Learn CSS In One Day and Learn It Well

Accompanying Resource

(Bonus Project)

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.

CSS Hands-On Project

Welcome to the CSS Hands-On Project guide where you'll be guided through the coding of a webpage from scratch, using HTML and CSS. The demo for this website can be found at http://learncodingfast.com/demo/jetspeed.html.

As you can see, the webpage employs a simple and elegant minimalistic design, with nothing too fanciful. The main reason for this is because I am a coder and not a desginer, so my graphics design skills at limited at best. Nonetheless, this page demonstrates all the important CSS concepts that we covered in my book *Learn CSS in One Day and Learn it Well*. We'll learn how to use background images, style borders, change font size, float and position boxes, create and style tables and more.

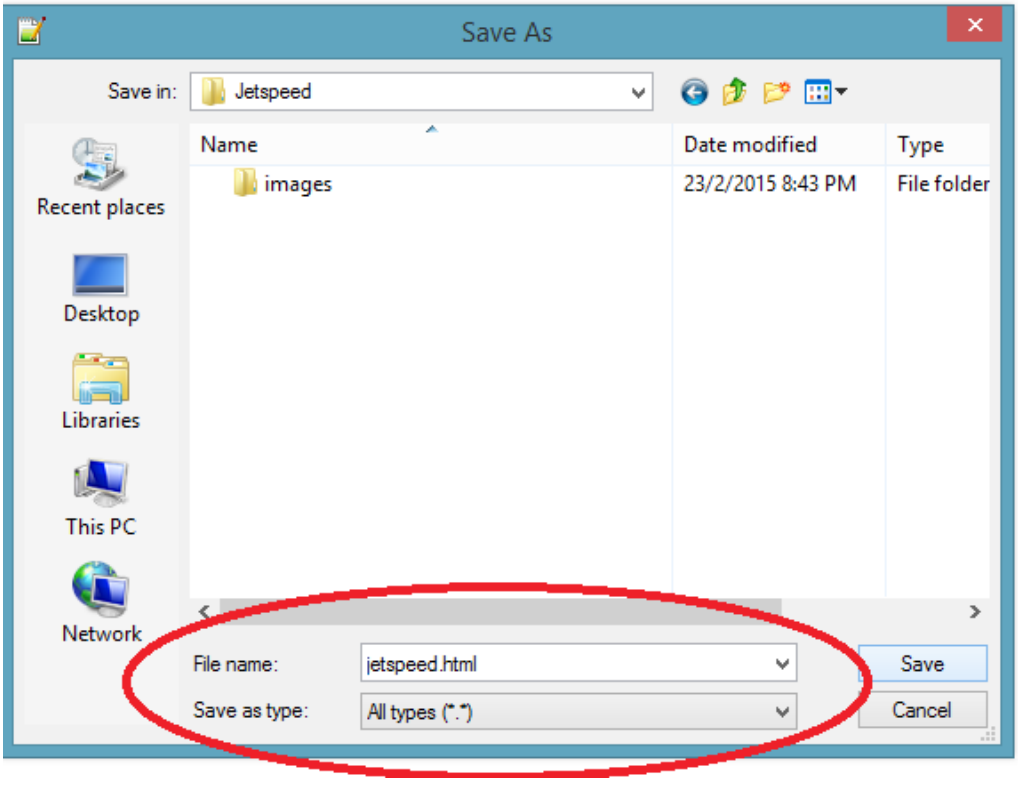
Ready to get started? Let's move on.

Chapter 1: The HTML Code

We'll be coding the homepage for a fictitious travel agency ***Jetspeed Travel***. The tagline of the agency is ***Roam the World in Style***.

First, let's start by creating a folder called “Jetspeed” on our desktop. Inside the folder, create another folder called “images”. Copy all the dowloaded image files (*china.jpg*, *japan.jpg*, *malaysia.jpg*, *blacktile.jpg* and *logo.png)* to the “images” folder.

