# Intro to CSS

CSS is a tool used to styled your html page.

*CSS*, or Cascading Style Sheets, is a language that web developers use to *style* the HTML content on a web page. If you're interested in modifying colors, font types, font sizes, shadows, images, element positioning, and more, CSS is the tool for the job!

# Inline Styles

You can write CSS code directly within HTML code using *inline styles*.

To style an HTML element, you can add the `style` attribute directly to the opening tag. After you add the attribute, you can set it equal to the CSS style(s) you'd like applied to that element.

```
<p style="color: red;">I'm learning to code!</p>
```

If you'd like to add *more* than one style with inline styles, simply keep adding to the `style`attribute. Make sure to end the styles with a semicolon (`;`).

```
<p style="color: red; font-size: 20px;">I'm learning to code!</p>
```

# The <style> Tag

Inline styles are  faster but they also have limitations. for example, multiple `<h1>` elements, you would have to add inline styling to each element manually. In addition, you would also have to maintain the HTML code when additional `<h1>`elements are added.

Fortunately, HTML allows you to write CSS code in its own dedicated section with the `<style>` element. CSS can be written between opening and closing `<style>` tags. To use the `<style>` element, it must be placed inside of the `<head>` element.

```
<head>
    <style>



    </style>
  </head>
```

After adding a `<style>` tag in the head section, you can begin writing CSS code.

```
<head>
    <style>
      p {
        color: red;
        font-size: 20px;
      }
    </style>
  </head>
```

The CSS code in the example above changes the color of all paragraph text to red and also changes the size of the text to 20 pixels.

# The .css file

(HTML files contain only HTML code, and CSS files contain only CSS code).

You can create a CSS file by using the **.css** file name extension, like so: **style.css** to style a page without sacrificing the readability and maintainability of your HTML file.

Style.css

```
p {

    font-family: Arial;
  }
```

# Linking the CSS File

When HTML and CSS code are in separate files, the files must be linked. Otherwise, the HTML file won't be able to locate the CSS code, and the styling will not be applied.

You can use the `<link>` element to link HTML and CSS files together.
The `<link>` element must be placed within the head of the HTML file. It is a self-closing tag and requires the following three attributes:

1. `href` — like the anchor element, the value of this attribute must be the address, or path, to the CSS file.
2. `type` — this attribute describes the type of document that you are linking to (in this case, a CSS file). The value of this attribute should be set to `text/css`.
3. `rel` — this attribute describes the relationship between the HTML file and the CSS file. Because you are linking to a stylesheet, the value should be set to `stylesheet`.

When linking an HTML file and a CSS file together, the `<link>` element will look like the following:

```
<link href="c:/shubham/stylesheets/style.css" type="text/css" rel="stylesheet">
```

Specifying the path to the stylesheet using a URL is one way of linking a stylesheet.

If the CSS file is stored in the same directory as your HTML file, then you can specify a relative path instead of a URL, like so:

```
<link href="./style.css" type="text/css" rel="stylesheet">
```

Using a relative path is very common way of linking a stylesheet.

# Tag Name

CSS can select HTML elements by using an element's tag name.

```
p {

   }
```

In the example above, all paragraph elements will be selected using a CSS *selector*.

The selector in the example above is `p`. Note that the CSS selector matches the HTML tag for that element, but without the angle brackets.

In addition, two curly braces follow immediately after the selector (an opening and closing brace, respectively). Any CSS properties will go inside of the curly braces to style the selected elements.

# Class Name

It's also possible to select an element by its `class`attribute dor styling in .css file.

```
p class="brand">Sole Shoe Company</p>
```

The paragraph element in the example above has a `class` attribute within the `<p>` tag. The `class` attribute is set to `"brand"`. To select this element using CSS, a period (`.`) must be prepended to the class's name.

```
.brand {
}
```

# Multiple Classes

Luckily, it's possible to add more than one class name to an HTML element's `class` attribute.

```
.green {
    color: green;
  }

  .bold {
    font-weight: bold;
  }
```

Then, you could include both of these classes on one HTML element like this:

```
<h1 class="green bold"> ... </h1>
```

We can add multiple classes to an HTML element's `class` attribute by separating them with a space. This enables us to mix and match CSS classes to create many unique styles without writing a custom class for every style combination needed.

```
<html>
     <body>
        <h1 class="title uppercase">Top Vacation Spots</h1>
     </body>
</html>
```

.CSS

```
.title{
    color: teal;
  }
  .uppercase{
    text-transform: uppercase;
  }
```

# ID Name

If an HTML element needs to be styled uniquely (no matter what classes are applied to the element), we can add an ID to the element. To add an ID to an element, the element needs an `id` attribute:

```
<h2 id="large-title"> ... </h2>

<h1 class="title uppercase" id="article-title">Top Vacation Spots</h1>
```

Then, CSS can select HTML elements by their `id` attribute. To select an `id` element, CSS prepends the `id` name with a hashtag (`#`). For instance, if we wanted to select the HTML element in the example above, it would look like this:

```
#article-title {
font-family: cursive;
text-transform: capitalize;
}
```

The `id` name is `large-title`, therefore the CSS selector for it is `#large-title`.

# Classes and IDs

CSS can select HTML elements by their tag, class, and ID. CSS classes and IDs have different purposes. CSS classes are meant to be reused over many elements.

*For instance, imagine a page with two headlines. One headline needs to be bold and blue, and the other needs to be bold and green. Instead of writing separate CSS rules for each headline that repeat each other's code, it's better to write a `.bold` CSS rule, a `.green` CSS rule, and a `.blue`CSS rule. Then you can give one headline the `bold green` classes, and the other the `bold blue`classes.*

ID is meant to style only one element. Since IDs override class and tag styles, they should be used sparingly(use properly) and only on elements that need to always appear the same.

# Specificity

Specificity is the order by which the browser decides which CSS styles will be displayed. A best practice in CSS is to style elements while using the lowest degree of specificity, so that if an element needs a new style, it is easy to override.

***First priority css selector= id then classes then tags in css***

```
<h1 class="title" id="t">Breaking News</h1>
```

```
h1 {
   color: blue;
  }
  #t{
    color: red;
  }
  .title{
    color: green;
  }
```

# Chaining Selectors

When writing CSS rules, it's possible to require an HTML element to have two or more CSS selectors at the same time.

