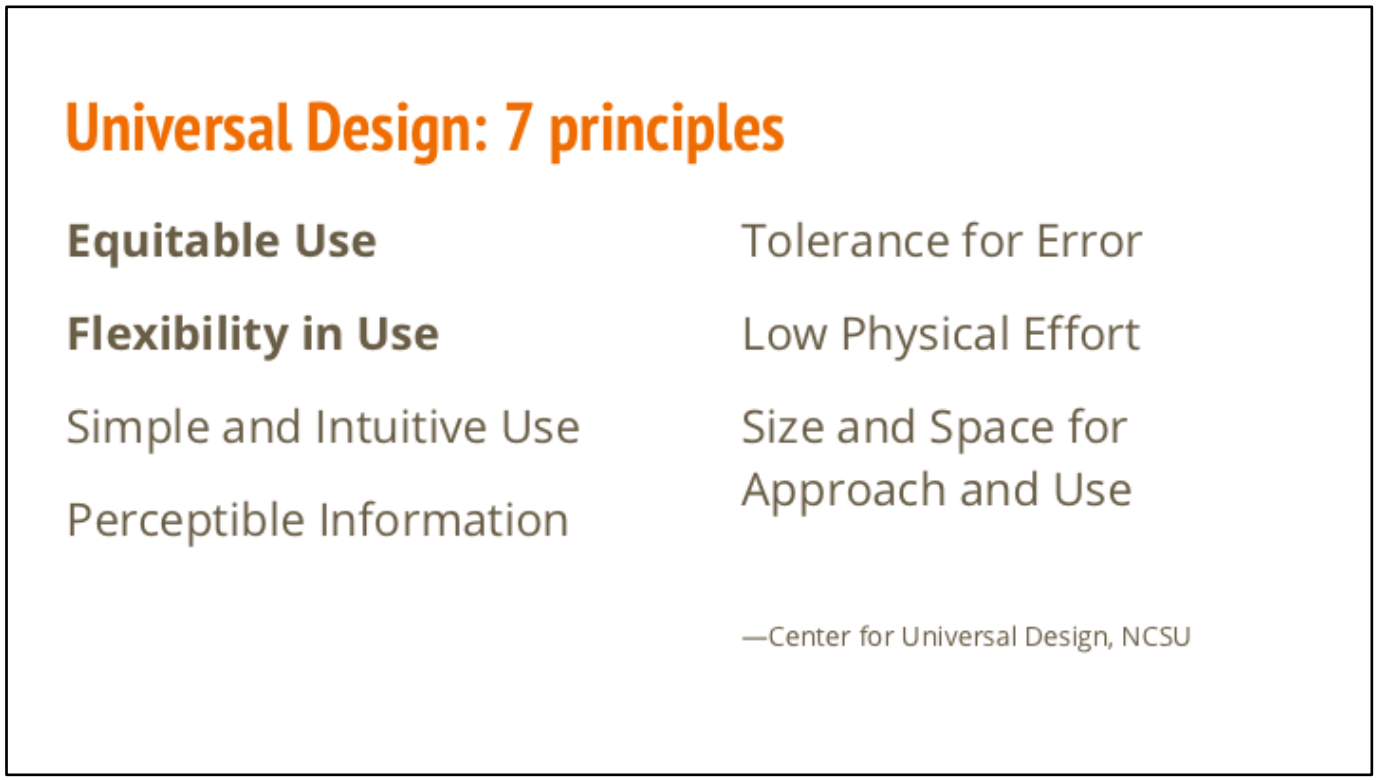
… By the way: these slides are not very accessible because I am using Google Slides, but after today, you can go to my website, robincamille.com, for an accessible version of these slides.

Universal Design

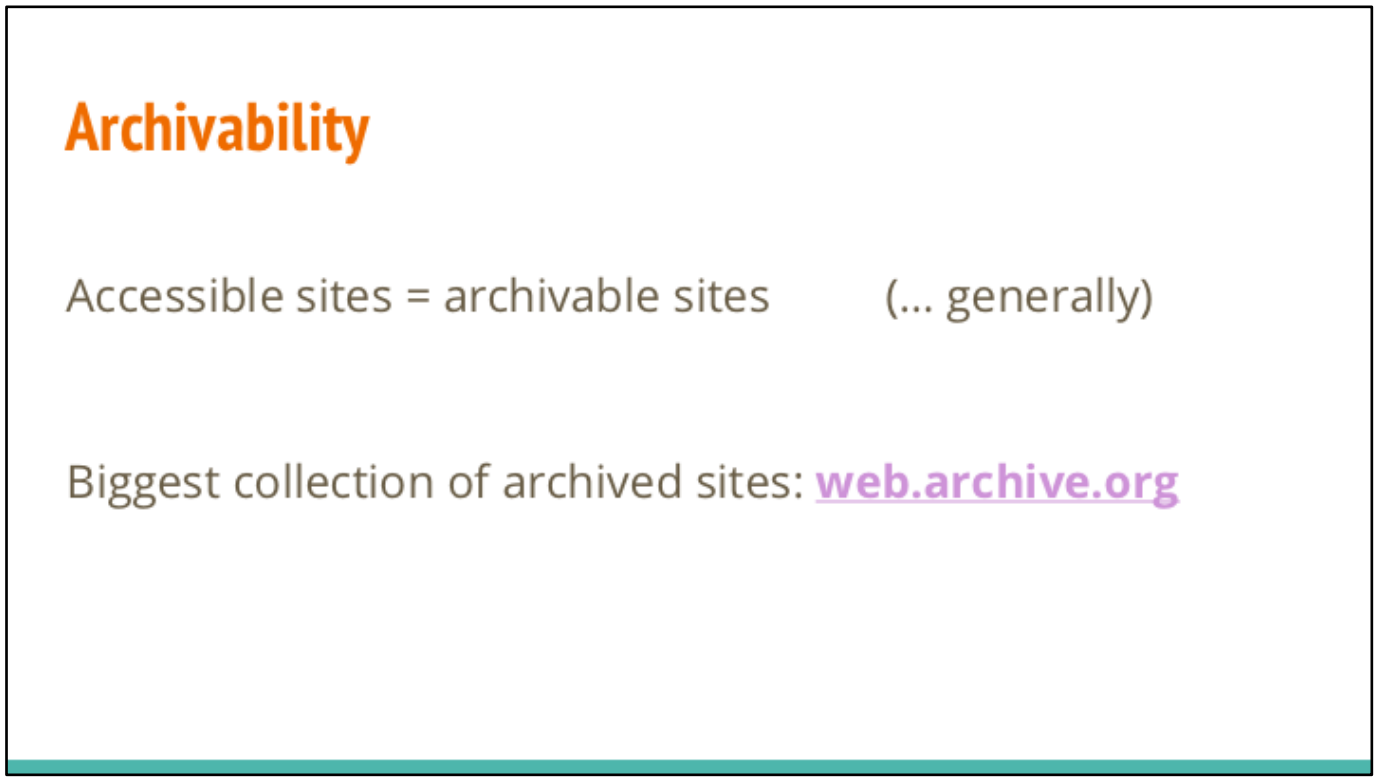
"The design of products and environments to be **usable by all people**, to the greatest extent possible, without the need for adaptation or specialized design."

