

10.2.2 **Using Chrome Developer Tools**

Robin has installed all of her tools and researched different HTML tags in preparation for her web-scraping project. She's really ready to jump in and start gathering data, but even with her initial research, she realized she wasn't quite ready to start scraping. The HTML components on the first site she visited quickly became more complex than she expected.

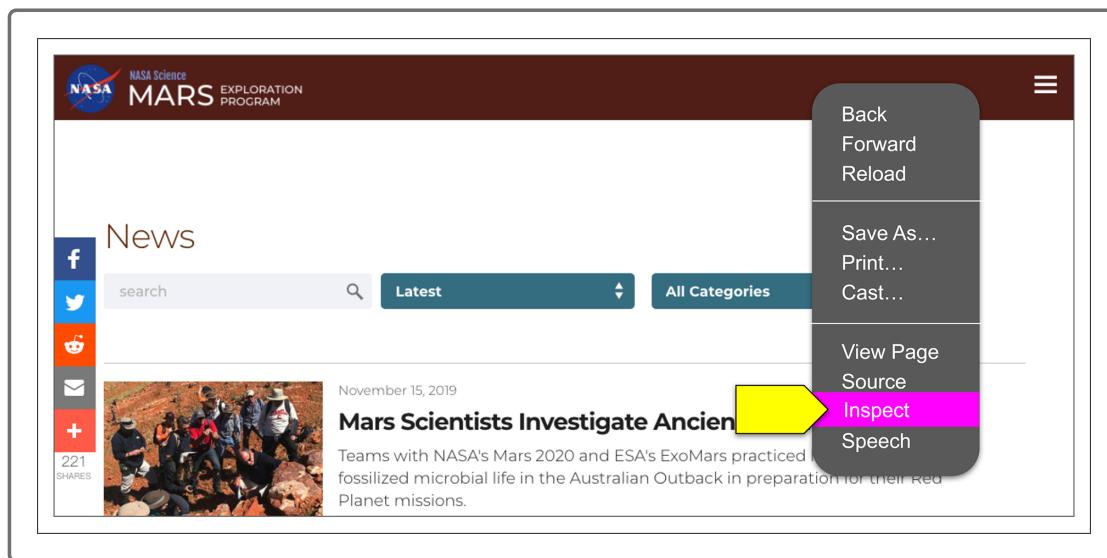
Instead, Robin has decided to practice identifying specific data using Chrome Developer Tools (also known as DevTools). This tool allows developers to look at the structure of any webpage. Not only that, but there's a search function as well. This should help make more sense of the tags and components that hold the data she's looking for.

Let's visit one of the websites Robin plans to use and take a peek at its structure, then practice finding different components.

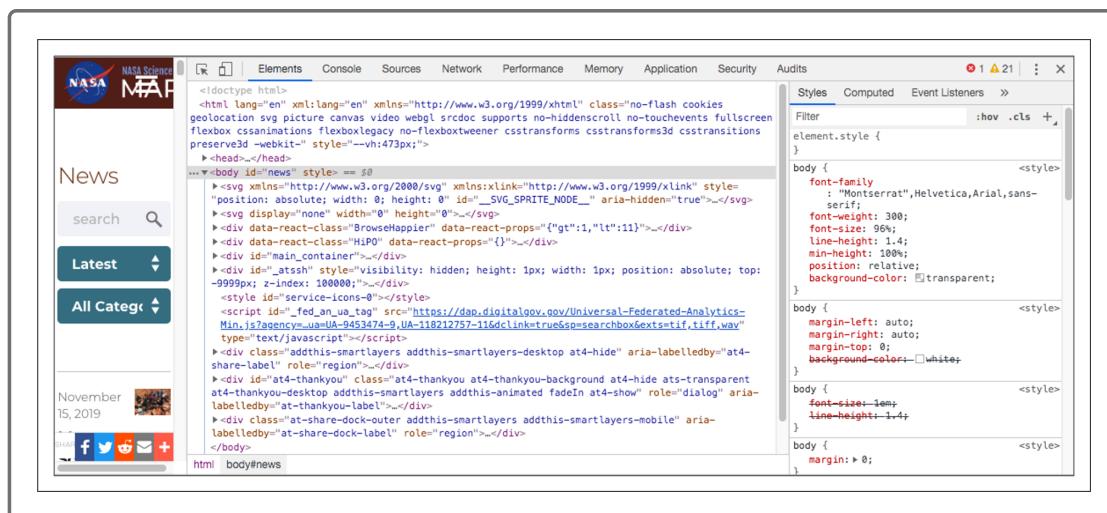
Robin wants to be kept up to date with different Mars news, and she's enjoyed the articles published on the [NASA news website](https://mars.nasa.gov/news/?page=0&per_page=40&order=publish_date+desc%2Ccreated_at+desc&search=&category=19%2C165%2C184%2C204&blank_scope=Latest). For her project

specifically she would like to extract the most recently published article's title and summary. Let's find the HTML components in the page so we can help her with that.

