



# **Learn CSS In One Day and Learn It Well**

**Accompanying Resource  
(End of Chapter Exercises)**

<http://www.learncodingfast.com>

Copyright © 2015

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

**Note: If you copy and paste any code in this document, you may have to retype the quotation marks (") yourself. Quotation marks tend to fail when copy-and-pasting.**

## Exercise 3

Download the source code for this exercise from <http://learncodingfast.com/css> and unzip the file. The source code for this exercise can be found in the *Chapter 3 - Basics of CSS* folder.

### Exercise 3.1

1. Open the file *Chapter 3 - Basics of CSS.html* concurrently in your browser and text editor.
2. First, look for the following lines in the source code in your text editor:

```
p {  
    background-color: yellow;  
}
```

This selects all the `<p>` elements and sets their background colors to yellow. The line 'This is some text in the div element.' is not selected because it is not within any `<p> . . . </p>` tags.

3. Now let's try changing the rule from

```
p {  
    background-color: yellow;  
}  
  
to  
  
P {  
    Background-color: YELLOW;  
}
```

Save the file in your text editor and refresh your browser. Notice that nothing changes? This is because CSS is not case-sensitive in most cases.

4. Now, let us try to select different HTML elements and observe which elements end up with a yellow background. For each item below, simply change the selector on line 6 in the HTML file to the required selector.

First, let's select the element with `class = "myClassPara"`. To do that, change the `p` selector in the CSS rule to `.myClassPara`. Save the file in the editor and refresh the page

in the browser. Notice which paragraph is selected now.

5. Now change the selector to `.myclasspara`. Notice that nothing is selected now? That is because CSS is case-sensitive when selecting classes and ids.
6. Next, let's select the element with `id = "myIDPara"`. Try doing it yourself.

Got it? You can change `.myClassPara` to either `p#myIDPara` or just `#myIDPara`. Notice which paragraph is selected.

Try changing `#myIDPara` to `#MYIDPARA`. Notice that nothing is selected?

7. Next, let's learn how to select more than one elements. Try selecting both the `h1` and `h2` elements.

The way to do it is simply to change the selector to `h1, h2`.

8. Next try selecting the `div` element.

To do it, simply change the selector to `div`.

9. Now, let's select the `p` element inside the `div` element. To do this, we write `div p` as the selector. Notice which elements are selected.

10. Next, try selecting all the link (`<a>`) elements.

What do you notice? The links are now highlighted in yellow right?

11. Next, we'll narrow down our selection based on HTML attributes. Try selecting the link with `href="http://www.learncodingfast.com"`. The correct way to do this is with square brackets as follows:

```
a[href="http://www.learncodingfast.com"]
```

Try it. Only the first link will have a yellow background now.

12. Next, we'll use the pseudo-class selector to change the background color of all link elements when we hover over them. Try changing

```
a[href="http://www.learncodingfast.com"]
```

to

```
a:hover
```

Save the file and refresh the browser. Notice that nothing is selected? Now hover your mouse over any of the hyperlinks and observe what happens.

13. Next, let's try to select the second child element of the `div` element. You do that by changing the selector to

```
p:nth-child(2)
```

14. Now, let's try selecting the first letter of all `<p>` elements. You use the pseudo-element `first-letter` to do that. Change the selector to

```
p::first-letter
```

15. Next, let's look at what happens when an element has more than one classes. Change the selector back to `.myClassPara` and add the following CSS code just before the `</style>` tag.

```
.mySecondClassPara {  
    text-decoration: underline;  
}
```

Notice which paragraph is both yellow in background AND underlined. This is because that paragraph has two classes: `myClassPara` and `mySecondClassPara`. Therefore, both rules apply to it.

16. Finally, let's try adding an exclamation mark to the end of all `<p>` elements. We'll use the `after` pseudo-element to do that. Add the following CSS code just before the `</style>` tag.

```
p::after{  
    content: "!";  
}
```

## Exercise 4

Download the source code for this exercise from <http://learncodingfast.com/css> and unzip the file. The source code for this exercise can be found in the *Chapter 4 - CSS Box Model* folder.