—Center for Universal Design, NCSU

https://projects.ncsu.edu/ncsu/design/cud/about_ud/udprinciples.htm
Let me set the stage for talking about accessibility. I think about accessibility in terms of Universal Design. The overall concept is that …
So however your users are choosing to access your site — on their phones, with a screenreader, using only a keyboard and so on — your site should be equally accessible to all users. So accessibility isn't about designing for edge cases or disabled people, it's about making a great experience for everybody. We do all use the web differently, out of necessity, habit, or preference.



https://projects.ncsu.edu/ncsu/design/cud/about_ud/udprinciplestext.htm
For this presentation, let's keep in mind the first two principles, as outlined by the Center for Universal Design.
Equitable Use: "Provide the same means of use for all users: identical whenever possible; equivalent when not."
Flexibility in Use: "The design accommodates a wide range of individual preferences and abilities."
These two principles, together in the context of web usability, mean your website should serve up the same content whether they're using a screenreader, have super-slow internet, or happen to have a broken mouse that day and are relying on the keyboard. Users shouldn't have to work around your UI. They should be able to use your site the way they would expect to. And I'll show you some examples later in this presentation.



We should all strive to achieve universal accessibility on our web projects, just based on principle alone. We do want everybody to be able to use our site!

But there's another reason why we should aim for accessibility, and that's to make our websites archivable and preservable. It turns out that following accessibility guidelines makes your site more likely to be archived whole by the Internet Archive and similar organizations.

The Internet Archive uses open-source software to crawl and capture a website, bundling all the HTML, CSS, JavaScript, images, and other files together so that they can be viewed again in the future. Archived versions of sites have been used in court; they've been publicly available evidence that journalists sometimes point to; and for the everyday user, it's a great way to see what used to be at a broken link. Essentially, archived sites are part of the historical record.

And there's a good chance the websites you make are being archived, too. Moreover, there's a good chance that your future self will want to look back at the site you're working on as it is right now. So even if you don't really care about the historical record, think about doing your future self a favor.

It is a good rule of thumb that accessible websites are archivable websites, but I do want to mention that there are other factors that affect archivability, like free access to site content.

Some archiving issues

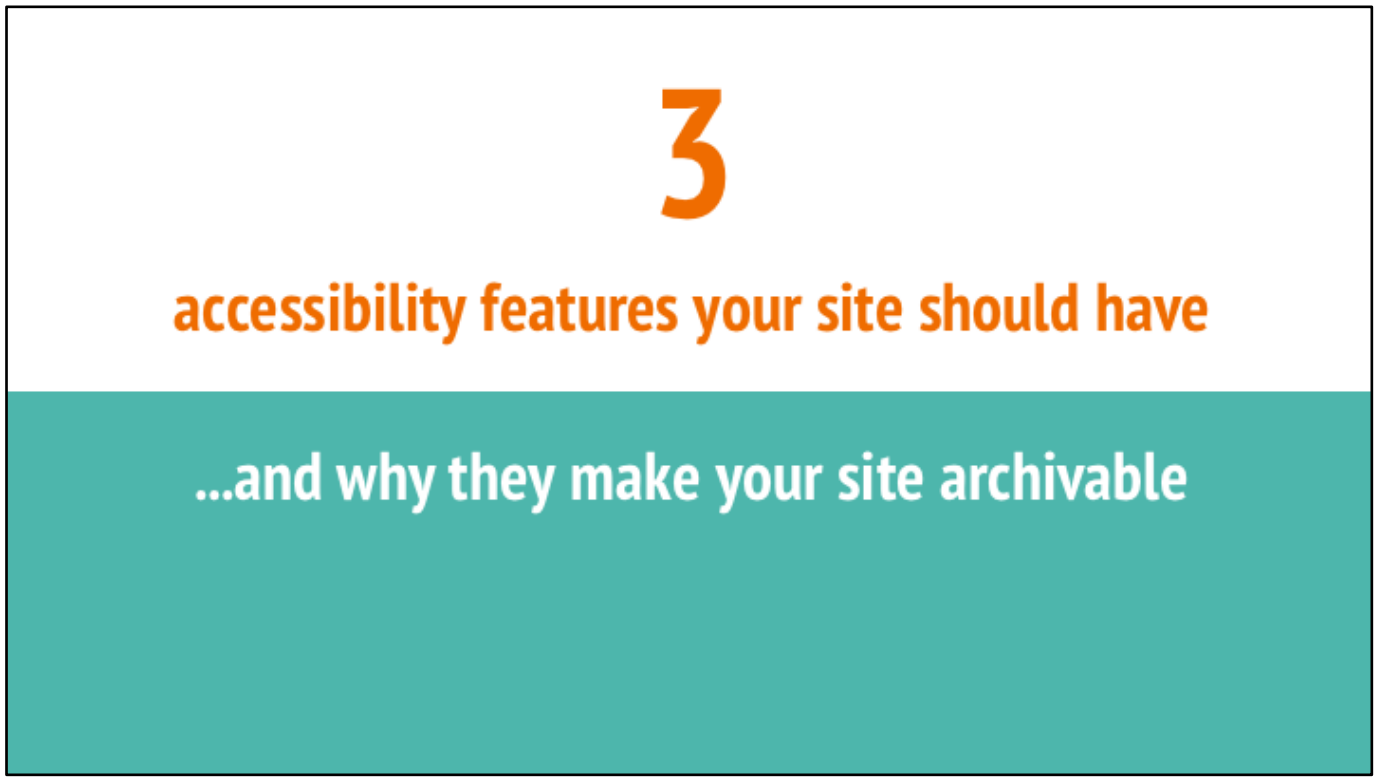
Page not archived (example, CUNY news article)

Wonky display due to JavaScript (example, JJ Library)

Image not captured (example, NYT front page)

Video not captured (example, NYT video)

Saving a website for the future is shockingly difficult! Websites now have so many elements — JavaScript libraries, videos embedded from other sites, and so on — and these are all hard to capture and "replay." Let's take a look at some examples of poorly-archived sites. I'll just show you two of these four examples, Wonky Display and Video Not Captured.