Now let's create a .html file and save it in the “Jetspeed” folder. I'll assume you are using Notepad++ as your text-editor. Open Notepad++ and select *File>New*. Then select *File>Save As...* Save this blank file as *jetspeed.html* in the “Jetspeed” folder. Make sure you select “All Type (*.*)” for “Save As Type” (refer to diagram below).



Now, we are going to code the html file. We'll start by writing the usual HTML tags (refer to Chapter 2 in the book for a description of these tags). Add the following code to *jetspeed.html*.

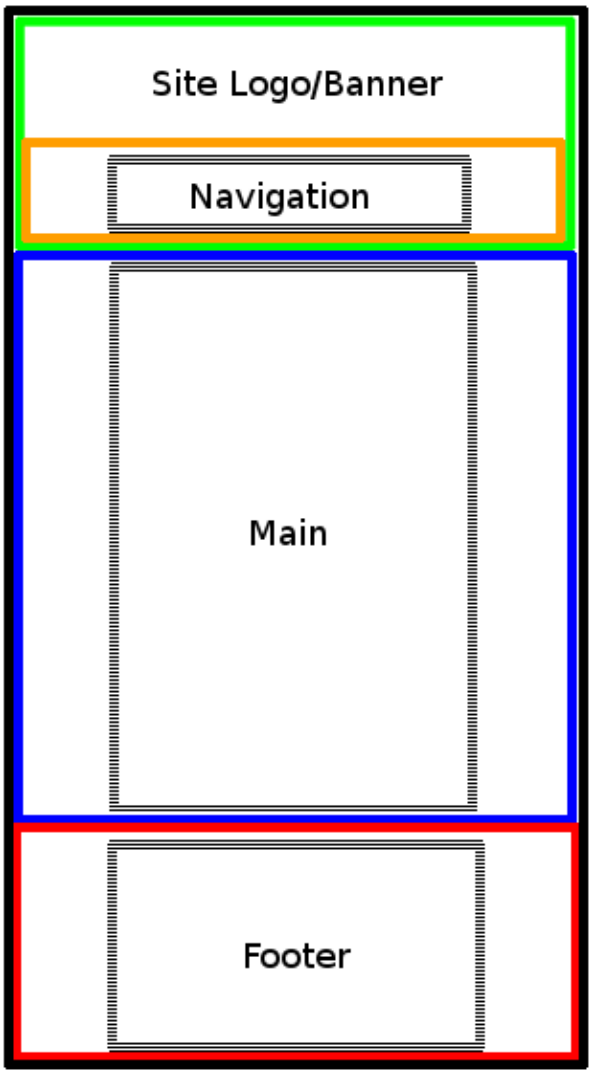
```
<!doctype html>
<html>
<head><title>JETSPEED TRAVEL | Roam the World in Style</title>
<meta name="viewport" content="width=device-width">
```

```
<style type="text/css">
      /* Our CSS Codes will be added here*/
</style></head>
<body>
      <!--This is where the bulk of the HTML codes will be added-->
</body>
</html>
```

Note: If you copy and paste the code above, you may have to retype the quotation marks (“) yourself. Quotation marks tend to fail when copy-and-pasting.

Our CSS code will be inserted between the `<style type="text/css">...</style>` tags and our HTML code will be inserted between the `<body>...</body>` tags.

Now, let us look at the `<body>` tag in detail. The image below shows the structure of the `body` element.



Exercise 1: The main structure

The black border represents the `<body>` tag.

The green border represents the `<header>` element. Inside the `<header>` element, we have the site logo (an image file named “logo.png” in the “images” folder) and the `<nav>` tag. The `<nav>` tag is represented by the orange box. Inside the `<nav>` element, we have a `<div>` tag with `class=”container”`, represented by the black dotted box.

The “container” class is added to give us greater flexibility in adjusting the width of the main content. This class is used for the `<nav>`, `<main>` and `<footer>` elements so that all three of them have a consistent width.

Next, let us look at the `<main>` and `<footer>` tags. These two tags are relatively less complicated compared to the `<header>` tag. The blue box represents the `<main>` tag and the red box represents the `<footer>` tag. Inside both the `<main>` and `<footer>` elements, we have a `<div>` element (represented by the black dotted line) with `class = “container”`.