Start with opening the news site in a new browser window. At first glance, we can see that there are article titles and a quick sentence describing each article. Open the DevTools by right-clicking anywhere on the page, then click "Inspect" from the pop-up menu.



After clicking "Inspect," a new window should open under the webpage. This new window is docked to the webpage itself—it's part of the webpage, it's attached to the webpage, but it has a different job.

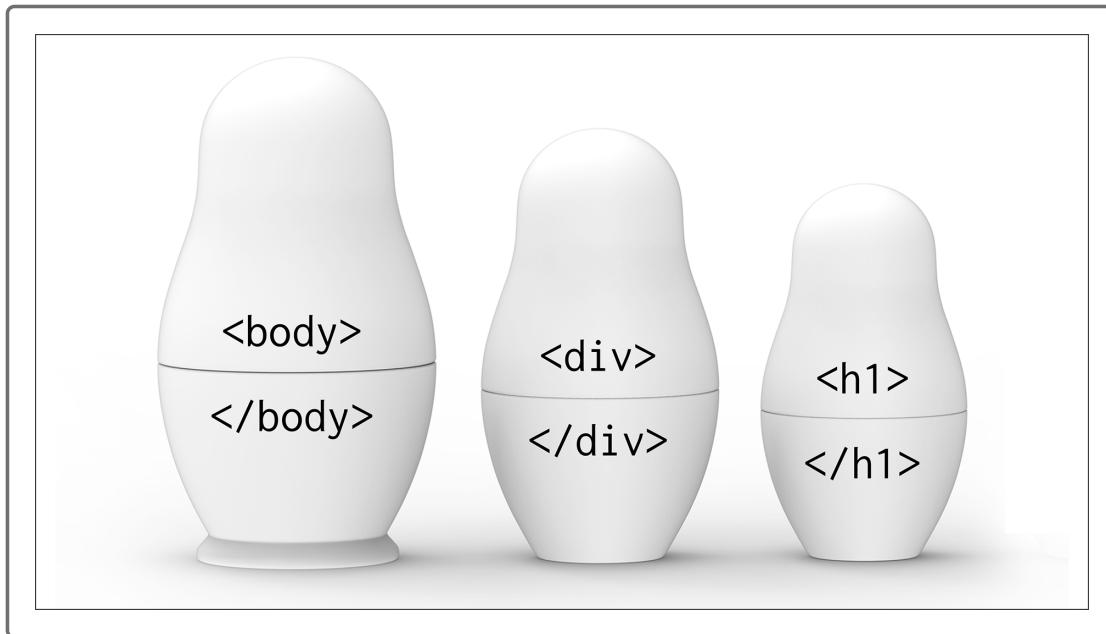


There is a lot going on in this site. What we're currently looking at is how this news site is assembled. The `<html lang="en" ...>` line should look familiar, as well as the `<head />` and `<body />` tags, but what is all of this other stuff? And the stuff inside the familiar tags? It's a good thing Robin wanted to take a deeper look at this webpage before trying to extract the data from it.

 [Retake](#)

