TODOs

Not blockers – some will be used for JS post.

- give specific examples for use of each feature
    - ~~alignment issue~~
        - ~~icons & buttons in nav bar~~
    - ~~button with hover state~~
    - removing a broken modal
    - color not being applied to links
    - take existing nav bar and make it horizontal
- ~~answer the question of “what is X useful for?” throughout~~
- look up popular HTML/CSS questions on Stack Overflow
- ~~transfer repo/Pages site to @onemonth~~
    - ~~https://github.com/afeld/frontend-debugging~~
    - ~~replace afeld.github.io URLs~~
- have all links to exercise use the same `target` attribute
- ~~find tool to show clicks & keystrokes~~

Open questions

- does this post need to be trimmed down?
- does the example site need a nicer / more One Month-y design? note that the simpler the site [markup] the clearer the examples will be.

Things left out

- Styles panel
    - search box
    - "Animations Controls"
- computed properties

Introduction

This is the first in a series of blog posts on debugging, starting off with troubleshooting HTML and CSS. This guide will show debugging strategies in Chrome, though similar tools exist in all major browsers if there's another you prefer. While the official Chrome Developer Tools documentation ("**DevTools**" for short) does a good job of explaining the features, we will be walking through how each feature can be used in the context of a realistic frontend problem. Note that I'll be dealing with super simple static HTML pages, but the same HTML/CSS debugging approaches apply if you were using Angular, Rails, or any other framework to do the rendering.

Let's suppose you run into an issue where your page isn't rendering the way you would expect it to. What do you do? Well, you could simply add and delete and modify the markup in your HTML file, refreshing each time to see if you get lucky and it finally appears like you expect. This is a slow, frustrating, yet all-too-common way to troubleshoot your site, because there's a disconnect between making the changes and seeing what those changes do. Thankfully, there's a better way!

A few notes about this guide:

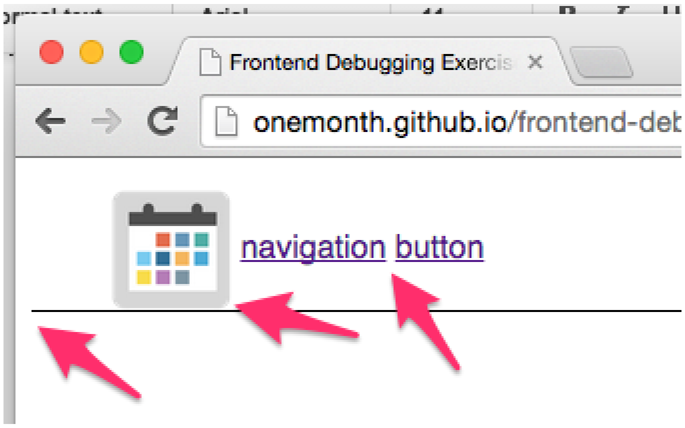
- This guide assumes you are familiar with HTML and CSS. Need to brush up on your skills? Check out One Month HTML!
- This guide was created using Chrome 45.0.2454.15 for Mac, though it should look/act nearly the same on other operating systems.
- All Chrome-specific terminology is in **bold**.
- All of the examples are available at onemonth.github.io/frontend-debugging/button/; you are encouraged to open them in a new tab and follow along.

Getting started with DevTools

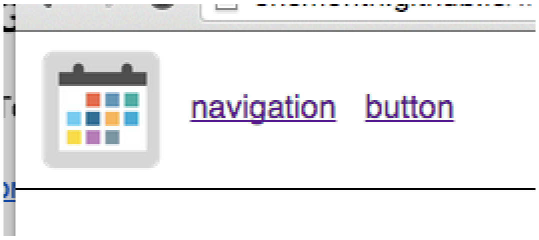
To start, open up this page in a new tab to follow along:

onemonth.github.io/frontend-debugging/navigation/

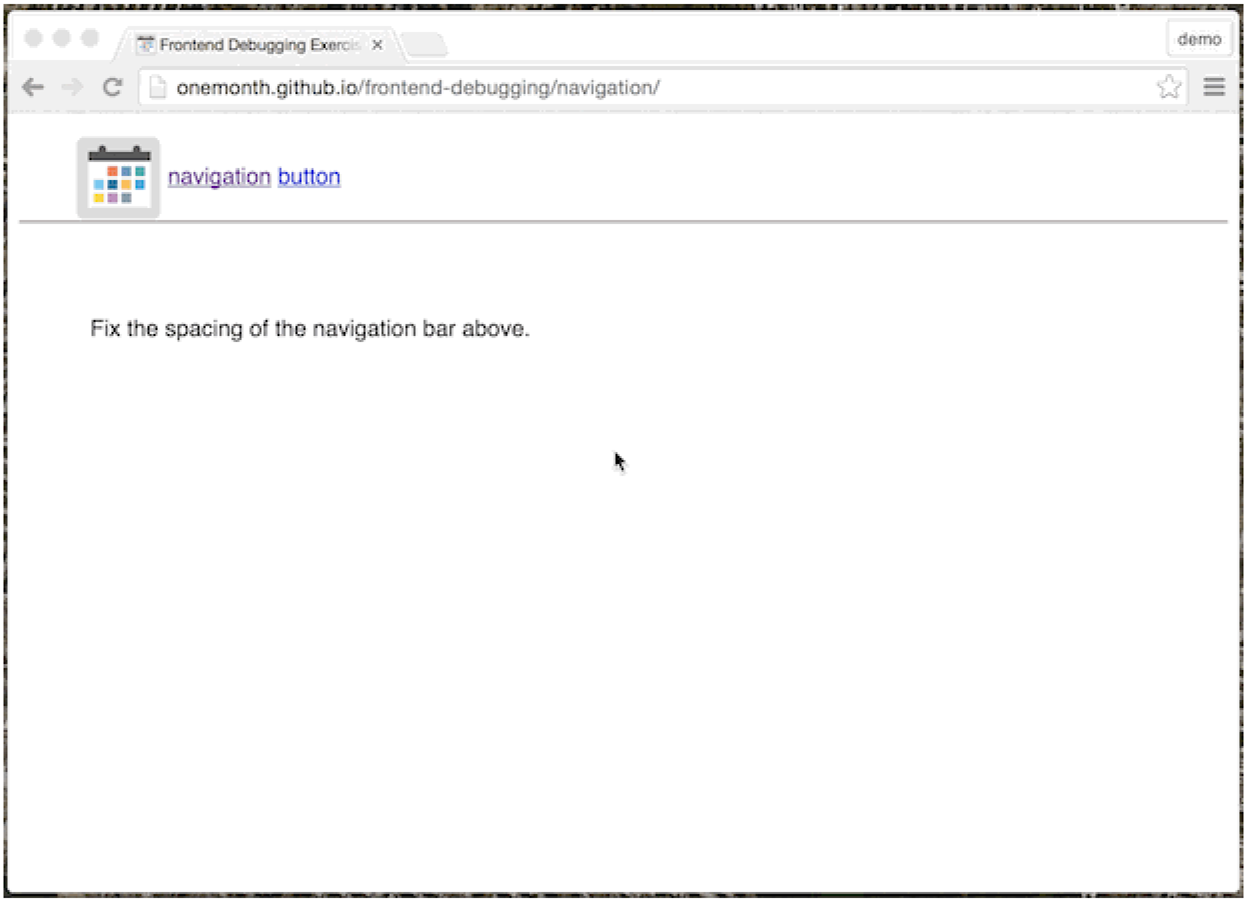
Notice the alignment issues in the navigation bar:



We want to get it looking like this:



What would you do next? Rather than take guesses about what needs to be changed, let's use in-browser tools to diagnose what's causing the issue. Right-click on some item in the navigation bar, then click **Inspect Element** in the drop-down menu.


Frontend Debugging Exerci
demo
onemonth.github.io/frontend-debugging/navigation/
navigation button

