lesson6.2 Go to file Code

This branch is 6 commits ahead, 4 commits behind master.

guyroutledge Update starter code ... on Feb 6, 2017 7

06/start	Update starter code	6 years ago
CSS_Animations_hando...	uploading handouts and updating readme	7 years ago
README.md	uploading handouts and updating readme	7 years ago

README.md

In this lesson, we will look at how we ensure the animations we create perform well, and don't break our sites in the wide variety of web browsers that exist.

While most modern browsers are great at handling transitions and animations, we can't always rely on people using the most up to date software. On top of that, animations can put pressure on the hardware that might not always be able to handle it. Thankfully, there are things we can do to try to ensure people who can't use our animations can still access our content, and approaches we can use to make sure our animations perform well.

If you've not used it before, Modernizr is a great and popular way to make sure we're making good use of browsers features. It's an open source project that inserts tests to your web page. And if the browser passes the tests, adds classes to the HTML tag of your source.

It can be use directly within JavaScript to detect if certain JavaScript features are available. In our example today though, we'll be using Modernizr to detect if our browser can handle CSS animations. And only present the animation code if we know the browser can make use of it. Before we look at bringing Modernizr in, let's see where we might need it in this example.

In the HTML, we have a set up that should look familiar. Just a demo container containing some text. We'll be animating that onto the scene. In our stylesheet, we can see a set up similar to the 3D animation we used earlier. If we open this example in our browsers, we will most likely see the box animate into place. Let's simulate how it might look if a browser did not support CSS animations by commenting out the animation line.

As we can now see the box is gone, it's being rotated back 90 degrees on the x axis. So it's basically invisible. It's relying on the animation property to make it visible. If a browser didn't support CSS animation, and we relied on the animation to show the content, this might cause problems for our visitors. Let's fix it, the first thing we're going to do is add Modernizr to our page.

There are a couple of ways to do this, but the most straightforward is to add the Modernizr code directly to the head of our documents. We want the Modernizr JavaScript to run before any CSS is applied to the page. Since Modernizr is a JavaScript tool, we could call it as an external script at the end of the document.

About

No description, website, or topics provided.

Readme
Activity
8 stars
7 watching
9 forks
Report repository

Releases

No releases published

Packages

No packages published

Contributors 2

spxds

donovanh Donovan Hutchinson

Languages

CSS 93.2% **HTML** 6.8%

This would defer processing of the JavaScript until after the CSS has rendered the document. While this is a good idea generally, it would mean there's a gap between when the CSS renders on the page and when our Modernizr enabled styles show later. So for now, we'll add it to the head of the documents.

First, go to modernizr.com. We're going to add a detect. Click the button, then in the list find "CSS animations". Pressing the + adds it to our custom build. We then press "Build" in the top right to finish up. And copy the Build version to our clipboard. Back in our HTML, we paste the contents of our clipboard into a script tag at the top of our page, before the closing head tag.

It looks like a lot, but all this is really doing is testing to see if the browser can support CSS animation. And then if so it will add the class CSS animations to our HTML tag. Let's see that in action. Nothing seems to have changed, but let's inspect the source.

In our inspector, we can see that the HTML tag, now has a class added. We can use this to ensure our animation CSS is only delivered to browsers that can support it. Let's update our CSS to make use of this. We begin by adding a new block, specifically for the CSS animations version of demo.

Any styles here will only apply if the browser supports CSS animation. We will put our animation and transform styles in here. One side note, we could have also added transform to our Modernizr builds to be completely sure that we catch every situation where either a transform or an animation might be needed.

Let's save this, and in the browser see the result. And here we have the animation showing. But before we move on, we can test that this is working by simulating what might happen if no CSS animations class is added to the HTML tag. Return to our HTML, and remove the script we added earlier.

We now see that the animation is not showing. But crucially, we're able to see the content. There's a lot more we can do with Modernizr, and it's a great tool that you might want to delve into. But hopefully this has made it clear how useful it can be in ensuring our visitors are not denied access to the content of our site.

Next, let's take a look at another potential browser issue, performance.