So let's discuss 3 concrete examples of accessibility features your site should have, and why those features make your site archivable as well.

Alt tags for images

Easy-peasy, except for forcing your entire department to do this

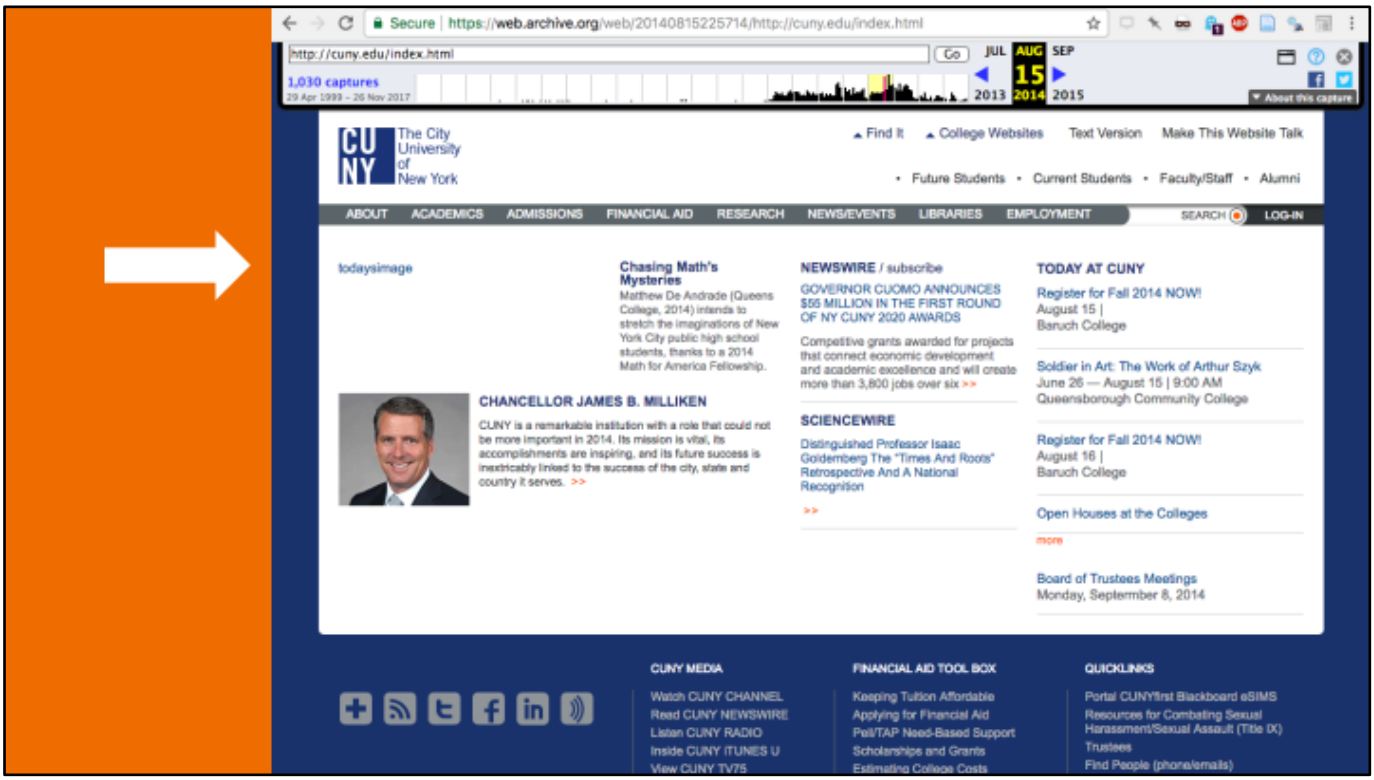
```
<img src="car.jpg" alt="Blue VW bug" />
```

The ALT tag should be included every time you use an image on the page (except for background images). It's a short textual description of what's in the image, so that if the image can't be displayed, you at least have a clue what should've been there. In my example, if the image of the car didn't load, you'd at least know that the image was of a blue VW bug.

In terms of accessibility, screenreaders can read the ALT description of an image, so users who are vision-impaired don't miss out on content. They get *equitable use* of the site.

ALT tags are something everyone always brings up when talking about accessibility, and it's such a simple piece of text, but the reason it's hard to implement is because it's usually an optional field when uploading an image to a CMS.

Turns out, ALT tags for images are important for web archiving as well. Capturing a webpage is tricky, and sometimes the software doesn't capture the image that should be there. Instead, just like when the image link is broken, it displays the ALT tag instead.
Here, the title of the page was an image, but whomever designed it wrote out what it said in the ALT tag, so we don't miss out too much. A+.



Here, whomever designed this page just gave a generic “todaysimage” ALT tag, all one word, to the main image on this page. Not very explanatory at all, and 3 years later, we future users don’t know what should have been there.
So ALT tags should provide equitable access to the site’s image content, for a variety of reasons. Screenreaders, display problems on live sites, and web archiving issues.

The second thing I want to touch on is navigation in two ways: in-page and site-wide.
First, let's talk about headings, like what I've demoed on the right of the screen.
This is super-basic HTML, and to be honest, for a long time, I never understood the reason to have these h1, h2, h3 headings because you could just use CSS to style what a subtitle should look like.
But heading tags are extremely useful for screenreaders, since users can quickly move between heading tags to skim the page without hearing every paragraph. And they're also useful for search engine optimization.