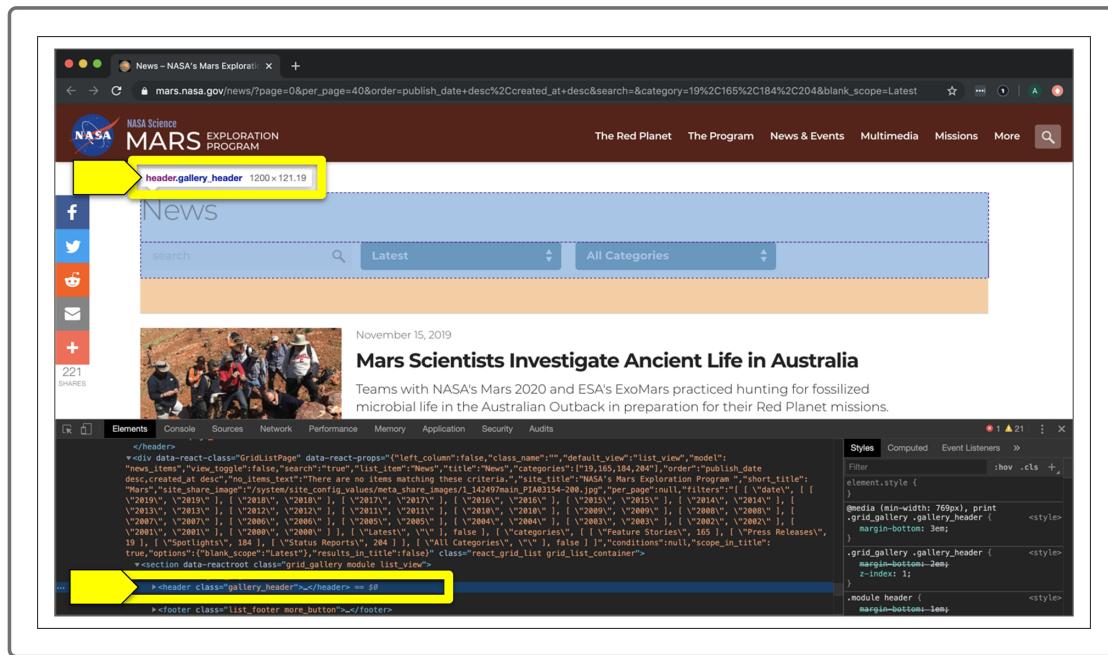
Let's start breaking this down a bit. Remember how we spoke about containers? For example, the `<body />` tag is a container for every visual component of a webpage, such as headers and paragraphs. Inside that `<body />` tag are other containers, which are nested much like a nesting doll. In the case of this website (and most websites), these other containers inside the body are `<div />` tags.

Take a look at the image below. Each container is nested within another. There can be multiple levels of nesting, depending on how elaborate the website is.



Another perk of the DevTools is that if you hover over the code displayed in the window below, the connected visual is highlighted in the page above at the same time. This is helpful because it shows us which code is specifically tied to features of the website above.

There is a lot of custom code included in this website, so instead of scrolling through all of it to find a certain element, we will search for it instead. In your DevTools, press "ctrl + f" or "command + f" to bring up the search function. Input "gallery_header" into the search bar then press enter. Make sure the line "header class="gallery_header" is selected, then hover over it with your mouse pointer. This will highlight the header section of the page: the title and its container element.



Hover over the next line of code, `<h2 class="module-title">News</h2>`.

If the header isn't displaying the nested contents, click the arrow beside it to expand it, and then hover over the line `<h2 class="module-title">News</h2>`.

Notice how the highlighted portion of the above site has become smaller? That's because we're now looking at an element that is nested inside of a container, instead of the full container.

This is a great way to pinpoint where on the website we want our web scraping code to pull data from. We can't just tell the code to grab a div or a header though, because there could be *many* of these on the website when we only want one. This is where the **class** and **id** attributes come into play.

HTML Classes and IDs

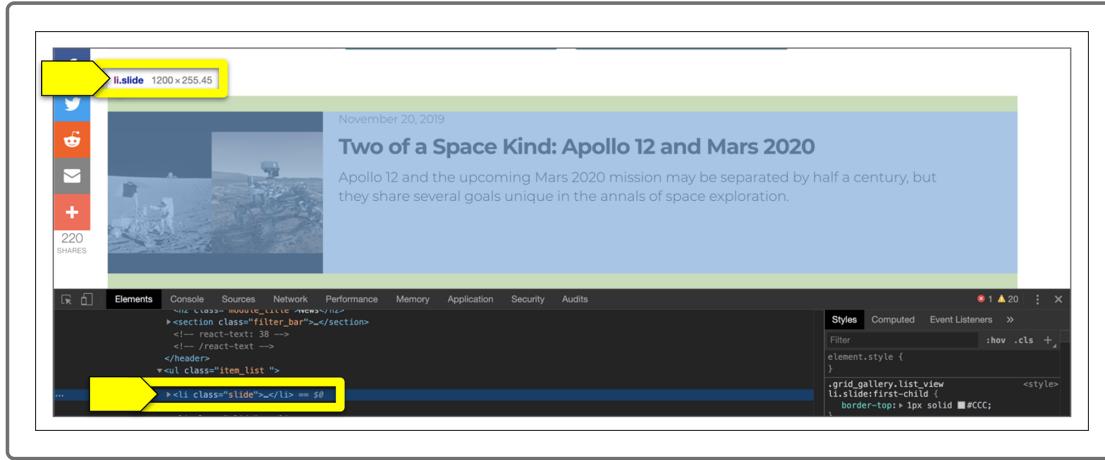
Because of how quickly HTML code can get bloated and confusing, it's important to keep specific containers unique. With everything contained within HTML code, it can be really difficult to find what we're looking for.