### Exercise 4.1

1. Open the file *Chapter 4 - CSS Box Model.html* concurrently in your browser and text editor.
2. Resize your browser window and observe how the gibberish text flows around the bigger box.
3. Modify the CSS declaration for `box1` by changing `width: 100px;` and `height: 100px;` to each of the following:

- (a) `width: 200px; height: 200px;`
- (b) `width: 60%;`
- (c) `height: auto; width: auto;`

Save the file in the text editor and refresh the page in the browser. Notice what happens to `box1` in each case.

4. Now change the width and height of `box1` back to 100px and change the text between the tags `<div id="box1">...</div>` to:

*“Learn CSS in One Day and Learn It Well. This example shows what happens when text overflows the dimension of the box.”*

Next, add each of the following to the CSS declaration of `box1` and notice what happens in each case:

- (a) `overflow: visible;`
- (b) `overflow: hidden;`
- (c) `overflow: scroll;`
- (d) `overflow: auto;`

5. Modify the CSS declaration for `box1` by changing `margin: 20px;` to each of the following and notice what happens to the space around `box1` in each case:

- (a) `margin: 50px;`
- (b) `margin: -50px;`
- (c) `margin: 25px 200px;`
- (d) `margin: 25px 50px 60px 10px;`

6. Remove the margin rules for `box1` and add the 4 lines below:

```
margin-top: 25px;
margin-right: 50px;
margin-bottom: 60px;
margin-left: 10px;
```

Notice any changes to box1? There should be no change as this is the same as 5(d) above.

7. Remove the margin rules for box1 and add the line below:

```
margin: 0 auto;
```

Next, change the width of box1 to 80%. Finally, remove the line `float: left;`. Notice what happens. (We'll explain what `float` does in the next chapter). box1 is now center-aligned.

8. Now, let's work on box2. Modify the CSS declaration for box2 by changing `padding: 50px;` to each of the following and notice what happens inside box2.

- (a) `padding: 25px;`
- (b) `padding: -10px;` (refer to note below)
- (c) `padding: 25px 50px;`
- (d) `padding: 25px 50px 60px 10px;`

8(b) will not work as paddings cannot have negative values. You'll just end up with a padding of 0 pixel.

9. Remove the padding rules for box2 and add the 4 lines below:

```
padding-top: 25px;
padding-right: 50px;
padding-bottom: 60px;
padding-left: 10px;
```

Notice any changes to box2? There should be no change as this is the same as 8(d) above.

10. Now let's change the border style for box2. Remove the line `border: 5px solid black;` from the CSS declaration of box2 and add each of the following. Notice what happens to the border in each case:

- (a) `border-style: solid dotted;`
- (b) `border-style: none dashed double groove;`
- (c) `border-style: inset;`
- (d) `border-style: outset;`

11. Next, remove the `border-style` rule for box2 and add the following:

```
border-top-style: dotted;
border-left-style: double;
```

12. Now remove the rules from part 11 and change border style to solid (`border-style: solid;`)

Next add each of the following rules. Notice what happens to `box2` in each case.

- (a) `border-width: 25px;`
- (b) `border-width: 25px thin;`
- (c) `border-top-width: 30px;`

13. Now we are going to change the border color for `box2`. Try adding each of the following rules:

- (a) `border-color: rgb(255, 0, 0);`
- (b) `border-color: red green;`
- (c) `border-top-color: #12FF5F;`

14. Next, we'll try the border shorthand. Remove all the border properties for `box2`. Add the following:

```
border: 5px solid green;
```

15. Now, let's try the `border-radius` property. Try each of the following:

- (a) `border-radius: 20px;`
- (b) `border-radius: 10px 20px;`
- (c) `border-radius: 25px 5px 0 50px;`
- (d) `border-top-left-radius: 10px;`

16. Finally, let's try to create a circle. First, change the width and height of `box2` to 100px, padding to 20px and the property

```
border: 5px solid green;
```

to

```
border: 50px solid green;.
```

Now, remove all previous rules for border radius and add the following rule:

```
border-radius: 120px;
```

You end up with a circle right? That's because the border radius is half of the total height (and width) of `box2`.