This is done by combining multiple selectors, which we will refer to as chaining. For instance, if there was a `.special` class for `h1` elements, the CSS would look like:

```
<h2 class="destination">1. Florence, Italy</h2>
<h5 class="destination">Top Attractions</h5>
```

```
h2.destination {
    font-family: cursive;
  }
```

Here, h1 and h2 element both have same name class but, This CSS is applicable for h2 element.

# Nested Elements

In addition to chaining selectors to select elements, CSS also supports selecting elements that are nested within other HTML elements. For instance, consider the following HTML:

```
<ul class='main-list'>
    <li> ... </li>
    <li> ... </li>
    <li> ... </li>
  </ul>
```

The nested `<li>` elements are selected with the following CSS:

```
.main-list li { }
```

In the example above, `.main-list` selects the `.main-list` element (the unordered list element). The nested `<li>` are selected by adding `li` to the selector, separated by a space, resulting in `.main-list li` as the final selector (note the space in the selector).

# Chaining and Specificity

Specificity refers to how a browser decides which styles to display when there are multiple styles defined that could apply to the same element.

```
<h5 class="pp">By: Stacy Gray</h5>
```

```
h5{
    color: green;
  }
  .pp h5{
    color: red;
  }
```

here, h5 css selector works.

# Important

There is one thing that is even more specific than IDs: `!important`. `!important` can be applied to specific attributes instead of full rules. It will override *any* style no matter how specific it is. As a result, it should almost never be used. Once `!important` is used, it is very hard to override.

```
<body>
   <p class="main">hello</p>

    </body>
```

```
p {
    color: blue !important;
  }

  .main {
    color: red;
  }
```

The `!important` flag is only useful when an element appears the same way 100% of the time. Since it's almost impossible to guarantee that this will be true throughout a project and over time, it's best to avoid `!important` altogether

# Multiple Selectors

I it's possible to add CSS styles to multiple CSS selectors all at once. This prevents writing repetitive code.

```
h1 {
    font-family: Georgia;
  }

  .menu {
    font-family: Georgia;
  }
```

Instead of writing `font-family: Georgia` twice for two selectors, we can separate the selectors by a comma to apply the same style to both, like this:

```
h1,
.menu {
  font-family: Georgia;
}
h5,p {
  font-family: Georgia;
}
```

By separating the CSS selectors with a comma, both the `h1` and the `.menu` elements will receive the `font-family: Georgia` styling.

# Introduction To Visual Rules

# Font Family

To change the typeface of text on your web page, you can use the `font-family` property.

```
h1 {
    font-family: Garamond;
  }
```

The default typeface for all HTML elements is `Times New Roman`.

# Font Size

To change the size of text on your web page, you can use the `font-size` property.

```
p {
    font-size: 18px;
  }
```

# Font Weight

In CSS, the `font-weight` property controls how bold or thin text appears.

The `font-weight` property has a another value: `normal`.

```
p {
```

```
    font-weight: bold;
  }
```

# Text Align

text always appears on the left side of the browser.

To align text we can use the `text-align` property.

```
h1 {
    text-align: right;
  }
```

The `text-align` property can be set to one of the following three values:

- `left, center, right`

# Color

Color can affect the following design aspects:

1. Foreground color
2. Background color

- `color`: this property styles an element's foreground color
- `background-color`: this property styles an element's background color

```
h1 {
    color: red;
    background-color: blue;
  }
```

In the example above, the text of the heading will appear in red, and the background of the heading will appear blue.

# Opacity

Opacity is the measure of how transparent an element is. It's measured from 0 to 1, with 1 representing 100%, or fully visible and opaque, and 0 representing 0%, or fully invisible.

```
.overlay {
    opacity: 0.5;
  }
```

# Background Image

CSS has the ability to change the background of an element.

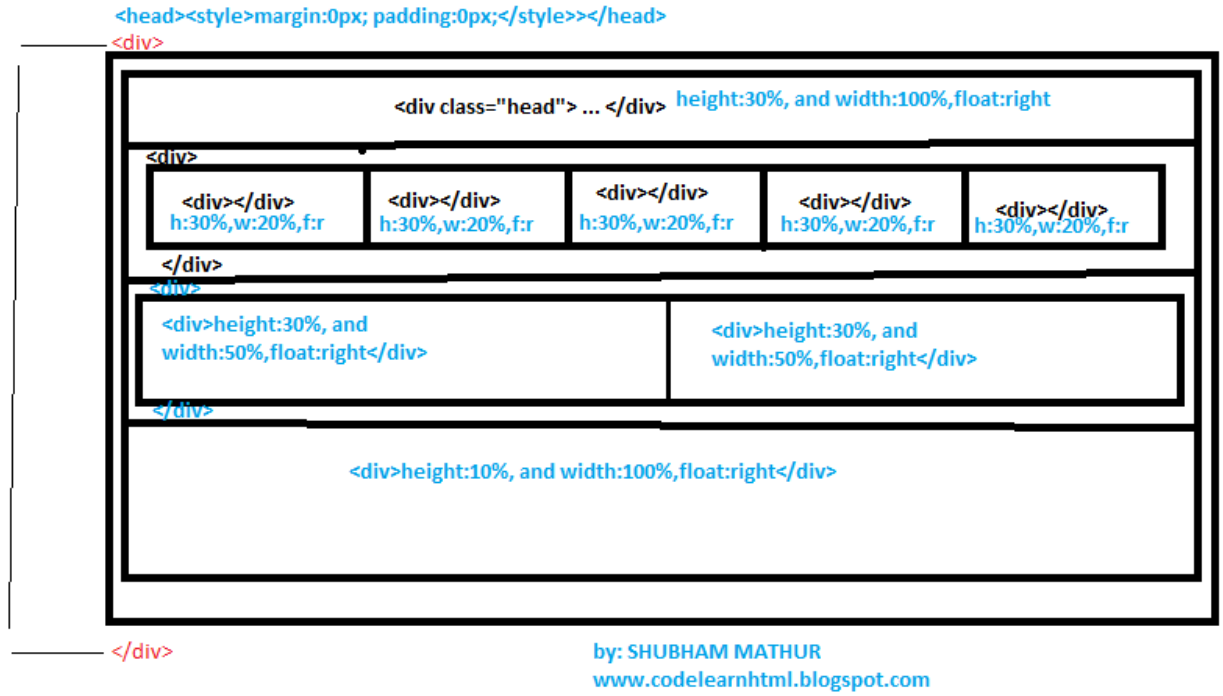
```
.main-banner {
  background-image: url("https://www.example.com/image.jpg");
}
```

The value provided to `background-image` is a `url`. The `url` should be a url to an image.