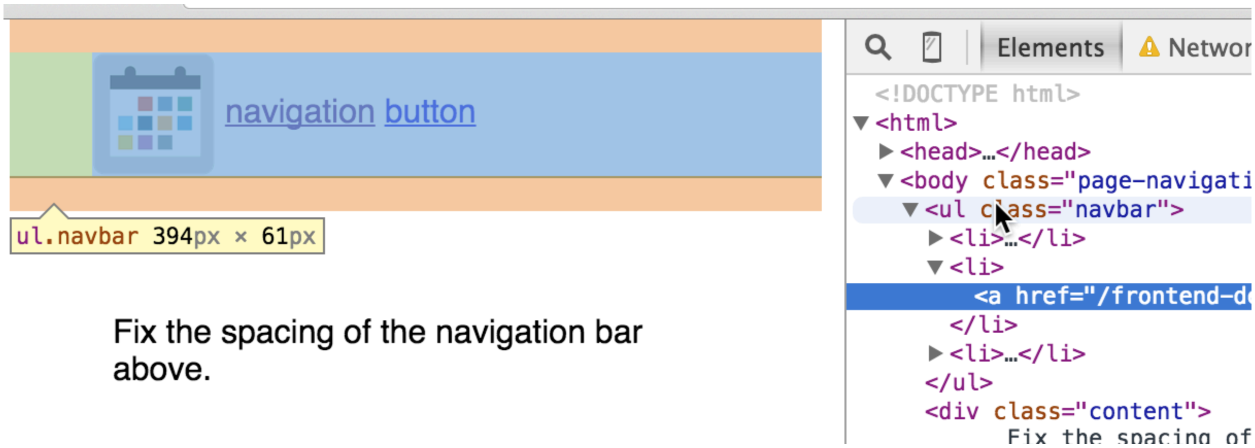
Fix the spacing of the navigation bar above.

You should now see the **Developer Tools** pop open on the bottom or right of the window, with the associated HTML element highlighted. Your DevTools panel may show up in a different position on the screen – you can change this by holding down the button at the top right of the DevTools and selecting one of the other options.

See also: alternative means of [Accessing the DevTools](#).

Exploring the DOM

The **Elements** tab of DevTools shows your markup as Chrome interprets it, allowing you to click the triangles to the left of elements to expand or collapse them. Navigate the `<ul>` element. As you do, notice that whatever element you have hover over in DevTools is highlighted on the page itself.



Fix the spacing of the navigation bar above.

```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
  ▼<body class="page-navigati
    ▼<ul class="navbar">
      ▶<li>…</li>
      ▼<li>
          <a href="/frontend-d
        </li>
      ▶<li>…</li>
      </ul>
      <div class="content">
          Fix the spacing of
```

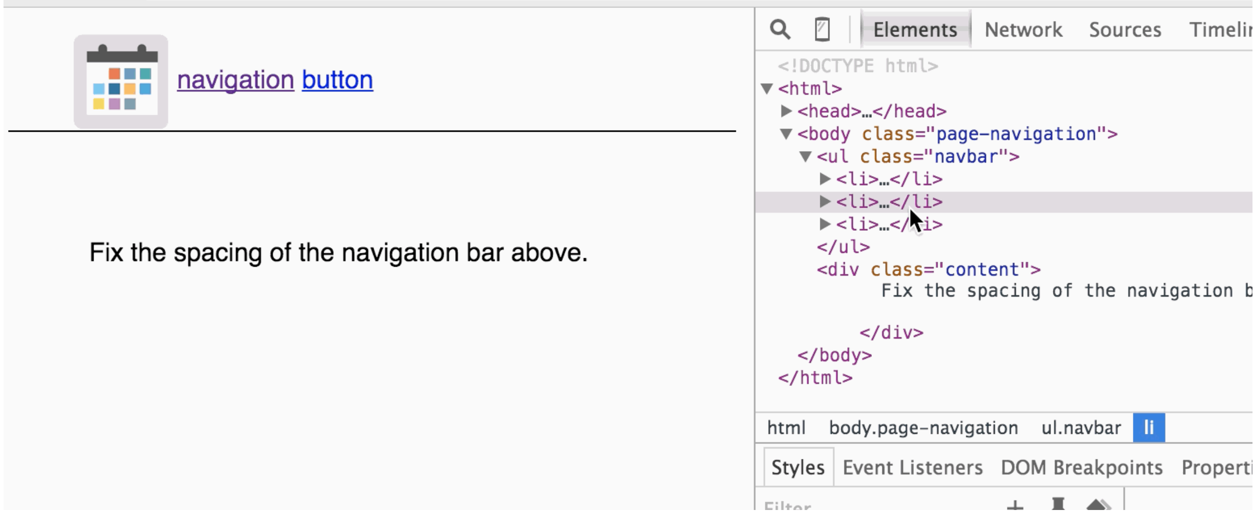
This is useful for seeing the boundaries of elements as they actually exist on the page, as well as the padding and margins. Notice that the CSS “path” to the element also appears at the bottom of the Elements panel, which is useful if you need to figure out what selector to use for a particular element.