But how are developers able to distinguish one `<div />` from another? By adding attributes unique to each container or element. That's another reason to practice using DevTools. We can use it to search for these attributes. How exactly do they work?

Think of it like a litter of puppies. They all look pretty similar, but they each have a personality quirk or trait that makes them act a little differently from their siblings. By adding a different color collar to each puppy, we can now tell them apart just by looking. HTML class and id attributes are like those collars.

Robin knows that she will want to pull the top article and summary sentence. How do we identify those components, though? Let's look at our DevTools again. This time, let's drill further down into the nested components—we want to find the element that highlights only the top article on the page.

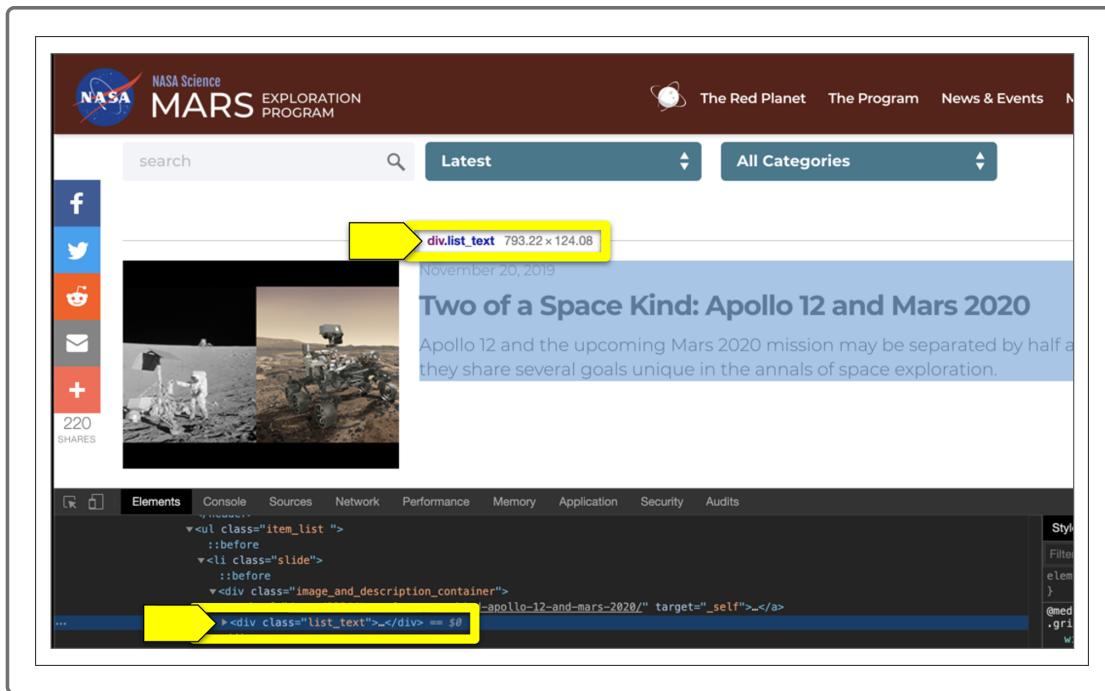
The first `` element with a class of "slide" highlights the top article on the page.



The section we're aiming for (the article title and text) is nested further in, and there are quite a few steps we'll need to take to get there.

First, click the drop-down arrow on the `<li class="slide">` element (if it isn't already open). From there, we're directed to another element: a div with the class of "image_and_description_container." Click that drop-down arrow as well. Within that, we have another element, `<div class="list_text">`.