To link to an image inside an existing project, you must provide a relative file path.

```
.main-banner {
    background-image: url("images/mountains.jpg");
  }
```

# Partition of Screen using <Div>




```
<html>
    <head>
        <style>
            * {

        margin: 0px;
        padding: 0px;
      }

        </style>
    </head>
    <body>
        <div class="head">
            <div class="layer1" style="background-color: red; width:100%;
height:30%; margin:0px;padding:0px;">

            </div>
            <div class="headerlayer2">
```

```
                    <div class="layer1" style="background-color: green;
width:50%; height:30%;float:right;">

                    </div>
                    <div class="layer2" style="background-color: blue; width:50%;
height:30%; float:right;">

                    </div>
            </div>

<div class="headerlayer3">
                    <div class="layer31" style="background-color: red; width:20%;
height:30%;float:right;">

                    </div>
                    <div class="layer32" style="background-color: pink;
width:20%; height:30%; float:right;">

                    </div>
                    <div class="layer33" style="background-color: yellow;
width:20%; height:30%;float:right;">

                    </div>
                    <div class="layer34" style="background-color: peru;
width:20%; height:30%;float:right;">

                    </div>
                    <div class="layer35" style="background-color: powderblue;
width:20%; height:30%;float:right;">

                    </div>
            </div>
    <div class="layer31" style="background-color:saddlebrown; width:100%;
height:10%;float:right;">

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

## Using Chrome DevTools for CSS Visual Rules

Requirements:

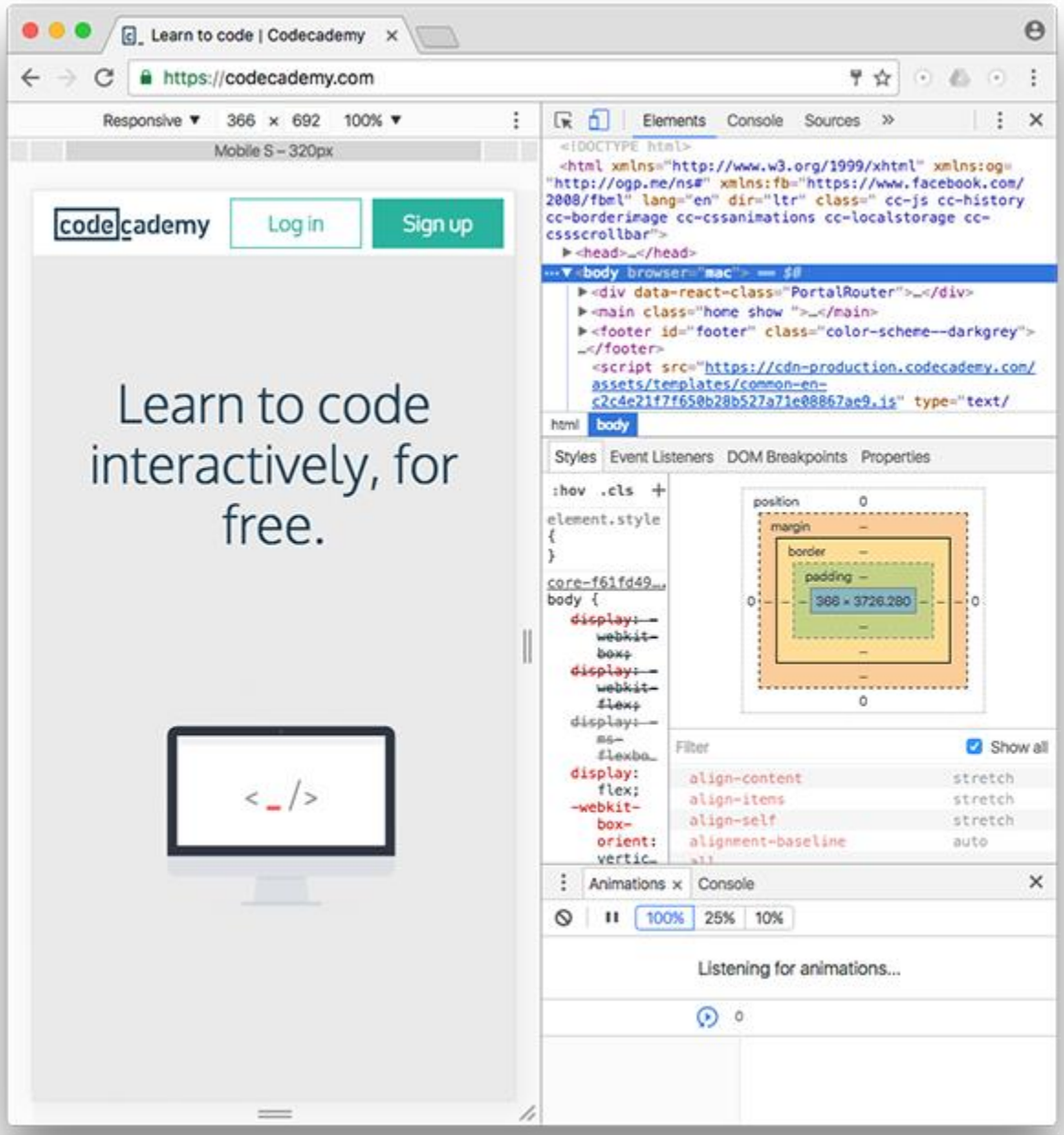
- An Internet browser

***Introduction***

Browser developer tools allow web developers to quickly collect important information on most websites. These tools are available within most major web browsers, like Chrome, Safari, and Firefox, to name a few. Because Google Chrome is the preferred browser for many professional developers, we'll learn how to use the browser developer tools within Google Chrome, known as Chrome DevTools.

***Step 1: Accessing DevTools***

The quickest way of accessing DevTools in Chrome is to navigate to any website (like this one) and *right click* (press `Ctrl` and click for a single button mouse) anywhere on the page. Upon doing so, a menu will appear directly beside the area you clicked on. In the menu, select "Inspect." This will automatically launch DevTools within your browser. DevTools will appear as a window on either the bottom or right hand side of your screen. It should look something like this:




You should see the `Elements`, `Console`, `Sources`, and `Network` tabs, among many others.

This rest of this article will focus exclusively on the `Elements` tab.