## Exercise 5

Download the source code for this exercise from <http://learncodingfast.com/css> and unzip the file. The source code for this exercise can be found in the *Chapter 5 - Positioning and Floating* folder.

### Exercise 5.1

1. Open the file *Chapter 5 - Positioning.html* concurrently in your browser and text editor.
2. Add the rule `position: static;` to the CSS declaration for `#box1` and `#box2` and observe what happens to the positions of the two boxes.

Nothing changes right? That's because all HTML elements are positioned using `static` positioning by default.

3. Remove the rule `position: static;` from the CSS declaration for `#box1` and `#box2`. Add the following lines to the CSS declaration of `#box2`.

```
position: relative;
top: 50px;
left: 70px;
```

Observe what happens to the position of the second box. It is now 50px from the top and 70px from the left of its normal position.

Play around with different values for the `top` and `left` properties and see what happens to the position of `box2`. Try negative values as well. You can also try using the `right` and `bottom` properties.

4. Now change the CSS declaration of `#box2` back to

```
position: relative;
top: 50px;
left: 70px;
```

Notice that the text is partially hidden by the red box?

If you want the text to appear in front of the red box, add the following lines to the CSS declaration of `p`.

```
position: relative;
z-index: 2;
```

You can set the position to either `relative`, `absolute` or `fixed`. It does not matter as long as it is not `static`.



Next, add the line `z-index: 1;` to the CSS declaration of `#box2`. Refresh the page. The text should now be in front of the red box since the `z-index` of the text is greater than the `z-index` of the red box..

If you want the red box to be in front of the text instead, just change the `z-index` of `#box2` to a value greater than 2.

5. Now, change the height of `#box1` to `5000px` and the positioning of `#box2` to `fixed` (`position: fixed;`).

Scroll the page. The red box does not move as it uses fixed positioning now.

6. Change the positioning of `#box2` back to `relative` and the height of `#box1` back to `100px`. Add the following line just before the `<p>` tag in the HTML code at the bottom of the file.

```
<div id="box3"></div>
```

Now add the following CSS declaration to style `box3`.

```
#box3 {  
    position: absolute;  
    top: 50px;  
    left: 150px;  
    background-color: yellow;  
    width: 50px;  
    height: 50px;  
    padding: 20px;  
    border: 5px dotted black;  
    margin: 10px;  
}
```

`box3` uses `absolute` positioning and since it is not a child element of any `<div>` elements, it is positioned relative to the page. That is, it is positioned `50px` from the top of the page and `150px` from the left.

7. Now change the `<div>` code for the three boxes to the following:

```
<div id="box1"></div>  
<div id="box2">  
    <div id="box3"></div>  
</div>
```

`box3` is now a child element of `box2`. Therefore it is positioned `50px` down from the top and `150px` right from the left side of `box2` as it is now positioned relative to `box2`.

## Exercise 5.2

1. Open the file *Chapter 5 - Floating.html* concurrently in your browser and text editor.

2. Resize your browser-window to see what happens when the `<div>` elements do not have enough room.
3. Modify the CSS declaration for `div` by changing `float: left;` to `float: right;`. Observe what happens to the positions of the boxes.
4. Now change the rule back to `float: left;` and add the code below to the HTML code, just above the `</body>` tag.

```
<p>This is some text that is not floated.</p>
<p>This is more text that is not floated.</p>
<p>This is yet more text that is not floated.</p>
```

Refresh the page and observe where the text is. Resize the browser window to see what happens when the width of the page is narrow.