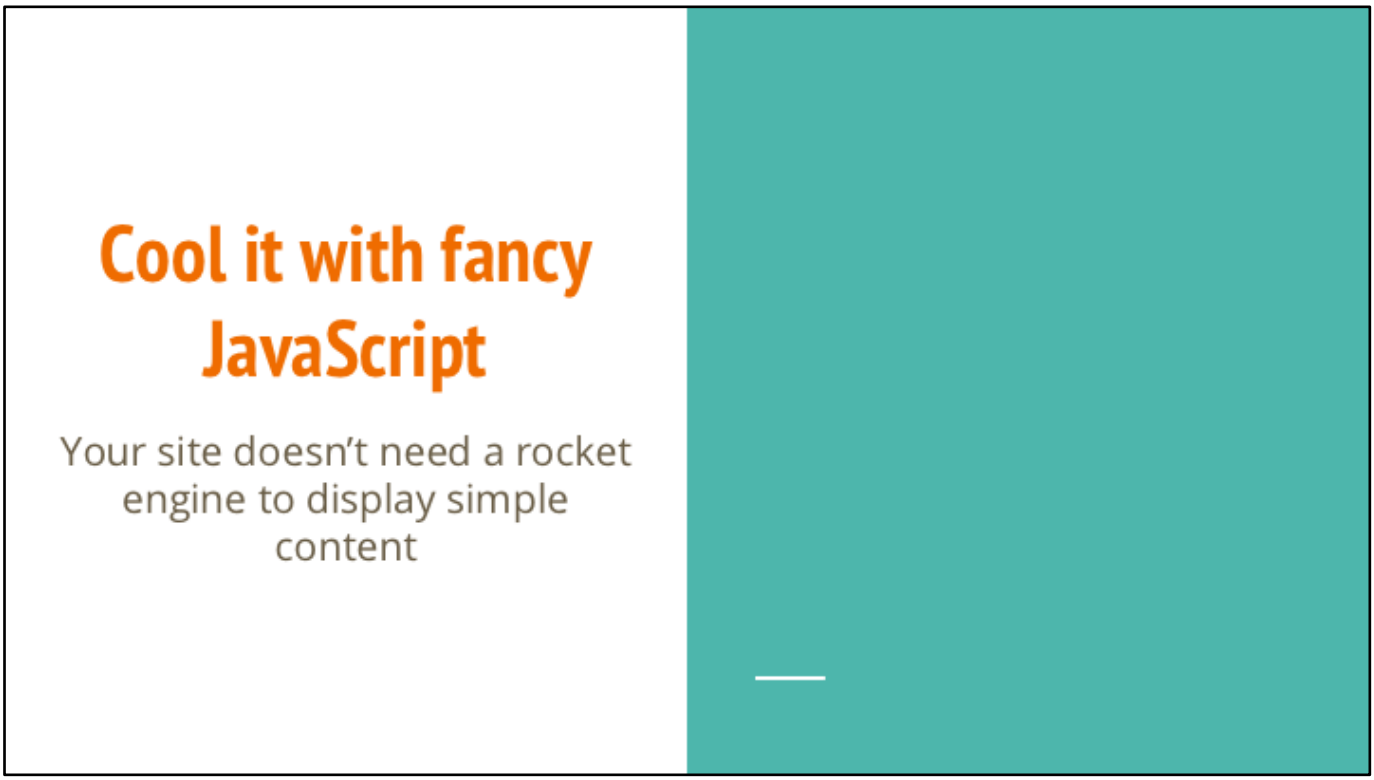
The other kind of navigation is site-wide. Your website should be organized like a tree: pages should have subpages. There should be a clear connection between your site content, and that should be reflected in your menus, like these menu options on the left from my library's homepage. Your URLs should also reflect this site organization, like the examples on the right.

Good site-wide navigation makes your site more usable for everybody. This is user experience design 101, yet it is hard to accomplish, especially if you're manging a legacy site or migrating a huge site to a new system.

Your navigation tree should be arranged in a dedicated sitemap page. That's where the list on the left comes from, actually. Remember the second principle of Universal Design? Flexible use. Sitemaps are useful for people who find your usual navigation scheme hard to use.

Sitemaps are also useful for web crawlers, which love to see a to-do list of pages to crawl. This makes it more likely that your whole site will be crawled by the Internet Archive.

A link to your sitemap should be in the footer of your page — that's where people expect to find it, and if it's on every page, that makes it more likely that a web crawler will find it, too.