***Step 2: Using DevTools to View CSS Styles***

DevTools can provide you with a lot of information about a website, but it's particularly exceptional at examining a page's HTML elements, along with the CSS styles for those respective elements. Let's try it out!

1. Open an incognito Chrome browser (in the browser's menu, click on "File" then "New Incognito Window"). This will allow you to read this article while completing the following steps.

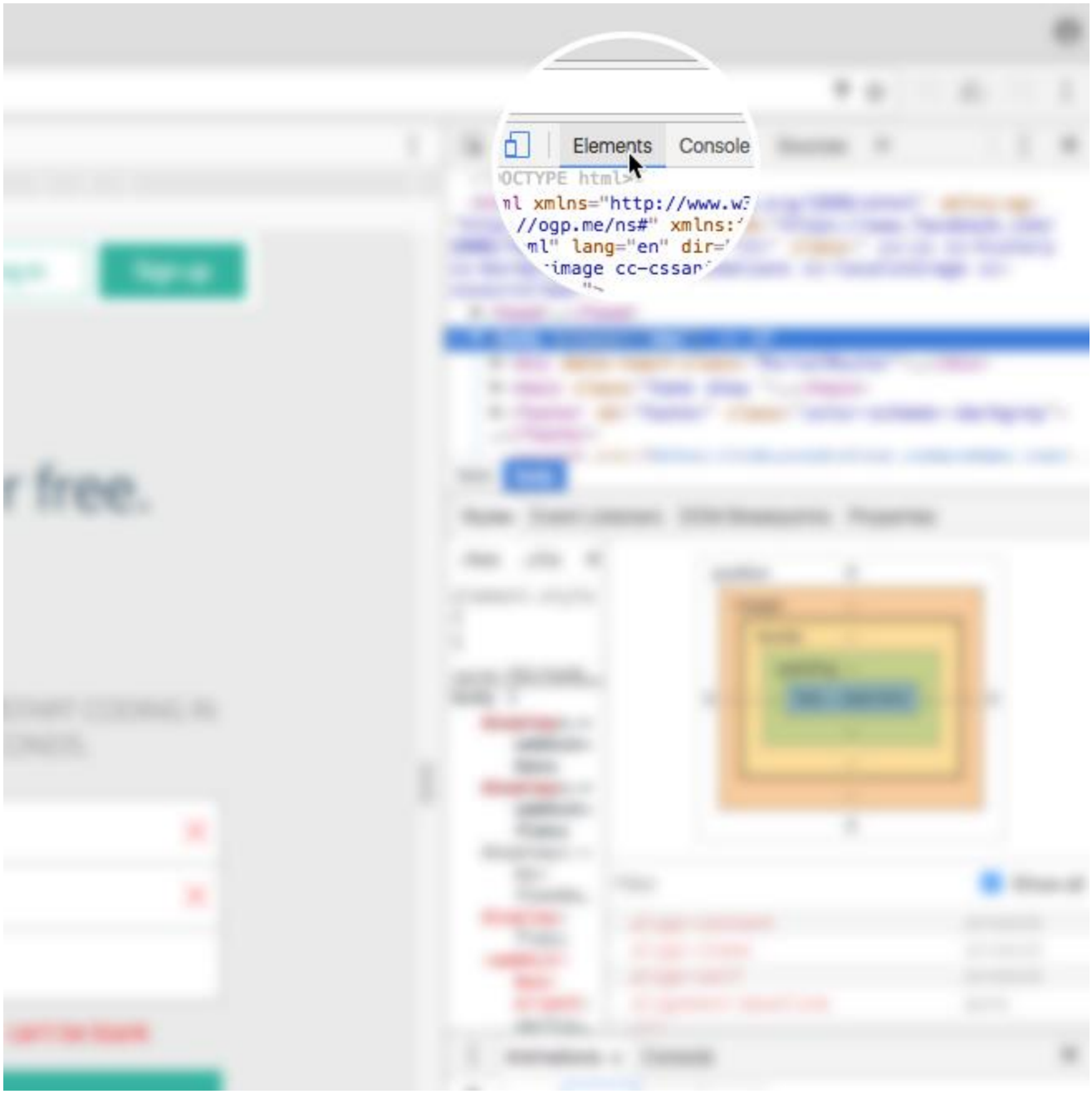
2. Navigate to Codecademy's homepage (make sure you are logged out).

3. Right click (or `Ctrl` and click simultaneously) on any text on the page. (This article uses screenshots from a previous version of the Codecademy home page. Your home page may look different from the one shown in the screenshots.)

4. Select "Inspect" in the menu that appears.

5. DevTools should appear at the bottom of your page (it's normal if its appears in another location, as its location can be changed).

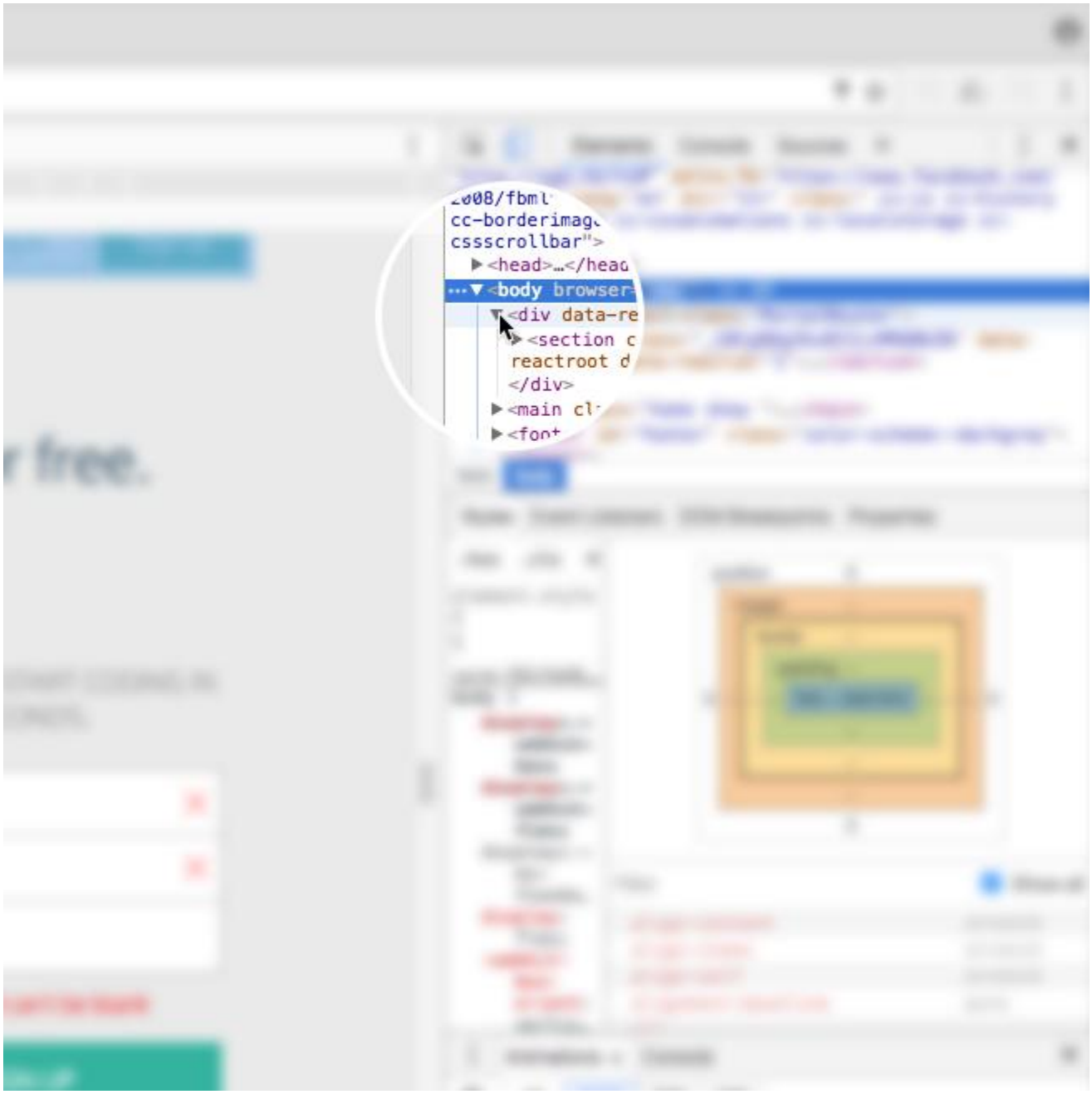
6. Click on the "Elements" tab of DevTools (if you're not already on it).



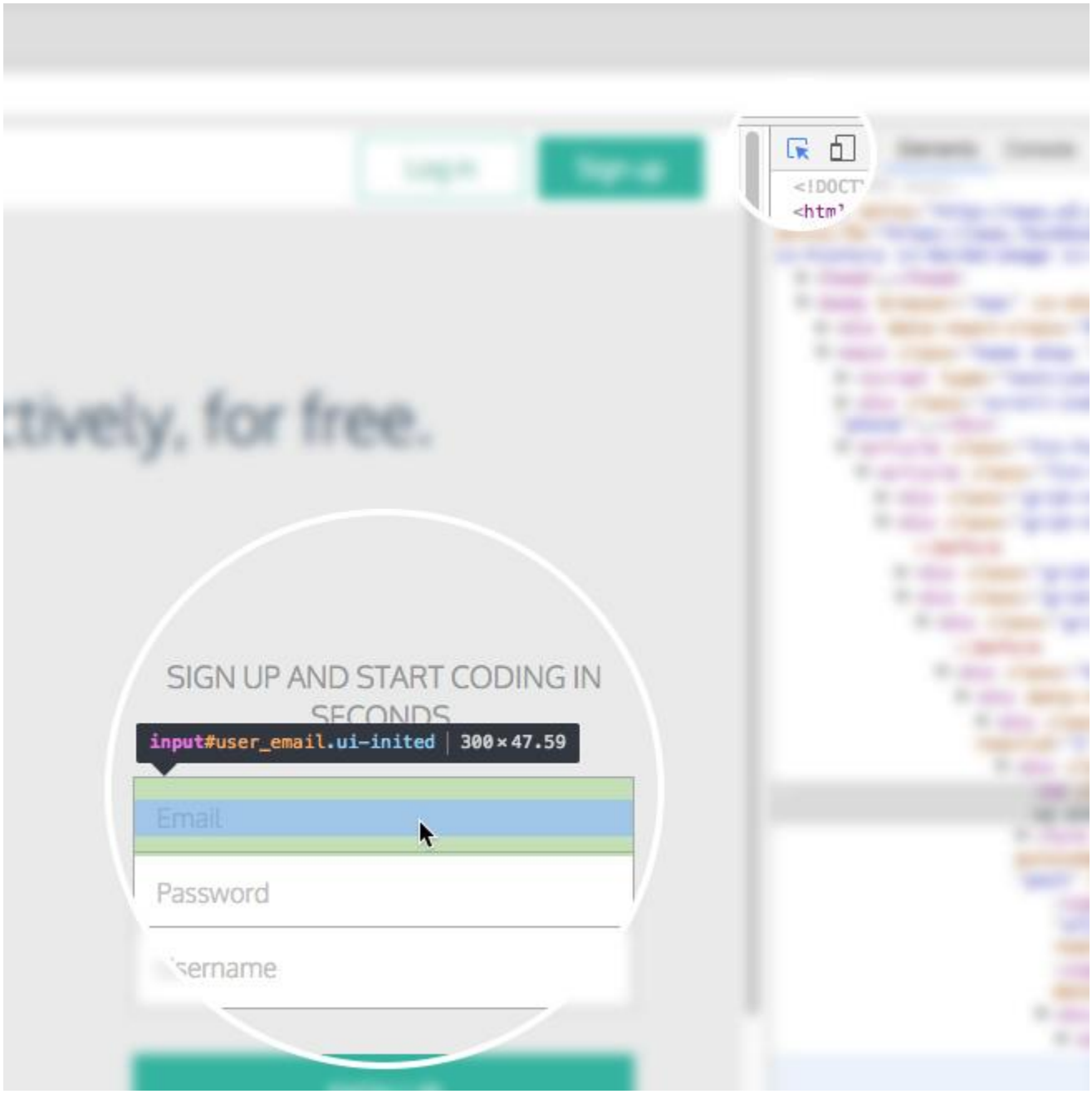

7. In the left pane, notice the interactive DOM (HTML elements) that contains the current content of the web page.

8. Mouse over the HTML code – as you mouse over, notice that DevTools will highlight the corresponding HTML element on the web page.

9. Note that you can expand closed elements by clicking the arrow directly to the left of them.

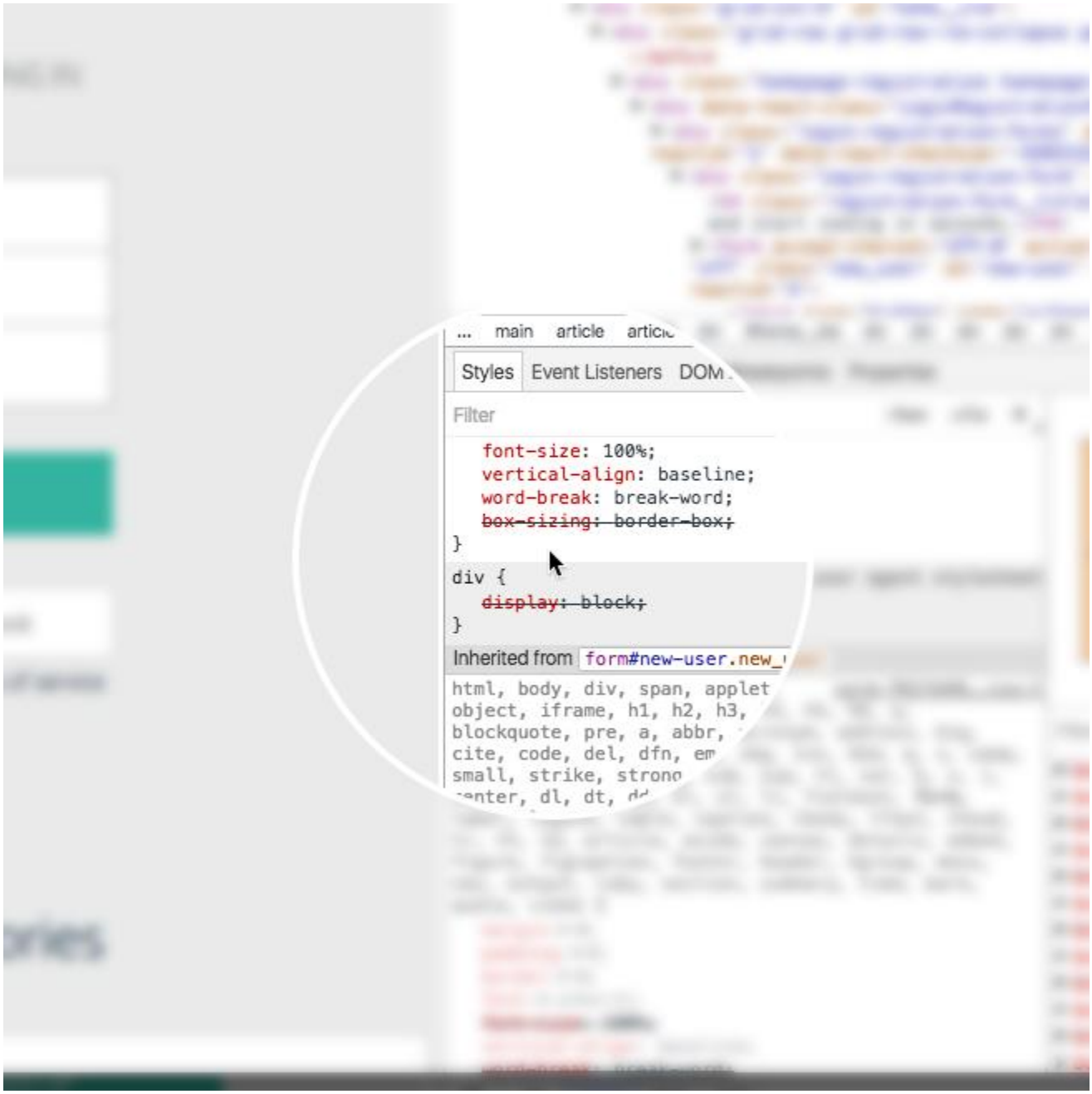