If you want to explore what elements exist on the page, toggle the 🔍 icon at the top left of the DevTools, and start moving your mouse around the page. You can then click on that highlighted element to bring it up in DevTools.

Manipulating the DOM

Now that we’ve explored the elements on the page, let’s use DevTools as a place to try out different solutions. The Elements panel allows you to modify your HTML/CSS and see those changes reflected real-time, without needing to go back-and-forth to your editor. Another advantage is that the changes are low-stakes, in that you can easily revert back to the saved version.

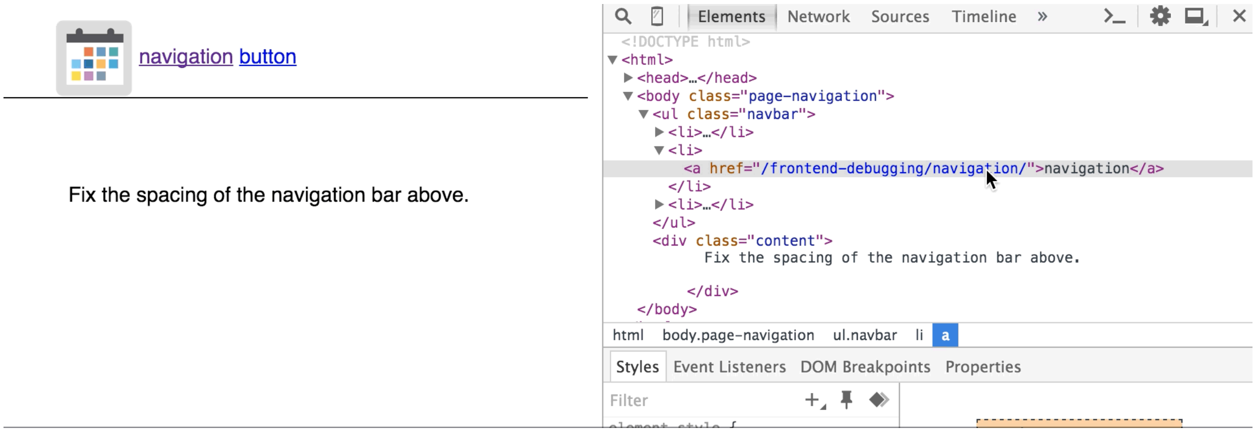
The first way we can manipulate elements is to reorder them. Click and drag one of the elements in the Elements panel up or down, and when you drop it somewhere, the change will appear on the rendered page. Try using this to reorder the items in the navigation bar.



Fix the spacing of the navigation bar above.

```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
  ▼<body class="page-navigation">
    ▼<ul class="navbar">
      ▶<li>…</li>
      ▶<li>…</li>
      ▶<li>…</li>
     </ul>
     <div class="content">
           Fix the spacing of the navigation b

         </div>
   </body>
 </html>
```

Suppose we also wanted to see how the navigation bar looks title-cased (Like This). Try double-clicking the text in the Elements panel to edit it. When you click away, the changes will be applied.



Fix the spacing of the navigation bar above.

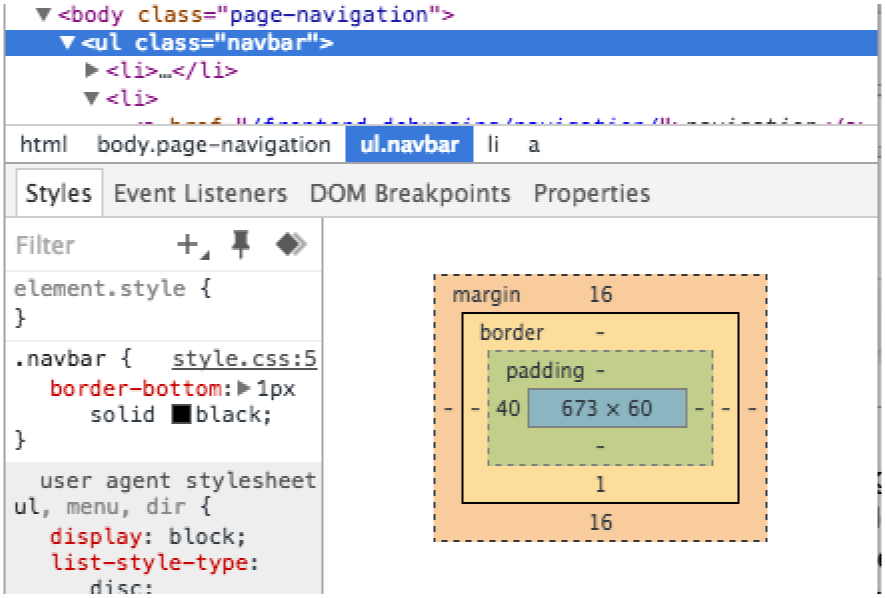
```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
  ▼<body class="page-navigation">
    ▼<ul class="navbar">
      ▶<li>…</li>
      ▼<li>
          <a href="/frontend-debugging/navigation/">navigation</a>
        </li>
      ▶<li>…</li>
      </ul>
      <div class="content">
            Fix the spacing of the navigation bar above.

          </div>
    </body>
```

