



Checkpoint 6

More web elements

As you might imagine, there are many different code snippets that can work together to style a web page. There are so many, in fact, that this module can't cover every possible element and formatting technique. But this checkpoint will get you started by introducing you to some of the most common styles in HTML and CSS.

By the end of this checkpoint, you should be able to do the following:

- Write comments to accompany HTML and CSS code
- Use common elements, like lists and dividers, to structure content
- Style links and images using CSS

Code comments

As a professional developer, you'll regularly use *comments*, or notes in your code, to explain what your code is doing. You learned a bit about using comments in JavaScript. You learned that although you can see comments in the code, they are actually disabled—they're ignored by the web browser when the code runs.



You'll also use comments in HTML and CSS. It is a best practice to be disciplined about adding comments to your code, regardless of the language you're working in. Comments will save you and your fellow developers time and energy, helping you prevent confusion about what your code was trying to accomplish. Quite often, developers who don't add comments will look back at their own code and wonder, "What was I thinking? What does *this* mean?" In those moments, they're struggling to understand the code they've written. When used appropriately, comments describe what the code is *intended* to do, which helps everyone down the road.

Here are some of the primary reasons for using comments in HTML, CSS, JavaScript, and most other programming languages:

- To describe and explain complicated code and create reminders for yourself and others
- To deliberately disable sections of code
- To improve collaboration with other developers working on the code, helping them clearly understand what needs to be added, fixed, or removed in the code
- To add titles to the sections of long pages of code to make the pages easier to scan

It's important to note that HTML and CSS comments are not tags. They also are not written in the same way; they have a distinct syntax.

HTML comments



Comments in the HTML code are written with dashes `-` and an exclamation point `!`, all enclosed in two end brackets `<>`. They're structured like this: `<!-- HTML comment -->`. Take a look at the example below.

```
<!-- This would be an HTML comment. Useful notes can  
  
<!--  
    This works  
    for multiple  
    lines as well  
-->
```

Typically, code editors will gray out the text in a comment to visually demonstrate that the block of code isn't in use.

CSS comments

Comments in CSS are written with asterisks `*` and forward slashes `/`. They're structured like this: `/* CSS comment */`. Take a look at the example below.

```
/* Hello, commenting! */  
  
/*  
    I can also be on multiple lines!  
*/
```

Any CSS code that sits between the opening `/*` symbol and the closing `*/` symbol is completely ignored by the web browser. In HTML and JS, CSS comments are grayed out in code editors.

Demo: Commenting

Review the code in the Repl.it below, and play around with the code comments. What do the code comments offer? Do they make the code sample easier to read and understand?

The screenshot shows a Repl.it code editor interface. At the top, there is a green 'Run' button with a play icon and a link to 'open in @replit'. Below this, a file named 'index.html' is open, showing a single line of code numbered '1'. The interface includes a sidebar on the left with a file icon and a cube icon. At the bottom, there is a URL bar with the address 'https://Commented-Sections--thinkful.repl.co' and a 'Console' button. A 'Shell' button is also visible. A large white circle is present in the bottom right corner of the interface.

HTML lists

Now that you understand how to use comments, you're ready to dive a bit deeper into styling. In building a website, you may want to structure your content into lists. *Lists* are used to break up and organize content within a web page, making it much easier for a user to read and understand.

HTML supports two kinds of lists: *unordered lists*, for list items that have no particular order, and *ordered lists*, for items that must be laid out in a specific sequence or arrangement. Unordered lists often use bullets, and ordered lists often use consecutive numbers or letters.

You'll use these HTML tags to make these lists:

- ``: This stands for *unordered list*. You'll use the opening tag `` and the closing tag ``.
- ``: This stands for *ordered list*. You'll use the opening tag `` and the closing tag ``.
- ``: This stands for *list item*. You'll use the opening tag `` and the closing tag `` to set off every item in a list.

Check out the examples below.

```
<h2>My Hobbies</h2>
<ul>
  <li>Skiing</li>
  <li>Painting</li>
  <li>Coding</li>
</ul>

<h2>My Top 3 Movies</h2>
```

```
<ol>
  <li>Inside Out</li>
  <li>Up</li>
  <li>Coco</li>
</ol>
```

Drill: List practice

Take a few minutes to code both an unordered and an ordered list in the Repl.it below. You can use any list items you wish.

(If you're struggling to think of something, consider making a list of your favorite foods in no particular order. Then, make the same list of your favorite foods in a specific order, ranking them from your favorite to least favorite.)