5. Next, add the code

```
.clearFloat { clear: both; }
```

to the CSS declaration just above the `</style>` tag.

Change the first `<p>` tag to `<p class="clearFloat">` and observe what happens.

## Exercise 6

Download the source code for this exercise from <http://learncodingfast.com/css> and unzip the file. The source code for this exercise can be found in the *Chapter 6 - Display and Visibility* folder.

### Exercise 6.1

1. Open the file *Chapter 6 - Display and Visibility.html* concurrently in your browser and text editor.
2. Change the height and width in the CSS declaration for `#displaydemo`. Observe what happens to the yellow box.

Nothing changes right? The yellow box is declared as an inline element. Hence it will only take up as much width and height as it needs. In fact, you can remove the height and width properties and nothing will change.

3. Now change the display property of `#displaydemo` from `display: inline;` to `display: block;`. What happens?

The yellow box is now displayed as a block element. It starts and ends with a new line. If you have declared its width and height, the width and height will be the values you specified. If you have not specified its width and height, the yellow box will occupy the full width of the page. It will take up as much height as it needs.

4. Try changing the height and width in the CSS declaration for `#displaydemo`. Observe what happens.
5. Now, let's move on to the `visibility` property. Change the CSS declaration of `#magic` from `display: inline;` to `visibility: hidden;`. Observe what happens to the word 'magic' in the blue sentence.
6. Next, change the CSS declaration of `#magic` from `visibility: hidden;` to `display: none;`. Observe what happens.

## Exercise 7

Download the source code for this exercise from <http://learncodingfast.com/css> and unzip the file. The source code for this exercise can be found in the *Chapter 7 - Background* folder.

### Exercise 7.1

1. Open the file *Chapter 7 - Background.html* concurrently in your browser and text editor.
2. Scroll the page in your browser window. The *Learn Coding Fast* logo is set as the background for the `body` element. Notice what happens to the logo as you scroll.

Now in the CSS declaration for `body`, change `background-attachment: fixed;` to `background-attachment: scroll;`. Refresh and scroll the page. Notice what happens.

3. Next, change `background-repeat: repeat-x;` in the `body` CSS rules to each of the following and notice what happens to the background logo.

- (a) `background-repeat: repeat;`
- (b) `background-repeat: no-repeat;`
- (c) `background-repeat: repeat-y;`

Notice what happens in each case.

4. Next, change `background-color: white;` to `background-color: #0000FF;`. This should change the background color of the page to blue.
5. Finally, we are going to look at the `background-position` property. We'll be modifying the rules for `#box1` in the examples that follow. The image used in this example is 'backgroundposition.png' stored in the same folder. Have a look at the image before proceeding.

Notice that when `background-position` is set to `0 0` (the default), area A is displayed. Changing the `background-position` property will shift the background image with reference to area A.

In the source code CSS declaration for `#box1`, change `background-position: 0 0;` to

- (a) `background-position: -100px 0;`

This shifts the background image 100px to the left. As the default area shown is Area A, after shifting 100px to the left, you should get Area B.

- (b) `background-position: 100px 0;`

This shifts the background image 100px to the right. As the `background-repeat` property is set to `no-repeat`, no image will be displayed. This is because the background image has shifted out of the boundaries of `box1`.

(c) `background-position: 0 -100px;`

This shifts the background image 100px up. You should get Area C.

(d) `background-position: 0 100px;`

This shifts the background image 100px down. No image will be displayed as the background image has shifted out of the boundaries of `box1`.

(e) `background-position: -100px -100px;`

Try to figure out which area you'll get.

## Exercise 8

Download the source code for this exercise from <http://learncodingfast.com/css> and unzip the file. The source code for this exercise can be found in the *Chapter 8 - Font and Text* folder.