Note that this is also useful for tweaking HTML attributes, adding/removing classes, etc. The good news is: if you want to start over, just refresh the page! The downside is that changes in DevTools aren't "sticky" – you throw away your changes any time you refresh or navigate to a different page. DevTools isn't intended to be a replacement for your editor – it's a way to try things out quickly.

Style inspector

Click on the navigation bar `<ul>` in the Elements panel again, and take a look at the **Styles** panel (which will be to the right of or below Elements). This is the part of the DevTools where you will spend most of your time when doing styling/layout.

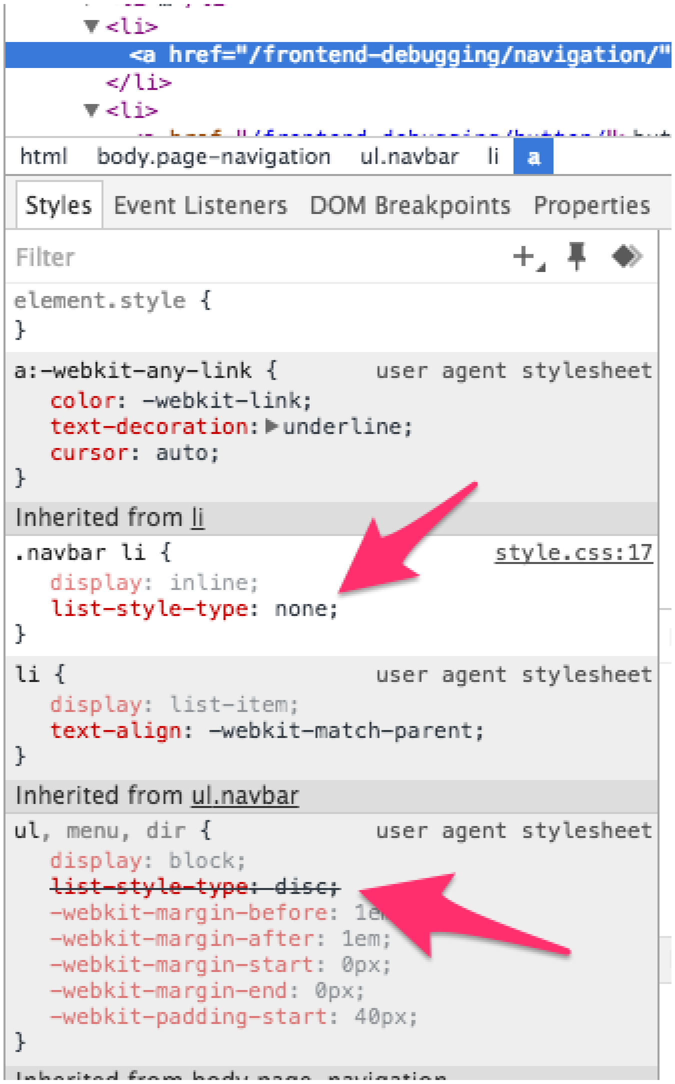
The **Metrics** pane shows the “box model” for the highlighted element.



```
▼<body class="page-navigation">
  ▼<ul class="navbar">
    ▶<li>…</li>
    ▼<li>
```

```
element.style {
}
.navbar {    style.css:5
   border-bottom: ▶1px
      solid ■black;
}
  user agent stylesheet
ul, menu, dir {
   display: block;
   list-style-type:
      disc;
```

Nearby, you will see the list of styles being applied to the element, in descending order of “priority”. In other words, each style rules overrides all the rules displayed beneath.