Try writing the HTML code for these elements. The answer is provided at the end of this chapter.

Exercise 2: Expanding the <nav> section

Now that we've done the HTML code for the main structure, let's work on the menu bar. In this exercise, we'll add the links for the navigation section. We'll use an unordered list for the links. There are six list items:

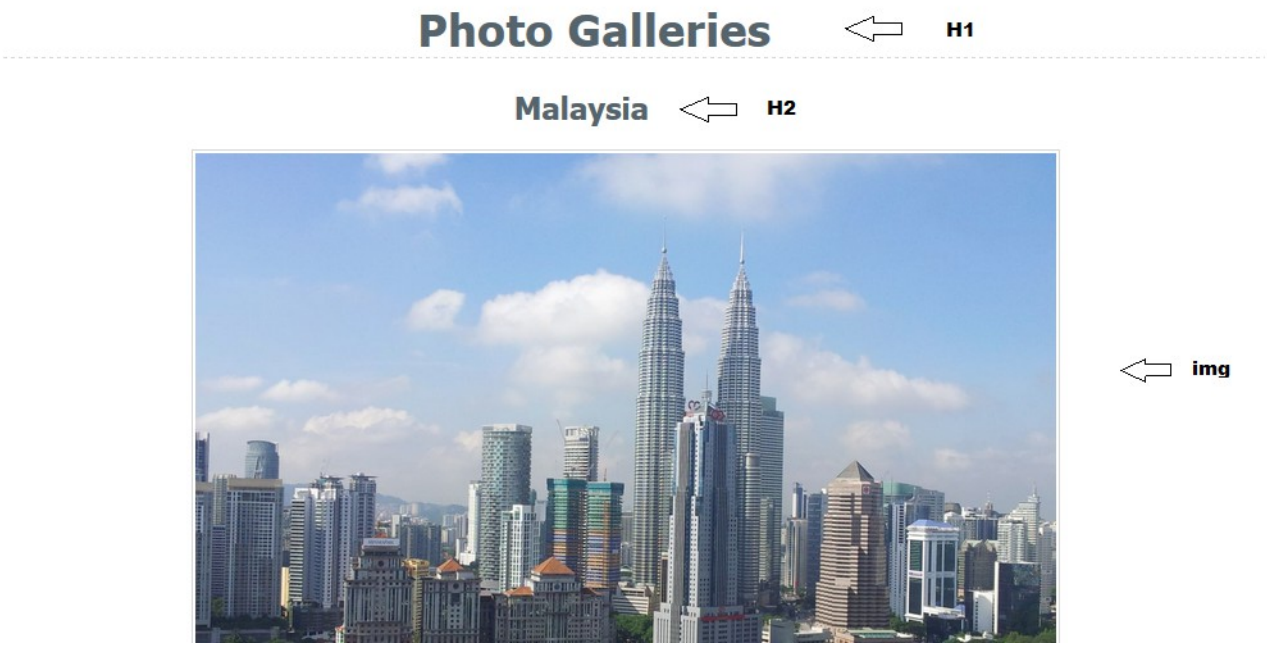
Home (points to index.html)
About Us (points to about.html)
Book Hotel (points to hotel.html)
Book Flight (points to flight.html)
Book Package (points to package.html) and
Travel Insurance (points to insurance.html)

Add this unordered list as a child element of the container `<div>` in the `<nav>` tag. Then convert each list item to a link using the `<a>` tag. You can refer to Chapter 9 in the book if you are stuck. As usual, the answer is provided at the end of this chapter.