### Exercise 8.1

1. Open the file *Chapter 8 - Font and Text.html* concurrently in your browser and text editor.
2. Modify the CSS property `font-family` and observe what happens to the sample text. The current font family is Sans Serif. Try each of the following:
  - (a) `font-family: Verdana, Arial, Helvetica, sans-serif;`
  - (b) `font-family: Courier, "Lucida Console", monospace;`
3. Modify the CSS property `font-size` and observe what happens to the sample text in each of the following cases. Try
  - (a) `font-size: 40px;`
  - (b) `font-size: 1.5em;`
  - (c) `font-size: x-small;`
  - (d) `font-size: 120%;`
4. Modify the CSS property `font-style` and observe what happens to the sample text. Try
  - (a) `font-style: italic;`
5. Modify the CSS property `font-weight` and observe what happens to the sample text. Try
  - (a) `font-weight: bold;`
  - (b) `font-weight: 300;`
6. Modify the CSS property `color` and observe what happens to the sample text. Try
  - (a) `color: blue;`
  - (b) `color: #00ff00;`
  - (c) `color: rgb(255,0,0);`
7. Modify the CSS property `text-align` and observe what happens to the sample text. By default, the text in this paragraph is left aligned. Now try each of the following:
  - (a) `text-align: justify;`
  - (b) `text-align: right;`
  - (c) `text-align: center;`

8. Modify the CSS property `text-decoration` and observe what happens to the sample text. Try

- (a) `text-decoration: underline;`
- (b) `text-decoration: overline;`
- (c) `text-decoration: line-through;`

9. Modify the CSS property `letter-spacing` and observe what happens to the sample text. Try

- (a) `letter-spacing: 5px;`
- (b) `letter-spacing: -5px;`

10. Change `letter-spacing` back to `0px`. Modify the CSS property `word-spacing` and observe what happens to the sample text. Try

- (a) `word-spacing: 10px;`
- (b) `word-spacing: -20px;`

11. Change `word-spacing` back to `0px`. Modify the CSS property `line-height` and observe what happens to the sample text. Try

- (a) `line-height: 2;`
- (b) `line-height: 25px;`
- (c) `line-height: 30%;`

## Exercise 9

Download the source code for this exercise from <http://learncodingfast.com/css> and unzip the file. The source code for this exercise can be found in the *Chapter 9 - Lists, Links and Navigation Bars* folder.

### Exercise 9.1

1. Open the file *Chapter 9 - List and Links.html* concurrently in your browser and text editor.
2. Modify the CSS declaration for `ol` and observe what happens to the 'Cars Ordered List'. Try each of the following:

- (a) `list-style-type: decimal-leading-zero;`
- (b) `list-style-type: lower-roman;`
- (c) `list-style-type: upper-roman;`
- (d) `list-style-type: upper-alpha;`
- (e) `list-style-type: lower-alpha;`
- (f) `list-style-type: lower-greek;`
- (g) `list-style-type: none;`

3. Modify the CSS declaration for `ul` and observe what happens to the 'Cars Unordered List'. Try each of the following:

- (a) `list-style-type: circle;`
- (b) `list-style-type: square;`
- (c) `list-style-type: none;`
- (d) `list-style-type: disc;`

4. Add the following rule to the CSS declaration for `ul`:

```
list-style-position: outside;
```

Refresh the browser. Nothing changes right? That is because `list-style-position: outside;` is the default. Now change `list-style-position: outside;` to `list-style-position: inside;`. You'll notice that the unordered list is shifted slightly to the right (i.e. there's more indentation).

5. Remove the `list-style-type` property for `ul` and add the following instead:

- (a) `list-style-image: url("myMarker.gif");`

Observe what happens to the 'Cars Unordered List'.

6. Now let's look at how to style hyperlinks. Try adding the following rules to the stated selector:



(a) Add `color: green;` to `a:link{}`  
and observe what happens to the 'Click Me' link.

(b) Add `font-size: 2em;` to `a:active{}`  
and click on the link.

(c) Add `background-color: red;` to `a:hover{}`  
and hover your mouse over the link.