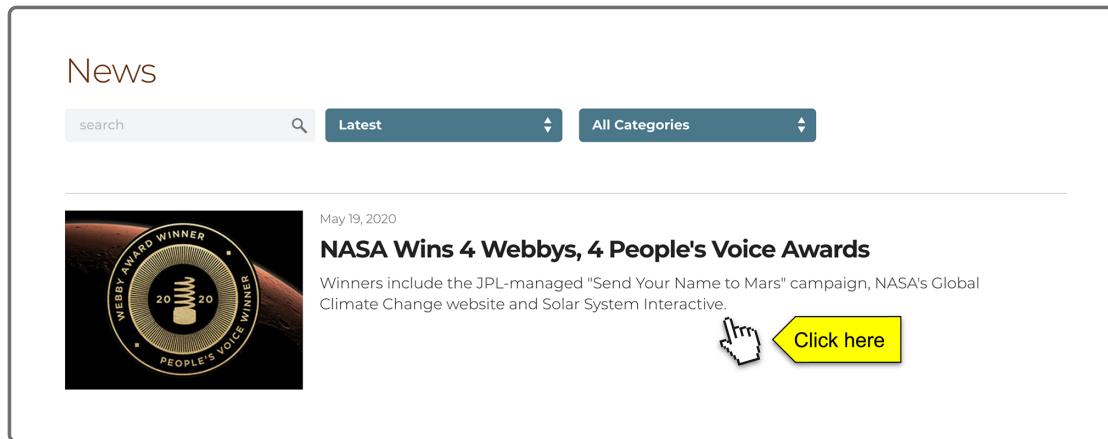
Maneuvering around these nested elements is called "drilling down," and it's a skill you'll encounter and employ fairly often as you continue to work with HTML.



This final container holds the information Robin will want: the article title and summary. With the use of DevTools, we've been able to sift through the nested HTML code to find the exact tags we'll need to reference in our scraping script. This process is something we'll be following with each additional webpage we want to scrape: visit the page, identify the data, then shift through the HTML code to pinpoint its location on the webpage.

That is a lot of clicking to get to one single section of a webpage. With our advanced technology is, isn't there a faster way? You bet! Let's condense the steps above.

Go ahead and close your dev tools window, then take another look at the webpage. Locate the first article's title and summary, and right-click the space below them. This time, click "inspect" from the pop-up menu.



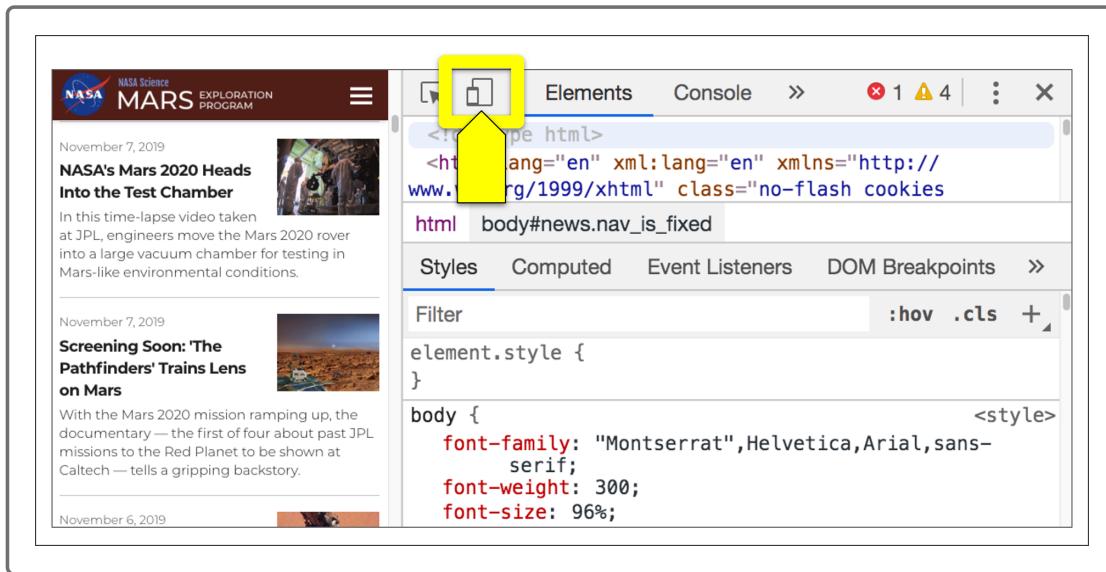
The dev tools window automatically opens again, but this time the highlighted section is already closer to the element you want to view, if it isn't already selected. You can tell by mousing over the highlighted element—it will simultaneously highlight the corresponding location on the webpage.

Using this method will reduce your time spent sifting through the different elements on the page.