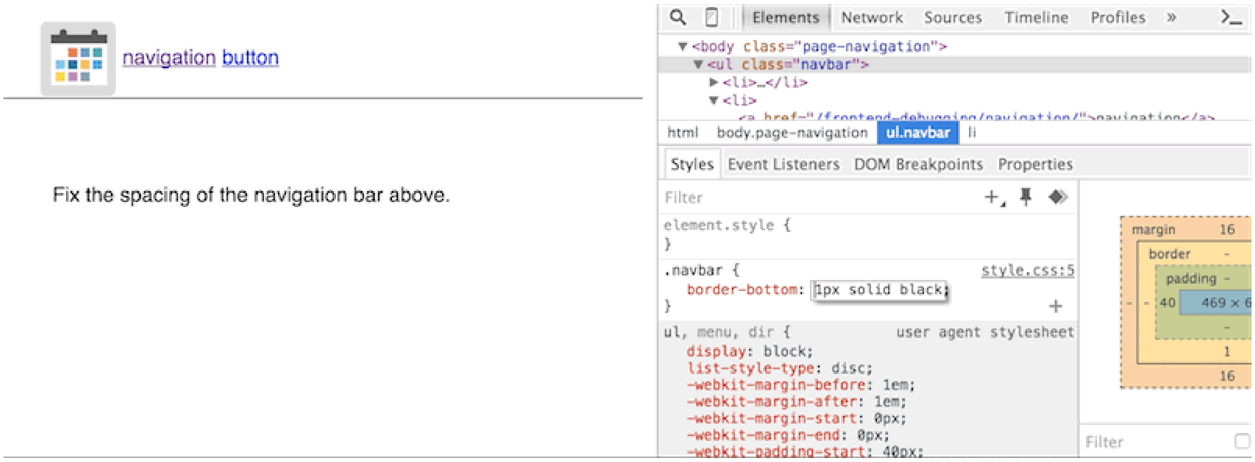


```
      ▼<li>
          <a href="/frontend-debugging/navigation/"
        </li>
      ▼<li>
```

```
element.style {
}
a:-webkit-any-link {          user agent stylesheet
   color: -webkit-link;
   text-decoration:▶underline;
   cursor: auto;
}
Inherited from li
.navbar li {                           style.css:17
   display: inline;
   list-style-type: none;
}
li {                          user agent stylesheet
   display: list-item;
   text-align: -webkit-match-parent;
}
Inherited from ul.navbar
ul, menu, dir {               user agent stylesheet
   display: block;
   ~~list-style-type: disc;~~
   -webkit-margin-before: 1em;
   -webkit-margin-after: 1em;
   -webkit-margin-start: 0px;
   -webkit-margin-end: 0px;
   -webkit-padding-start: 40px;
}
Inherited from body.page-navigation
```

Note: Somewhere in there, you will see `user agent stylesheet`, which are the styles the browser provides by default. Unfortunately, these differ slightly from browser to browser, which is where [CSS resets] or libraries like [Bootstrap] come in handy.

The coolest part about this pane is that you can tweak style rules, and see the changes immediately. Click on existing property declaration to edit it, or click below them to add a new one. For number/pixel values, you can also press the up/down arrows to increase/decrease.



Fix the spacing of the navigation bar above.

```
▼<body class="page-navigation">
  ▼<ul class="navbar">
    ▶<li>…</li>
    ▼<li>
```

```
element.style {
}
.navbar {                                    style.css:5
  border-bottom: 1px solid black;
}
ul, menu, dir {              user agent stylesheet
  display: block;
  list-style-type: disc;
  -webkit-margin-before: 1em;
  -webkit-margin-after: 1em;
  -webkit-margin-start: 0px;
  -webkit-margin-end: 0px;
  -webkit-padding-start: 40px;
```

Go ahead and play around – make the navigation bar look better!