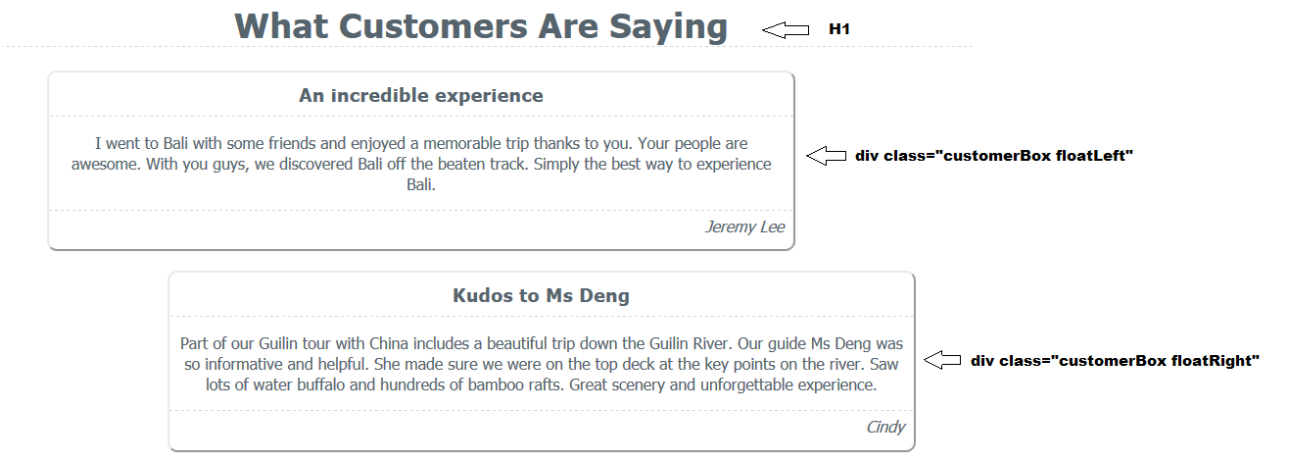
Exercise 3: Expanding the <main> section

In this exercise, we'll add the code for the `<main>` section. Everything in this exercise is a child element of the `<div class="container">` element in the `<main>` section.
The images below show the HTML tags used for each subsection.

The heading "Why Travel With Us?" uses a `<H1>` tag, while the three points below are individual paragraphs (`<p>` tags).



The heading "Photo Galleries" uses a `<H1>` tag, while the subheadings "Malaysia", "Japan" and "China" are `<H2>` tags. The three images used are "malaysia.jpg", "japan.jpg" and "china.jpg" stored in the "images' folder.



The heading “What Customers Are Saying” is a `<H1>` tag. The boxes below are `<div>` tags. Boxes that are aligned left belong to classes `”customerBox”` and “`floatLeft`”, while boxes that are aligned right belong to `”customerBox”` and “`floatRight`”.

Inside the box, the heading (e.g. An incredible experience) uses a <h1> tag, while the name of the customer (e.g. Jeremy Lee) is a <h2> tag. The main comment (e.g. I went to Bali....) uses a <p> tag.

Finally we have the *Upcoming Tour Dates* table. The heading uses a H1 tag as usual. The table is created using the `<table>` tag. The first row is a table heading (`<th>`) while the remaining rows are just `<td>`.

Try coding this section. Once you are done, the bulk of the HTML portion of our website is complete. What's left is the footer.

Exercise 4: Expanding the <footer> section

In this exercise, we'll add the code for the `<footer>` section. This section is very similar to the `<nav>` section. We'll use an unordered list for the links. There are five list items:

Contact Us (points to contact.html)
FAQ (points to faq.html)
Press Release (points to press.html)
Investor Relations (points to investor.html) and
Privacy Policy (points to privacy.html)

Add this unordered list as a child element of the `<div class=”container”>` element in the `<footer>` section. Then convert each list item to a link using the `<a>` tag.

Done?

Congrats! You have completed the HTML code. Compare your code with the answer given. Is it the same? If you try to load this web page on a browser, you'll see that it looks nothing like the demo site. Do not worry about it. This is because we have yet to write the CSS code. We'll do that in the next chapter.