Run ►

open in repl.it



index.html



```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-wi
6      <title>List: Practice</title>
7      <link href="style.css" rel="stylesheet" type="
8    </head>
9    <body>
10
11   Types of Lists:
12   To Do
13   Cleaning
14   Shopping
15   Gift
16   Packing
17   Song
18
```

<https://List-Practice--thinkful.repl.co>

Console

Shell

Outline

Feel free to compare your code with the complete sample below:

[List Practice Complete](#)

Styling HTML lists

Remember how you *targeted*, or selected in the code, HTML headings and paragraphs to apply CSS styles to? You can do the same thing with HTML lists. The ``, ``, and `` tags can all be targeted and styled with CSS. In targeting these tags, you have a few options you can play with:

- `ul { }`: This targets all unordered lists.
- `ol { }`: This targets all ordered lists.
- `li { }`: This targets all list items of both unordered and ordered lists.

Demo: List items

Take a moment to review and experiment with the Repl.it below. This code sample shows how styles can be applied to different types of lists and list items. What do you notice?



Run ►

open in  replit

index.html



```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5    <meta charset="utf-8">
6    <meta name="viewport" content="width=device-widt
7    <title>Lists: Styled</title>
8    <link href="style.css" rel="stylesheet" type="te
9  </head>
10
11 <body>
12
13
14   <ul>
15     <li>First Item</li>
16     <li>Second Item</li>
17   </ul>
18
```

<https://Lists-Styled--thinkful.repl.co>

Outline

Console

Shell

Embedded lists

Sometimes, however, you might have a list item that has additional list items—a list within a list. Fortunately, HTML lets you nest lists inside one another.

When you nest lists in HTML, the inner list, which is the sublist, will be indented inside the outer list. An indented sublist uses a circle icon, but with a little extra customization, you can create unique embedded lists with unique icons. Take a look at the code for these embedded lists below. Can you track where and how the sublists begin?

```
<h1>Lessons to Write Today</h1>
<ol>
  <li>Lesson Introduction</li>
  <li>HTML Lists
    <ul>
      <li>Unordered</li>
      <li>Ordered</li>
    </ul>
  </li>
  <li>Code Comments
    <ul>
      <li>HTML</li>
      <li>CSS</li>
    </ul>
  </li>
  <li>Link Breaks</li>
  <li>RollOvers</li>
</ol>
```

Outline


Changing the position

You also have the ability to shift the positioning of the list using the `list-style-position` property. Specifically, this allows you to move bullets or numbers inside or outside (which is the default) of the list-item *container* (you'll learn more about containers in subsequent checkpoints). This technique can be particularly useful if you're assigning a background color to a list and moving the bullets within the container.

Demo: List item positions

Run the Repl.it below to see the `list-style-position` property in action. What do you notice?

Run ▶

open in  repl.it

index.html

1

<!DOCTYPE html>

2

<html>

3

4

<head>

5

<meta charset="utf-8">

6

<meta name="viewport" content="width=device-widt

7

<title>Lists: Style Position</title>

8

<link href="style.css" rel="stylesheet" type="te

9

</head>

10

11

<body>

12

13

<h1>Outside (Default)</h1>

14

15

Item One

16

Item Two

17

Item Three

18

↻


<https://Lists-Style-Position--thinkful.repl.co>

🔗

Console

Shell

Connected!



Page dividers and horizontal rules

You're becoming an HTML styling pro! Now, you're ready to take on horizontal rules.

Page dividers allow you to organize web page content by creating a separation between distinct blocks of content. In HTML, these dividers are called *horizontal rules* (sometimes called *rulers*). You'll see these written into the HTML as `<hr>`. Horizontal rules are self-closing; in other words, there is no closing tag.

Demo: Horizontal rules

To get started, play around in the REPL below. Can you identify the horizontal rule in the code?





Styling horizontal rules

The horizontal rule in the REPL above was fairly simple, and getting fancy, horizontal rules are pretty simple by default. But with a little CSS, they

can be styled to display in better, more compelling ways. Here are some basic CSS styles for horizontal rules:

- `height`: This makes the border a certain height, in pixels (`px`).
- `background-color`: This makes the ruler a certain color inside the border.
- `border`: This requires three values to change the style of the border: the size in pixels, the color, and the stroke.
- `margin-top`: This adds empty space above the line.
- `margin-bottom`: This adds empty space below the line.

Here are a few fun examples for you to review and play around with:

- [Horizontal Ruler with Style](#)
- [Horizontal Ruler with Diamond Separator](#)
- [Horizontal Ruler with Graphic](#)

If you wish to see some creative and vibrant graphic rulers, check out these examples at [Smashing Magazine from 2008](#).

Text-formatting elements

There are 10 formatting elements in HTML that provide a default visual style to HTML text. These all change the visual style and formatting, but some also add meaning to the content and code, which can be useful for search engines. This is called *semantic coding*, which you'll learn more about later. For now, it's just important to know that it relies

logical descriptive terms to make it easier for search engines (and people) to read and understand what the code is doing.