## Exercise 9.2

1. Open the file *Chapter 9 - Navigation Bar.html* concurrently in your browser and text editor.
2. This page shows an example of how a horizontal navigation bar can be created using CSS rules for lists and links. The steps are explained in the book. Study the code and try modifying the CSS declaration to improve the design of this navigation bar.

## Exercise 10

Download the source code for this exercise from <http://learncodingfast.com/css> and unzip the file. The source code for this exercise can be found in the *Chapter 10 - Tables* folder.

### Exercise 10.1

1. Open the file *Chapter 10 - Tables.html* concurrently in your browser and text editor.
2. Modify the CSS declaration for `table` and observe what happens to the table. Try adding the following and observe what happens in each case:

- (a) `padding: 50px;`
- (b) `margin: 30px;`
- (c) `border-collapse: collapse;`
- (d) Change `border: dashed 1px black;` to `border: solid 2px green;`

3. Modify the CSS declaration for `th` and observe what happens to the table. Try adding:

- (a) `padding: 50px;`
- (b) `margin: 30px;` (Margin rules will be ignored. Try it)

Repeat the same for `td`.

4. Remove the `padding` and `margin` properties from Part 3 for both `td` and `th`. Modify the CSS declaration for `table` and observe what happens to the table. Try adding each of the following:

- (a) `width: 300px;`

- (b) `width: 50%;`

Notice that the width of the table is now 50% of the red box.

- (c) `height: 500px;`

- (d) `height: 80%;`

Notice that the height of the table is now 80% of the red box.

5. Remove the `height` and `width` properties for `table` from Part 4. Modify the CSS declaration for `th` and observe what happens to the table. Try changing the `height` property to:

- (a) `height: 100px;`

Remove the `height` property from `th` and add it to `td`. Notice what happens. Repeat the

same for `tr`.

6. Modify the CSS declaration for `th` and observe what happens to the table. Try changing the `width` property to:

(a) `width: 100px;`

Remove the `width` property from `th` and add it to `td`. Notice what happens. There should be no difference as whether you set the width of a table cell at the `th` or `td` level, that width will affect both elements.

7. Add the attributes `id="firstColumn"`, `id="secondColumn"` and `id="thirdColumn"` to the first, second and third `<th>` start tags respectively. Next, add the following CSS rules to adjust the width of the three columns separately:

```
#firstColumn {
    width: 100px;
}

#secondColumn {
    width: 200px;
}

#thirdColumn {
    width: 50px;
}
```

Refresh the page and observe what happens. Try adjusting the width with different values.

8. Modify the CSS declaration for `tr` and observe what happens to the table. Try adding:

(a) `text-align: center;`  
(b) `vertical-align: top;`

Remove the above properties from `tr` and add them to `th`. Notice what happens. Repeat the same for `td`. If you do not see any difference, try increasing the width of the table and/or the height of the rows.

9. Modify the CSS declaration for `tr` and observe what happens to the table. Try adding:

(a) `background-color: green;`  
(b) `color: white;`

Remove the above properties from `tr` and add them to `th`. Notice what happens.

Now, remove the properties from `th` and add them to `td`.

10. Finally, let's try coloring alternating rows. Remove the `background-color` and `color` rules from `td` first. Next, add the following code and observe what happens.

```
tr:nth-child(even) {  
    background-color: lightgreen;  
}
```

```
tr:nth-child(odd) {  
    background-color: lightgray;  
}
```

# Thank You

Before ending this tutorial, I would like to thank you once again for buying [my book](#). I sincerely hope the book has been useful to you and has helped you understand CSS better. If you enjoyed reading the book, I would appreciate it if you would take a few seconds to leave a review on Amazon. This will help with my ranking and help other readers find the book. Even a one sentence review would help. [You can click here to leave your review](#).

Thank you and as usual, if you have any problems or feedback, you can email me at [jamie@learncodingfast.com](mailto:jamie@learncodingfast.com).