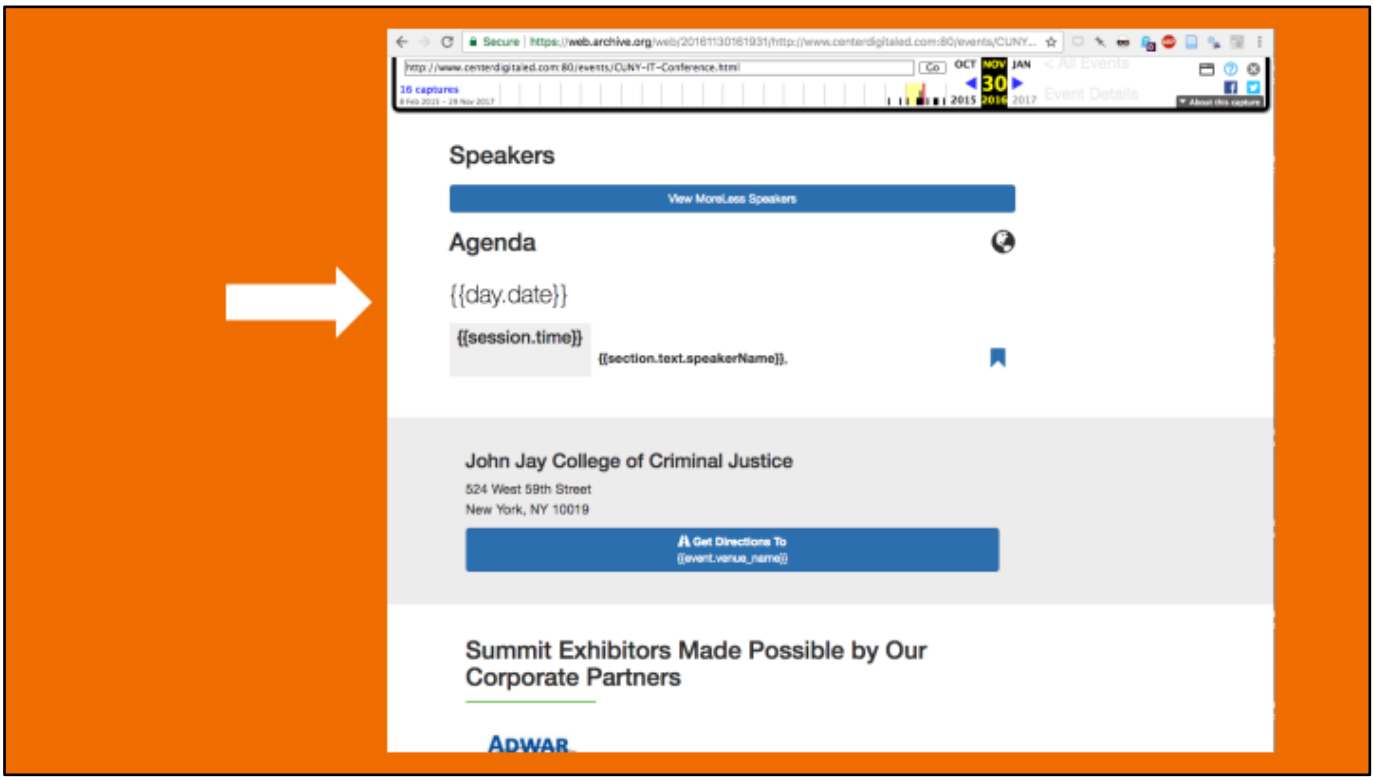


The last thing I'll talk about that makes your site accessible and archivable: cooling it with the fancy JavaScript.

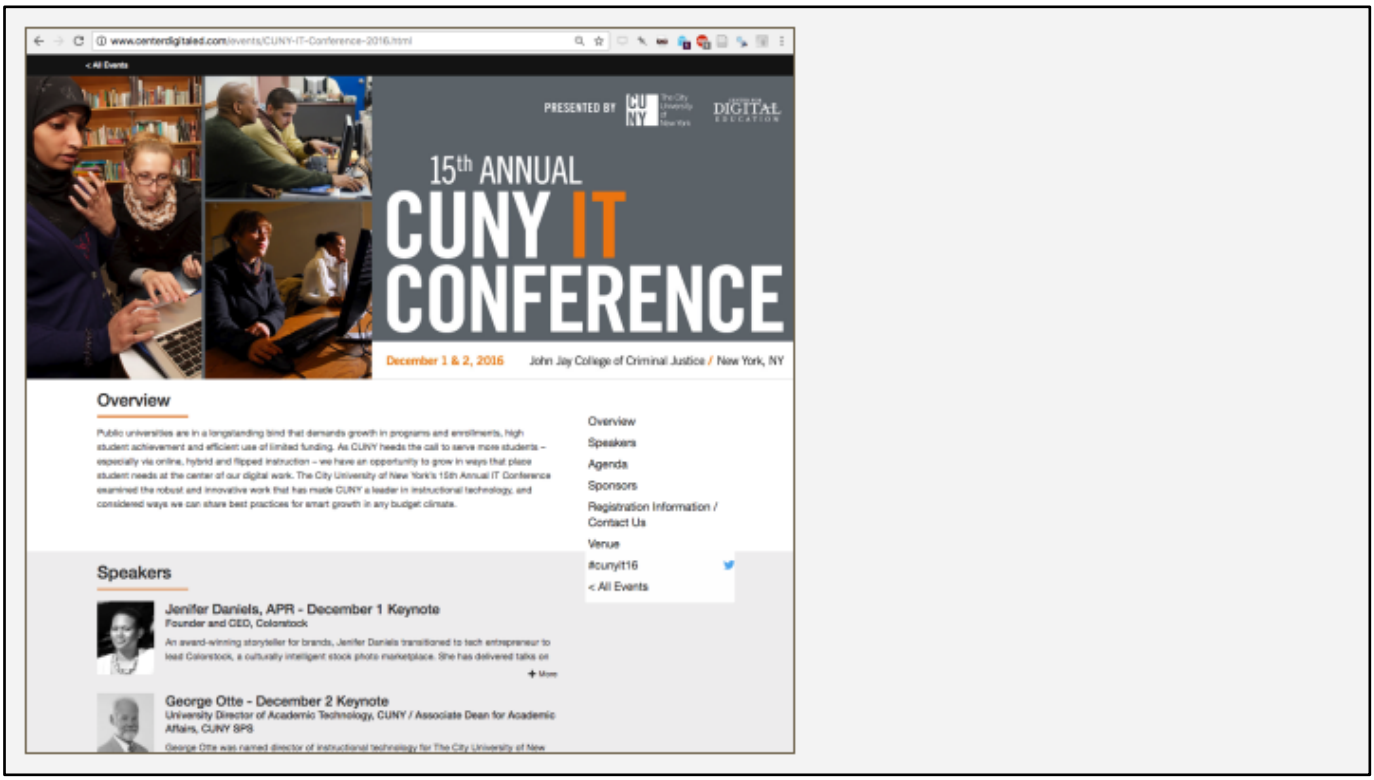
Some of the newer JavaScript frameworks might make your website really slick, they might add a lot of really cool features, but they do increase the risk of your page display being broken now and for the future. Using more mature standards and frameworks minimizes this risk.

This isn't always feasible. Sometimes the whole point of your webpage is to display dynamic content straight from a database. But if your site doesn't *have* to do that, it shouldn't.