- ``: Sets the text in **bold**.
- ``: Sets the text in **bold** and is semantically important.
- `<i>`: Sets the text in *italics*.
- ``: Sets the text in *italics* and is semantically important.
- `<mark>`: Sets the text as highlighted.
- `<small>`: Sets the text as smaller than the rest of the element.
- ``: Sets the text to display as ~~crossed out~~.
- `<ins>`: Sets the text to display as inserted by adding an underline to the text.
- `<sub>`: Sets the text as subscript , which is smaller and a bit below the other text.
- `<sup>`: Sets the text as ^{superscript} , which is smaller and a bit above the other text.

Demo: Text formatting links

Check out the code sample in the Repl.it below. What happens when you remove or change the styling of the text?





Line break

At times, you may want to create a line break in your web page. A line break in a line of text can be created using `
`. Like horizontal rules, *line*

breaks are self-closing—you only need to add the single opening tag: `
`. It's worth noting that in older versions of HTML, like HTML4, line breaks were written with a forward slash, `</br>`. You may see this from time to time in your online research.

However, this HTML element should be used *only* to make line breaks. Do not use `
` to separate paragraphs of text or to create space between HTML elements. If you need to create space between text or images in CSS, you should work with margins or padding. But don't worry; you'll learn more about this in upcoming checkpoints.

Styling links

Think you have what it takes to style your links? With all your new knowledge, you're ready to explore link styling options.

Outline

Demo: Text link rollovers

The code sample in the Repl.it below reveals a new concept: link rollovers. Take a few minutes to review the code, and try to answer the following questions on your own.

- What does `a:link` do?
- What does `a:visited` do?
- What does `a:hover` do?
- What does `a:active` do?
- Does the order of that code—link, visited, etc.—make a difference? How do you know?





Pseudo-classes

Okay, you're ready for the next piece of the puzzle. The interactive links of HTML are referred to as a *pseudo-class*. Pseudo-classes are useful in

changing the state of an element when the user engages with it, like in these situations:

- When an element, like a text link, has the cursor *roll over* (or *hover over*) it.
- When an element, like a text link, has the cursor *click on* it.

For anchor elements, which you've already learned a bit about, there are four pseudo-class selectors:

- `a:link`: This is the normal state of a text link.
- `a:visited`: This is the state of a text link that has already been visited by the current web browser.
- `a:hover`: This creates the rollover state for the element, which is triggered when the user's cursor rolls over, or hovers over, a link.
- `a:active`: This affects the state of the link when the user is actively clicking on it.

Due to the cascading aspect of CSS, the order of these pseudo-classes is very important. They must be written on CSS page in the specific sequence outlined below:

1. `a:link`
2. `a:visited`
3. `a:hover`
4. `a:active`



A clever way to help you remember the order of these pseudo-class selectors is this mnemonic device: **L O V E H A** TE. The order of the letters should help remind you of this order: *L* for link, *V* for visited, *H* for hover, and *A* for active.

Rounded corners and circular images

Now, you'll take a moment to start exploring image styling techniques. For instance, using the `border-radius` property, you'll be able to round the corners of any image (or container). Play around with the REPL below to see how this works.





But what if you continued to round those corners? By setting the `border-radius` to `50%` on all corners, you'll end up displaying a perfectly circular image, without the need for graphic design software. But in order to create a perfectly *circular* image, the original image must

be a perfect square. If the image is a rectangle, the resulting circular shape will look more like an oval.

Practice changing up the `border-radius` values for the images in the Repl.it below.





Assignment

In this assignment, you'll practice applying these new skills. Use the code sample in the Repl.it below to get started, update all the content

using appropriate page text and lists. Then, style the content as much as you wish using the following CSS styles.

Taking the time to practice writing HTML code and styling it with CSS will help you get comfortable adding all these elements to future projects. In your work, incorporate the following:

- A unique page title
- Text headings
- Paragraphs
- Unordered lists
- Ordered lists
- Horizontal rules
- Code comments

Outline

When you've finished, submit a link to your Repl.it in the box below. Then, share the Repl.it links with your classmates. Dig into each other's code, and suggest ways that they could update or improve their code.





When you have completed your practice project, feel free to compare your code with this completed one.

Practice Project #1 Complete



Checkpoint

Submit your ideas or a link to your work here and use it as a conversation starter during your next mentor session.

This checkpoint will not be graded, but is still required.

Your work

03.08.21



Share your ideas here...

bold *italic* ``code`` > quote - bullet list

Preview

Completed

Next checkpoint



How would you rate this content?

[Report a typo or other issue](#)

[Go to Overview](#)

Outline