10. Alternatively, click the "Select element" icon (shown in the image below) in the top-left corner of the console and then click on an element within the web page – this is a much quicker way of accessing a specific element on the web page that you want to inspect.



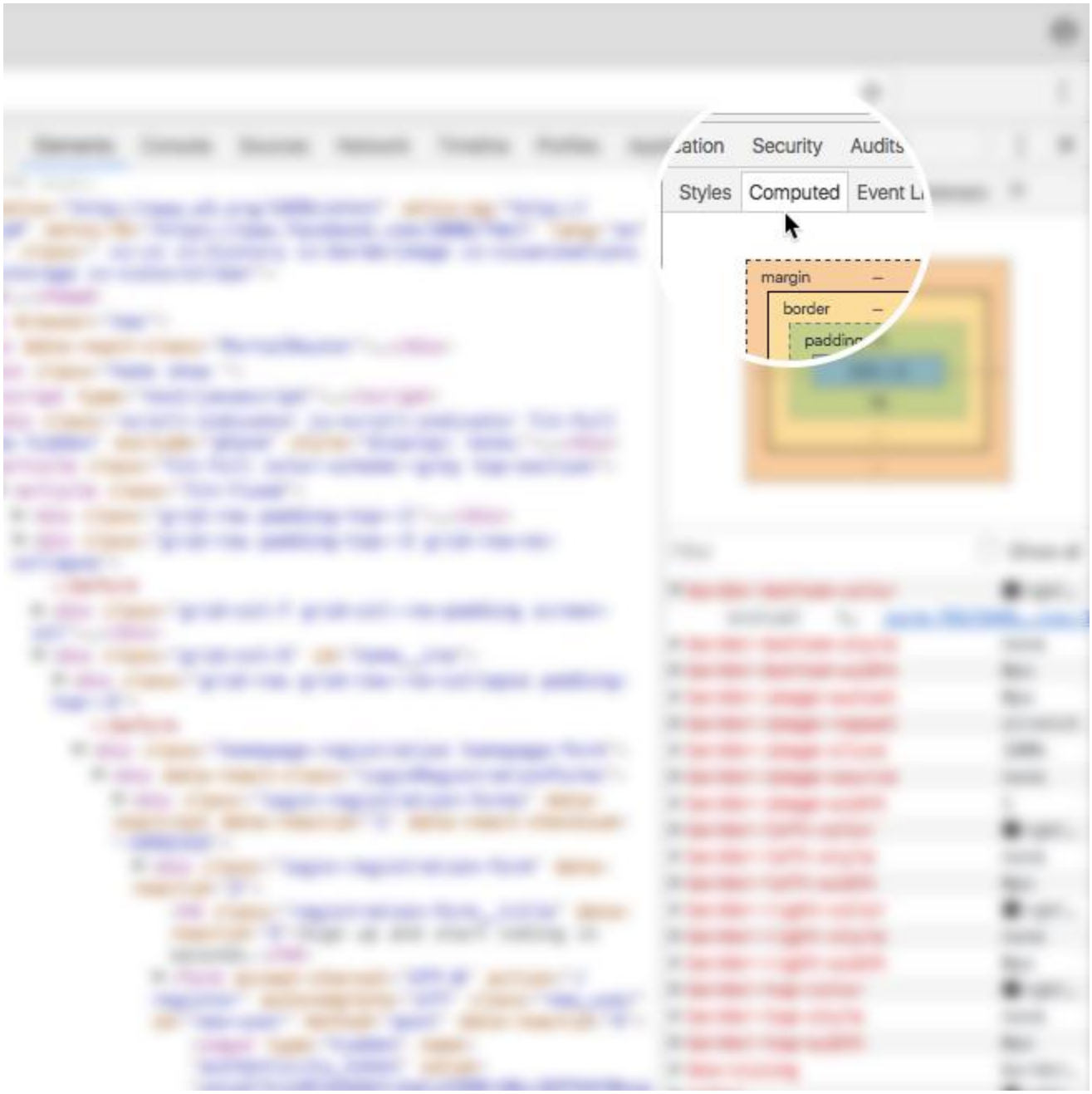

11. On the right hand side of DevTools, click on the tab named `Styles` (if you're not already on it) — this tab displays *all* of the CSS styles associated with the element highlighted in the left side of DevTools.

12. Scroll down in the `Styles` tab, notice that some CSS styles are crossed out with a horizontal black line.




13. Remember, the `Styles` tab shows *all* styles applied to that element (rules can often be overwritten by more specific rules, which causes the horizontal black line through some CSS rules, denoting that that rule is not being used).

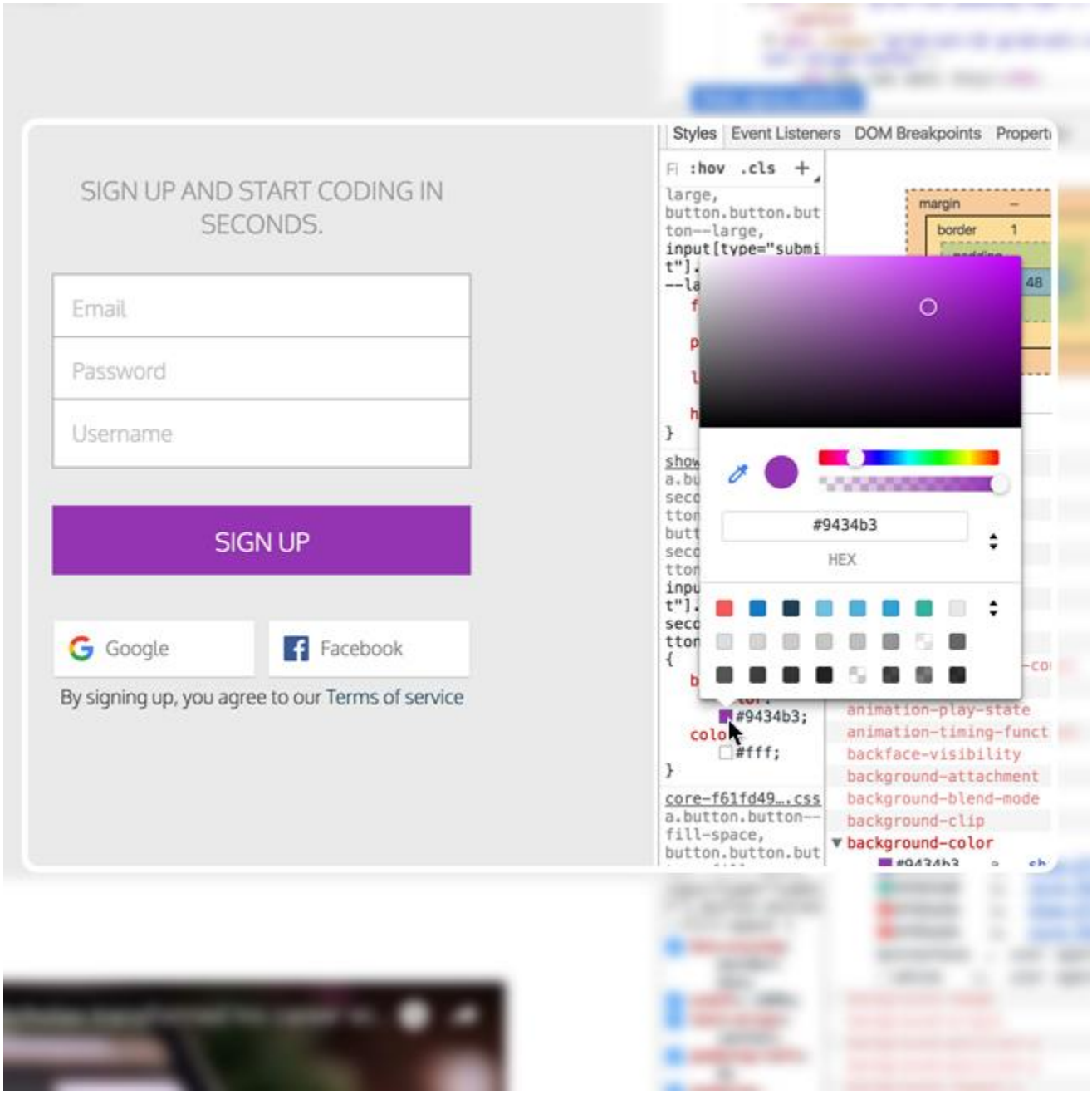
14. To instead view *only* the styles applied to that specific element, click on the `Computed` tab directly next to the `Styles` tab in the right pane. In this pane, you will see only the styles that are being applied to that element, also known as the *computed styles*. (If the `Computed` tab is not appearing for you, your browser may be sized too small. Expand the width of the browser until it appears.)




***Step 3: Modifying CSS Styles with DevTools***

DevTools is also useful for modifying *existing* CSS rules and previewing those changes directly on the page you're viewing.