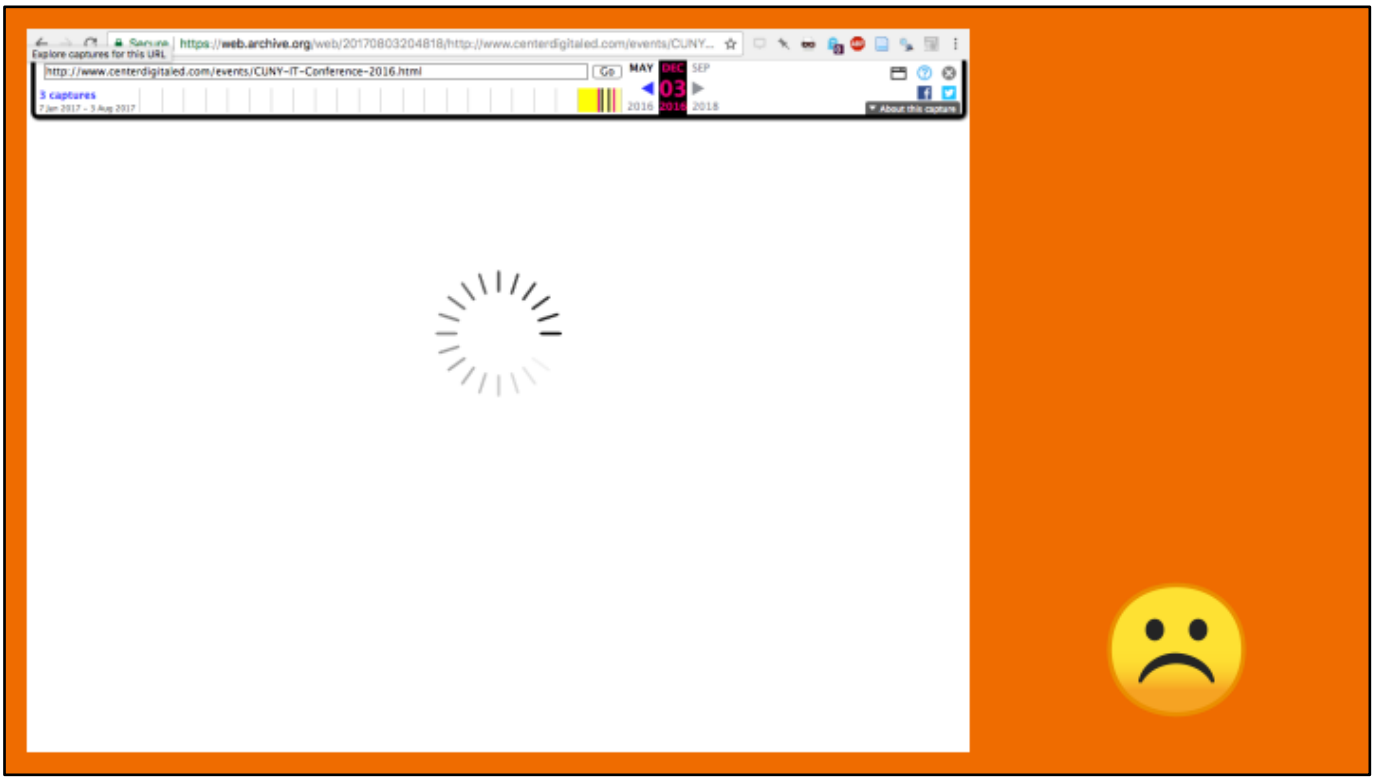
Let me give you a bad example to show you what I mean.



Say you plugged in the URL for the CUNY IT Conference to see what the schedule was for the 2016 conference. The Internet Archive has crawled it, but it looks like the dynamic content can't be filled in. It's probably missing the framework's back end that would have this information. That's too bad.



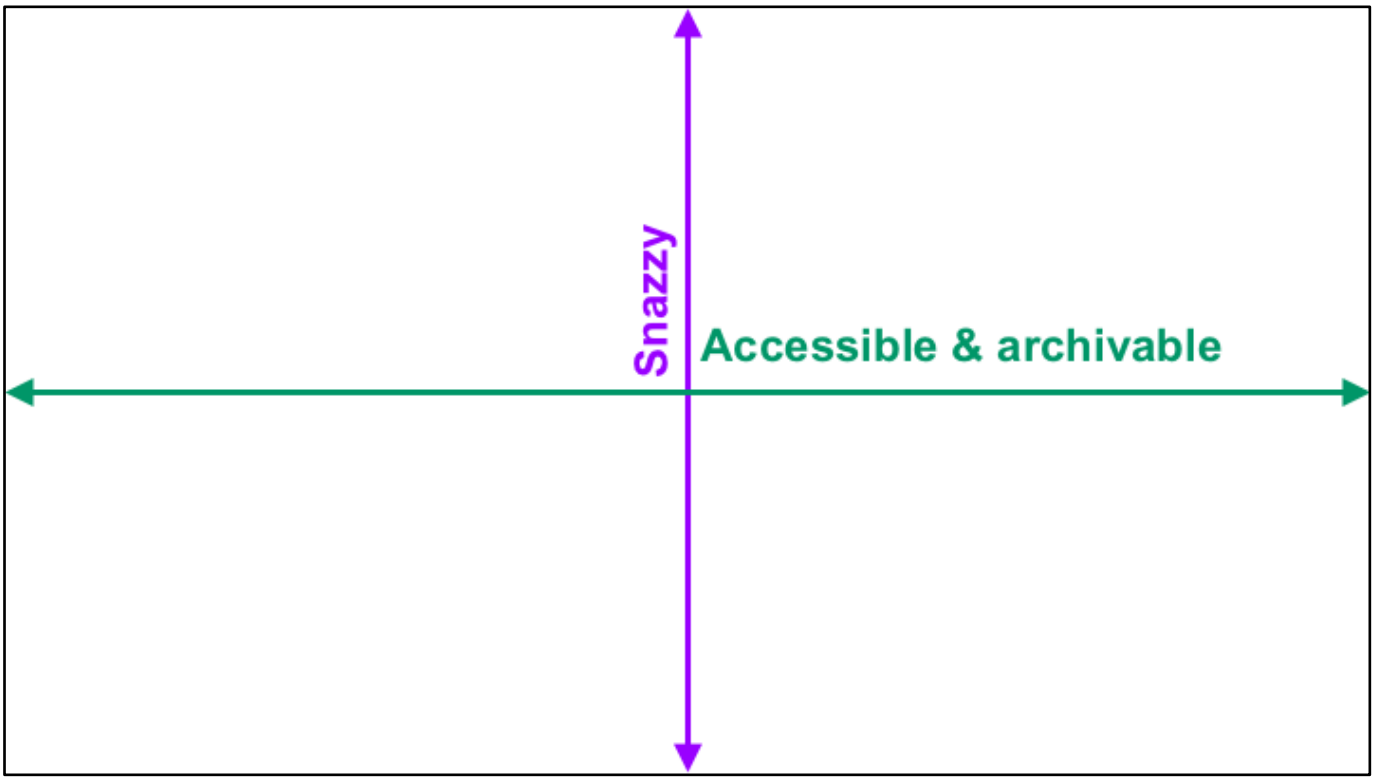
But wait!, you say. There is actually a special URL just for the 2016 CUNY IT Conference. Great! Let's see if the Internet Archive captured that one.