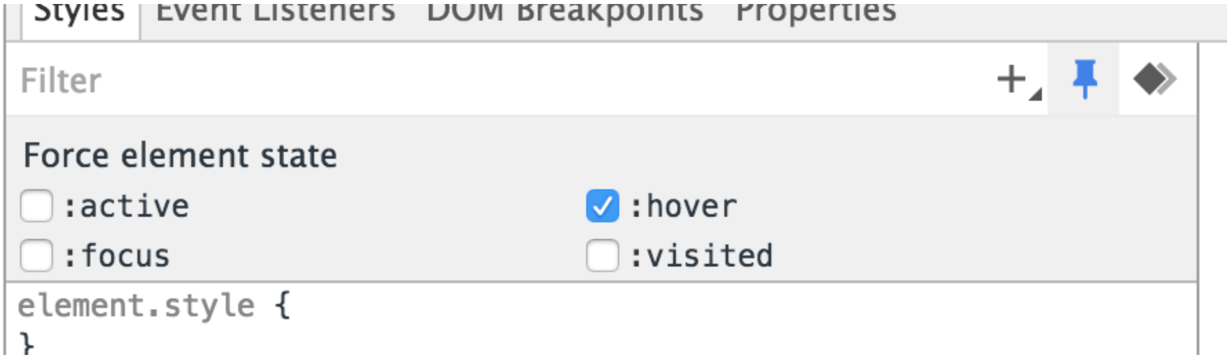
Page states

The Styles pane is extremely useful for troubleshooting styles that aren't being applied the way you intended. It's one thing when the page is static, but more challenging when the page changes as you interact with it. Let's take a look at another example:

onemonth.github.io/frontend-debugging/button/

Notice that when you hover over the button, the text disappears. Without a debugger, you would need to stare at your stylesheet to figure out what was changing between the two states, to figure out how to get the text to remain visible. Luckily there's a better way!

Click the 📌 button near the top of the Styles panel. This shows a handful of checkboxes, which allow you to toggle the element's state without having to actually interact with the page. Click the `:hover` box to enable it.



Force element state

□ :active ☑ :hover
□ :focus □ :visited

```
element.style {
}
```

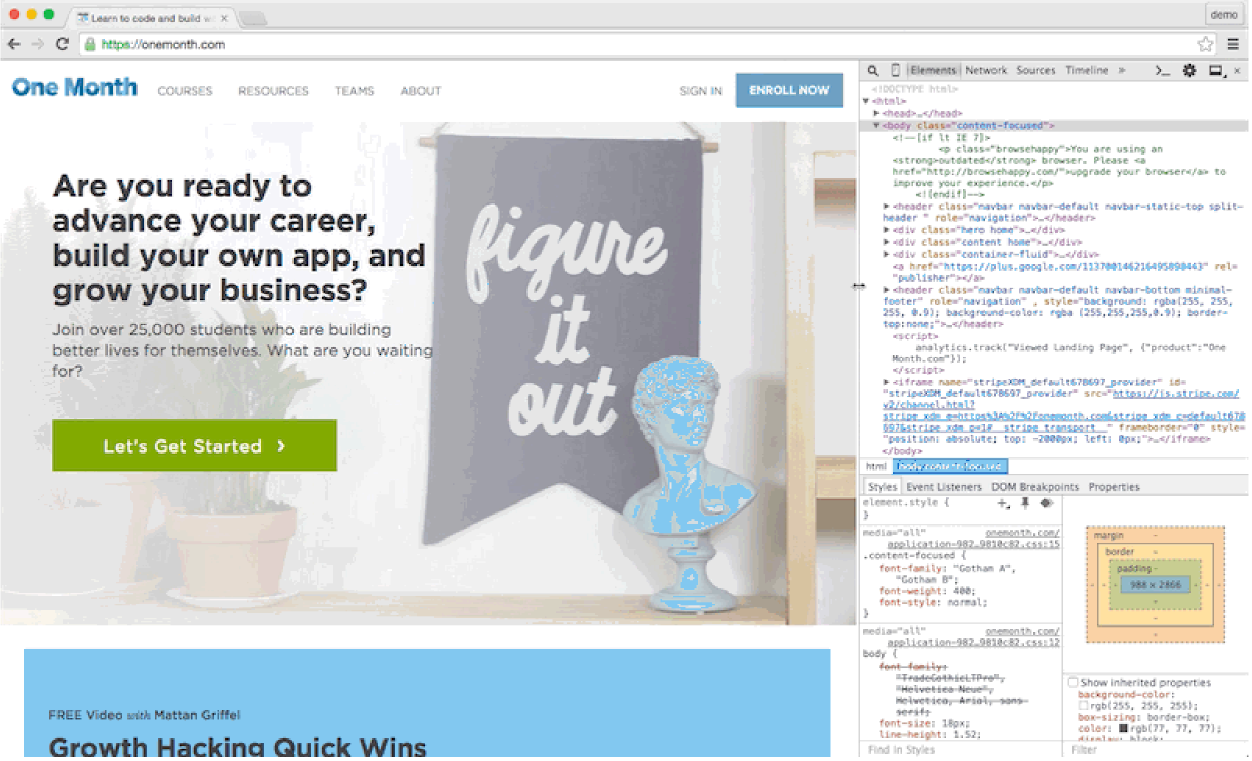
From here, let's make the colors invert when the button is hovered over: the background should go from white to blue, and the text should go from blue to white. Give it a try!