To try it out, click again on the `Styles` tab in the right pane of DevTools (feel free to use the Codecademy website again). Scroll down to a CSS rule (one that is *not*crossed out with a black line), click on the value of any applied CSS rule, change the value, and press "Enter" (or "return") on your . You should see the change automatically update on the page.




There are a few things to keep in mind when using DevTools to modify a web page:

1. When you modify or change a CSS rule, you may be affecting more than one element.

2. DevTools provides easy-to-use tools when you modify certain CSS rules. (For example, when modifying color values, DevTools will provide you with a color picker to help you select a color.)

3. DevTools is only a *sandbox* tool, meaning that any changes you make to the web page will *not be saved*, so make sure to write down any changes you'd like to make when using DevTools for your own web page!

***Step 4: Add CSS Styles with DevTools***

In addition to modifying existing CSS rules, you can add *new* CSS rules as well. Let's continue using DevTools on the Codecademy homepage.

1. Locate some text on the home page (i.e. find a heading, paragraph, link, etc.).

2. Right click on the text and click "Inspect" in the menu that appears. DevTools will highlight the corresponding HTML element in its left pane.

3. Take a look at the `Styles` tab and click on the `+` icon in the top-right corner of the right pane – notice that this creates a new, empty CSS rule for that element.

4. Within the element's selector, click and add a new CSS declaration. The following is an example (feel free to add your own declaration):