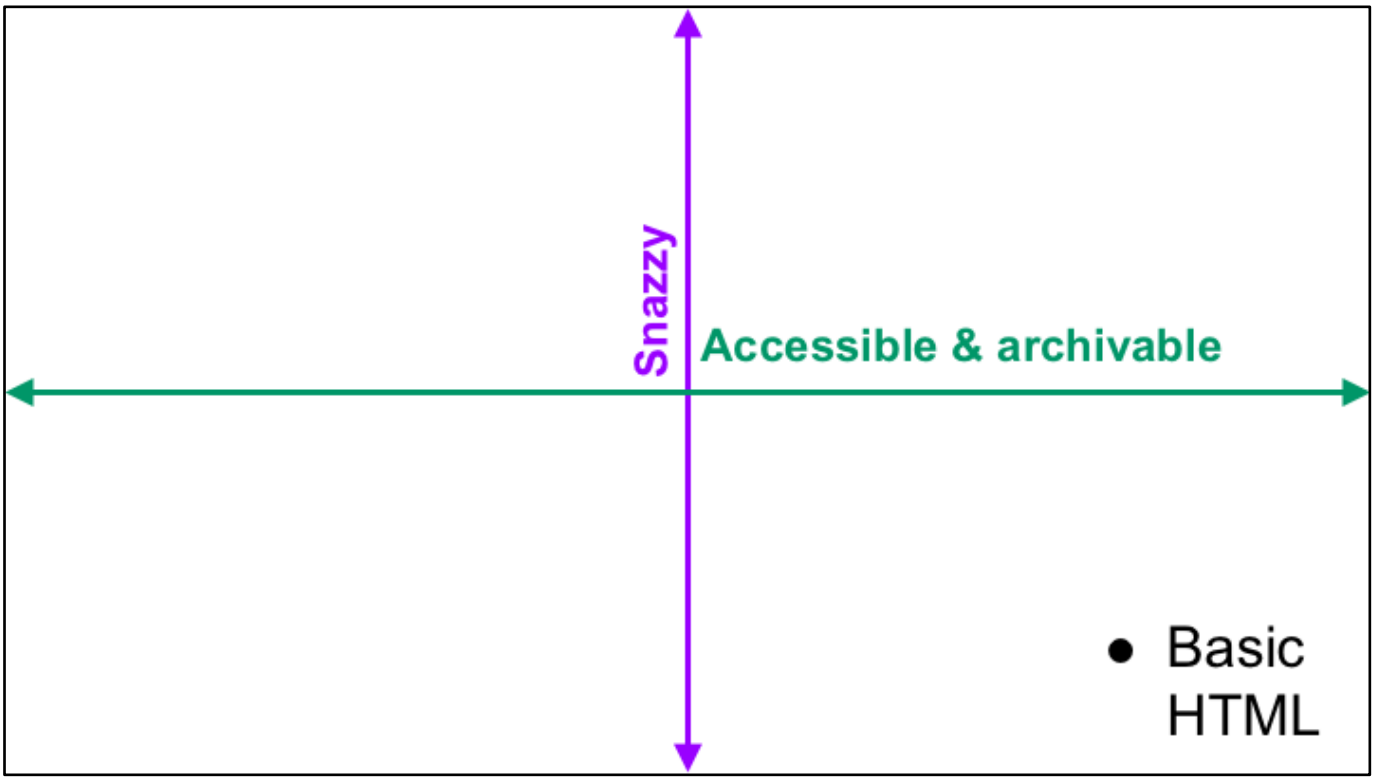
Sadly, you just get the spinning wheel of death, forever. You can't even get the text of the main page from the source code, because it looks just like the other page — lots of curly braces with no dynamic content.

So while the 2016 CUNY IT Conference webpage is still live on the web, it hasn't been archived by the Internet Archive, which is a shame. And this is just one example of many sites that have made themselves unarchivable. And that makes the information inaccessible.

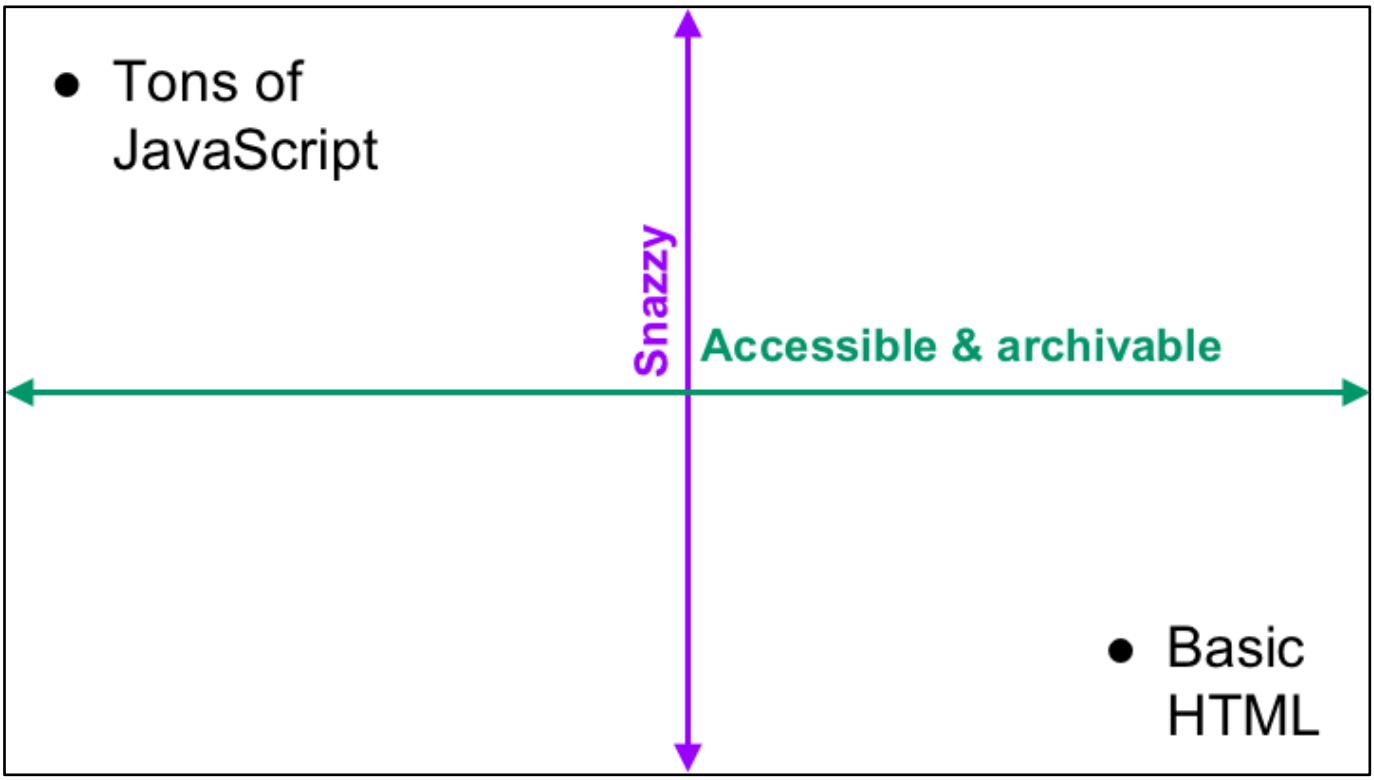
It is an easy fix — just saving the live page as a plain HTML file would surely sidestep whatever Angular JavaScript problems the page is having here.