A few other cool things to note about the Styles panel:

- The checkbox next to each style declaration allows you to toggle it on and off. This allows you to see what effect each has on the page.
- When a style declaration's value is a color (like the background of the `<button>`), it shows a preview in a small box next to it. Click the box to bring up the **Color Picker**.
- Each style rule shows the file and line number of the file that it appears in. Click that line to jump to that file in the **Sources** tab. We will be looking closer at this tab in our next post.

Emulating devices

When you have DevTools open and resize the browser (or the DevTools window), you will see the resolution of the page displayed. When building a site using responsive design, this is especially useful for seeing how pages appear at different dimensions.



Are you ready to advance your career, build your own app, and grow your business?

Join over 25,000 students who are building better lives for themselves. What are you waiting for?

FREE Video with Mattan Griffel

Growth Hacking Quick Wins

```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
  ▼<body class="content-focused">
      <!--[if lt IE 7]>
            <p class="browsehappy">You are using an
      <strong>outdated</strong> browser. Please <a
      href="http://browsehappy.com/">upgrade your browser</a> to
      improve your experience.</p>
          <![endif]-->
    ▶<header class="navbar navbar-default navbar-static-top split-
    header " role="navigation">…</header>
    ▶<div class="hero home">…</div>
    ▶<div class="content home">…</div>
    ▶<div class="container-fluid">…</div>
      <a href="https://plus.google.com/113700146216495898443" rel=
      "publisher"></a>
    ▶<header class="navbar navbar-default navbar-bottom minimal-
    footer" role="navigation" , style="background: rgba(255, 255,
    255, 0.9); background-color: rgba (255,255,255,0.9); border-
    top:none;">…</header>
      <script>
          analytics.track("Viewed Landing Page", {"product":"One
      Month.com"});
      </script>
    ▶<iframe name="stripeXDM_default678697_provider" id=
    "stripeXDM_default678697_provider" src="https://js.stripe.com/
    v2/channel.html?
    stripe_xdm_e=https%3A%2F%2Fonemonth.com&stripe_xdm_c=default678
    697&stripe_xdm_p=1#__stripe_transport__" frameborder="0" style=
    "position: absolute; top: -2000px; left: 0px;">…</iframe>
    </body>
```

```
element.style {
}

media="all"                    onemonth.com/
      application-982…9810c82.css:15
.content-focused {
   font-family: "Gotham A",
      "Gotham B";
   font-weight: 400;
   font-style: normal;
}

media="all"                    onemonth.com/
      application-982…9810c82.css:12
body {
   ~~font-family:~~
      ~~"TradeGothicLTPro",~~
      ~~"Helvetica Neue",~~
      ~~Helvetica, Arial, sans-~~
      ~~serif;~~
   font-size: 18px;
   line-height: 1.52;
```

```
□ Show inherited properties
  background-color:
  □rgb(255, 255, 255);
  box-sizing: border-box;
  color: ■rgb(77, 77, 77);
  display: block;
```

Going a step beyond just resizing the browser, the last major DevTools feature for styling we're going to cover is **Device Mode**. Device Mode allows you to do several things:

- Render the page at a specific resolution
- Pretend to be a particular device (a.k.a. "user agent spoofing")
- Trigger touch events
- Throttle ("slow down") the network, to see how your site would load over a slow connection

To enable, click the 📱 button at the top left, next to the magnifying glass.

If you need to test specific devices in even more depth, you will want to download a device simulator, e.g. the iOS Simulator or the Android Emulator.

Conclusion

We hope this tutorial was useful in helping to understand various techniques for troubleshooting the styling of your site. If you have any questions, let us know! Also, if you'd like to find out when the next blog post in this series (and beyond) is released, sign up for the One Month newsletter on the homepage. Thanks!