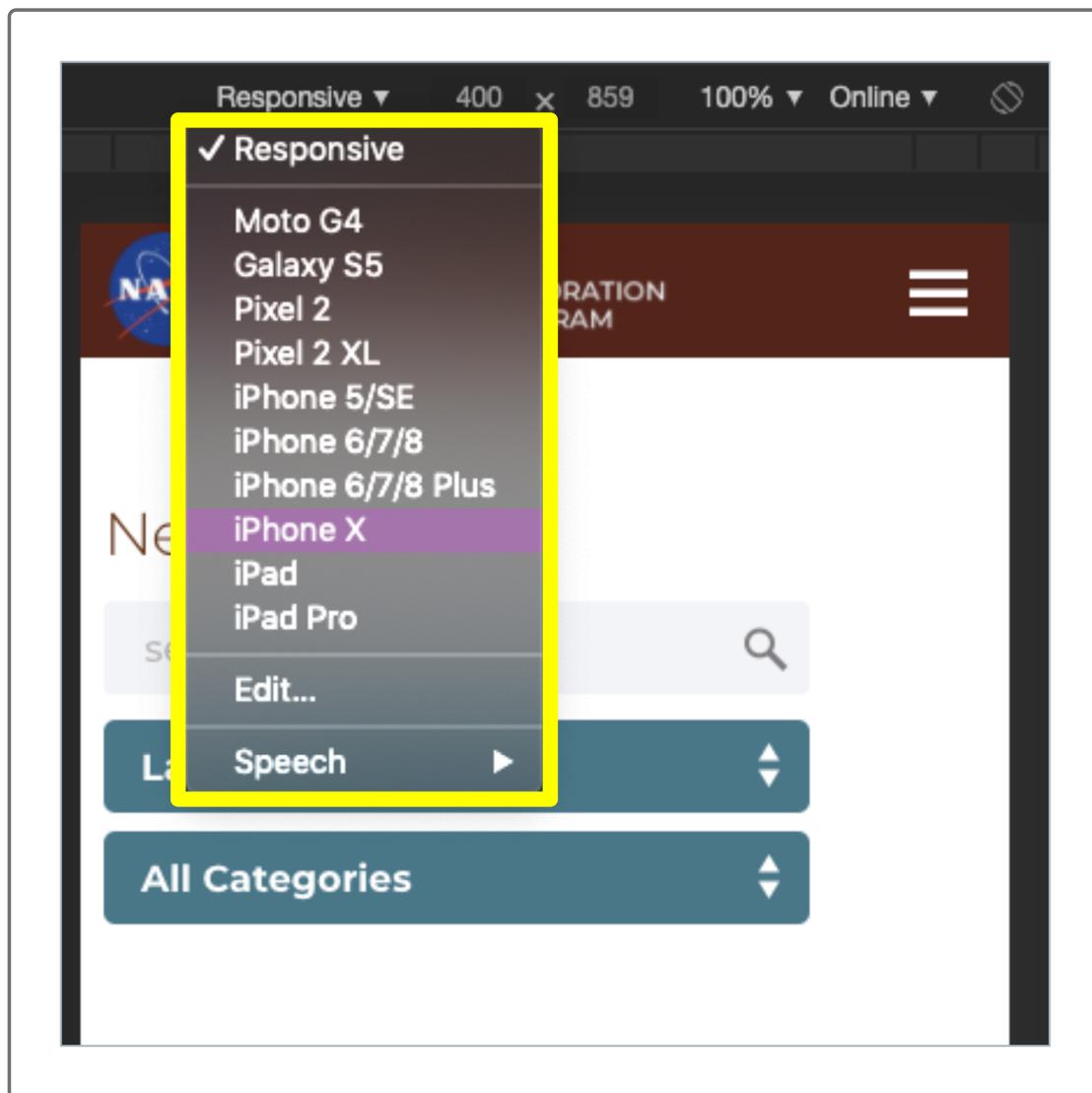
 [Retake](#)

Mobile Device Preview

DevTools also comes with a feature that allows us to view webpages as we would if using a phone or tablet. Not only that, but there are specific device models we can use to test the page. Let's look at the DevTools again—this time at the Device icon.



This button toggles the device selector. When clicked, the webpage we're viewing automatically adjusts to the height and width of a responsive mobile device. When in mobile mode, there is a drop-down menu at the top left of the screen; this menu provides a selection of devices to choose from and to view the site with.



Switching between devices alters the webpage to reflect how it interacts with each device.

When you're ready to view the webpage as it normally is shown on your computer, you can toggle the responsive view with the same button (the device icon).

We'll use this later in the module when we build a portfolio.