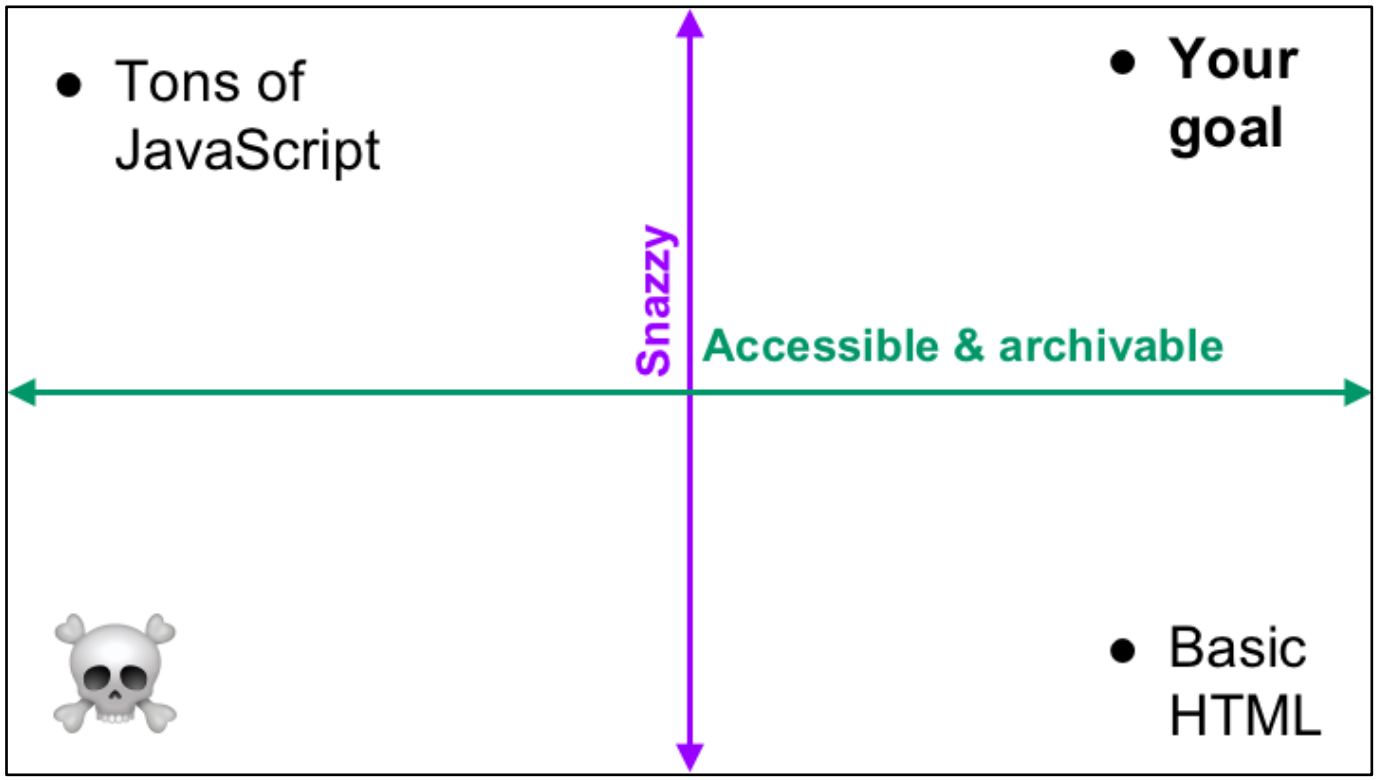
This touches on a balancing act: snazziness vs. accessibility. I consider this quite a lot when I design web experiences.
Creating universally designed websites can sometimes mean fighting against your CMS or navigating different directives for your project.



Extremely basic websites: ugly as heck to look at, but if the HTML is well-formed, hey, it's super accessible! And it's fast and looks the same on every device. The legal department will be really happy about this because you won't have a legal complaint about accessibility.



On the opposite end of both axes, a fancy-shmancy website that uses tons of JavaScript to make image slideshows and hover menus and cool fonts and some kind of smooth scrolling feature. It's a snazzy site! Your marketing office will be really happy to show it off. (I'm using 'marketing office' as a stand-in for whomever is convincing you to make a fancy-shmancy site. That might be yourself. I know I have an inner marketing office.)
Trouble is, a snazzy site is probably not accessible or archivable. It's slow, hard to navigate with a keyboard, it's not intuitive to use, and it might not be captured by an archiver.



Your goal is the perfect blend of snazzy and accessible. Something that looks “really nice” and serves the content equitably to users, regardless of how they’re exploring your site, and whether they’re users right now or users looking back at your site 5 years from now. This is a site that both marketing and legal are happy with. This is really hard to do.
If you somehow end up in the lower left quadrant, both the marketing office and legal office will kill you.

In sum:

- An accessible site is equitably, flexibly usable for everybody
- Follow basic accessibility guidelines, like:
  - Alt tags
  - Hierarchical navigation
  - Don't go nuts with fancy JavaScript if you don't have to
- Check your site on archive.org to see how it looks

Resources

Check sites for accessibility errors with WAVE and ChromeVox (screenreader plugin)

Mozilla Developer Network (MDN) docs

CUNY Assistive Technology Services (CATS) offers assistive technology training to faculty & staff

Five tips for designing preservable websites (blog post I wrote eons ago for the Smithsonian Archives)

—

Thank you! Robin Camille Davis • @robincamille • robdavis@jjay.cuny.edu

Images used: archive.org screenshots, Unsplash CC0 photo of car, Wikimedia emoji

Thank you!