```
background-color: red;
```

You should see the background color of the text you selected change to red. You can also continue to add your own CSS styling as you wish.

In the future, try this technique on your own website(s) as you build them from the ground up. Building with DevTools can result in a more efficient workflow, as it can help you avoid repeatedly saving and viewing changes manually.

***Step 5: Modify HTML with DevTools***

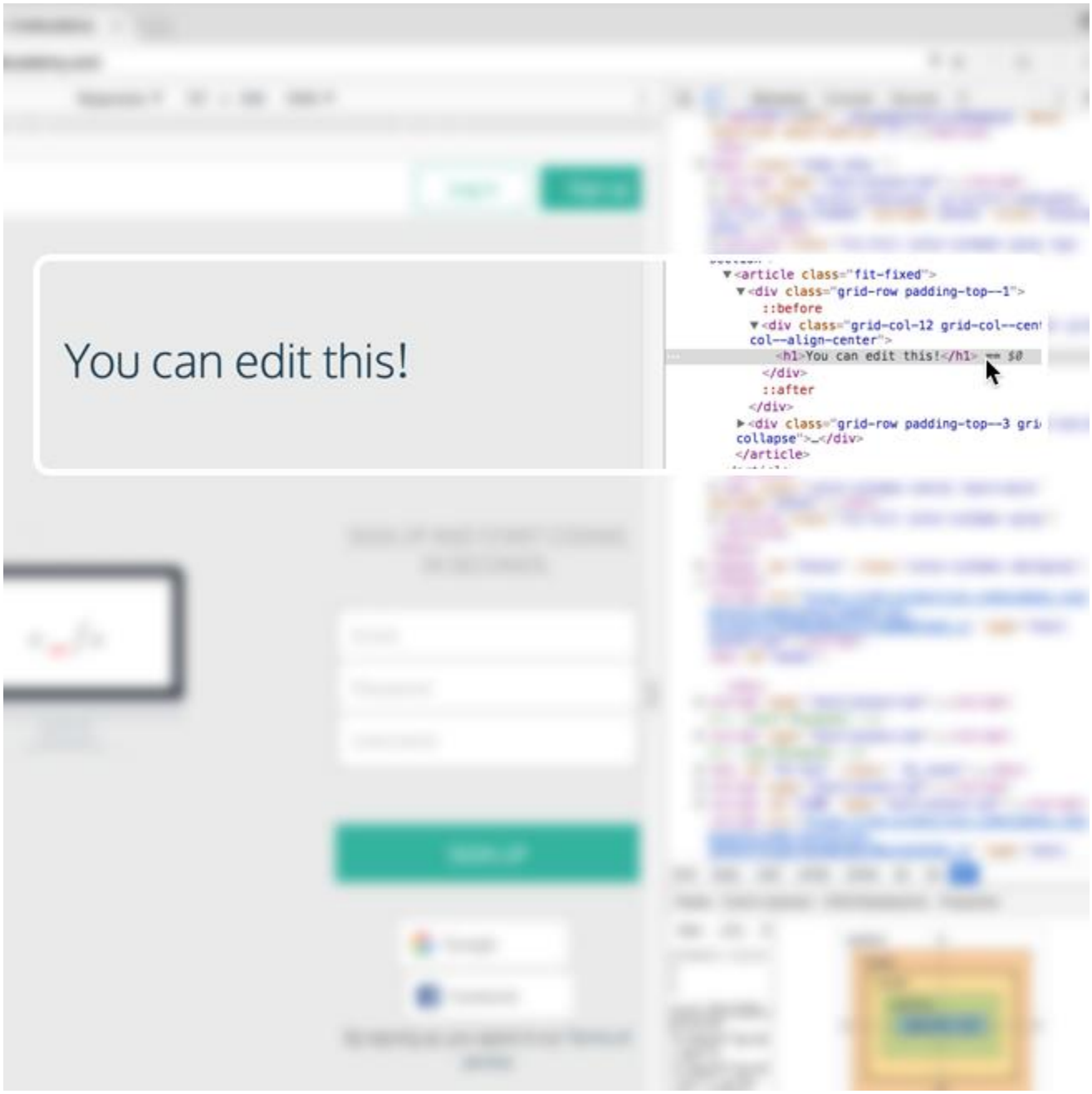
DevTools also lets you directly *modify* the HTML content of a web page. Let's try this out one more time on the Codecademy homepage.

1. Again, right click on a piece of text on the homepage and click "Inspect" in the menu that appears.

2. DevTools will automatically highlight the HTML code in the left pane associated with the content that you inspected on the web page.

3. In DevTools, double click on the text between the opening and closing tags of the text you right-clicked on.

4. Change the heading to say something else, like your name, or "Codecademy", and press Enter.




At this point, you should see the text change!

You can also add HTML of your own as well. Let's add an `<h2>` element directly below the element you just modified.

1. Right click on the element you just modified, a menu should appear. Click on "Edit as HTML." (You can also delete elements using this menu.)

2. A large text field should appear. Directly edit the HTML by adding an `<h2>` element below with the text of your choice.

3. To complete/view your changes, click on any other element in the left window pane or press `Command` and `Enter` at the same time (on a Mac keyboard).

What happens to the web page? Remember, these are sandboxed changes, so your changes will *not* be saved, nor permanently affect the website you are applying changes to.

***Review***

The Chrome web browser provides you with robust web developer tools known as DevTools. With DevTools, you can view a web page's existing DOM elements and associated styles, as well as modify and preview changes you make to the web page, resulting in an efficient workflow. If you're interested in learning more about DevTools, visit the official documentation at https://developer.chrome.com/devtools.

Happy coding!