Answers:

Exercise 1

```
<header>
      <img src=”images/logo.png”>
      <nav>
            <div class=”container”>
            </div>
      </nav>
</header>
<main>
      <div class=”container”>
      </div>
</main>
<footer>
      <div class=”container”>
```

```
    </div>
</footer>
```

Exercise 2

```
<ul>
    <li><a href="#">Home</a></li>
    <li><a href="about.html">About Us</a></li>
    <li><a href="hotel.html">Book Hotel</a></li>
    <li><a href="flight.html">Book Flight</a></li>
    <li><a href="package.html">Book Package</a></li>
    <li><a href="insurance.html">Travel Insurance</a></li>
</ul>
```

Exercise 3

```
<h1>Why Travel With Us?</h1>

<p>Awarded #1 Travel Agency for Families with Pre-school Kids</p>
<p>Accredited with National Travel Agencies Association</p>
<p>Awarded Best Tour Agency in 2013, 2014</p>

<h1>Photo Galleries</h1>

<h2>Malaysia</h2>
<img src="images/malaysia.jpg">
<h2>Japan</h2>
<img src="images/japan.jpg">
<h2>China</h2>
<img src="images/china.jpg">

<h1>What Customers Are Saying</h1>

<div class="customerBox floatLeft">
<h1>An incredible experience</h1>
<p>I went to Bali with some friends and enjoyed a memorable trip thanks
to you. Your people are awesome. With you guys, we discovered Bali off
the beaten track. Simply the best way to experience Bali.</p>
<h2>Jeremy Lee</h2>
```

```
</div>

<div class="customerBox floatRight">
<h1>Kudos to Ms Deng</h1>
<p>Part of our Guilin tour with China includes a beautiful trip down the Guilin River. Our guide Ms Deng was so informative and helpful. She made sure we were on the top deck at the key points on the river. Saw lots of water buffalo and hundreds of bamboo rafts. Great scenery and unforgettable experience.</p>
<h2>Cindy</h2>
</div>

<div class="customerBox floatLeft">
<h1>Great Food!!!</h1>
<p>Jetspeed Agency is one of the best, if not the best, travel agencies I've ever travelled with. Accommodation was top-notch, and the food is simply a highlight of the whole trip. From the crab buffet at Sapporo to the fine dining at Hakodate, it was worth every penny spent. </p>
<h2>Jovin Teo</h2>
</div>

<div class="customerBox floatRight">
<h1>You won't go wrong with Jetspeed</h1>
<p>I booked a food trip with Jetspeed and I am so glad I did. Our guide, Peter, brought us to all the local dining places where we got to a taste of authentic local food. From the mouth-watering Laksa to the legendary Chilli crab, we had such great fun.</p>
<h2>Simon Bond</h2>
</div>

<div class="customerBox floatLeft">
<h1>Excellent</h1>
<p>Our guide Yang is so knowledgeable and experienced. He goes out of the way to help us secure the best seats for all the performances, and also took good care of my grandma who was sick on the trip. Thanks, Yang. Look forward to travelling with Jetspeed again.</p>
<h2>Gail Scott</h2>
</div>
```

```
<h1>Upcoming Tour Dates</h1>

<table>
<tr><th>Tour Package</th><th>Dates</th><th>Status</th></tr>
<tr><td>8 Days 7 Night China Tour</td><td>1 Mar 15 to 8 Mar
15</td><td>Confirmed Departure, No Vacancies Left</td></tr>
<tr><td>10 Days 9 Night Africa Tour</td><td>5 Mar 15 to 14 Mar
15</td><td>Confirmed Departure, No Vacancies Left</td></tr>
<tr><td>8 Days 7 Night Greece Tour</td><td>15 Mar 15 to 22 Mar
15</td><td>Confirmed Departure, Vacancies Available</td></tr>
<tr><td>13 Days 12 Night Korea and Japan Tour</td><td>20 Mar 15 to 1 Apr
15</td><td>Pending</td></tr>
<tr><td>5 Days 4 Night Canada Tour</td><td>23 Mar 15 to 27 Mar
15</td><td>Confirmed Departure, Vacancies Available</td></tr>
</table>
```

Exercise 4

```
<ul>
      <li><a href="contact.html">Contact Us</a></li>
      <li><a href="faq.html">FAQ</a></li>
      <li><a href="press.html">Press Release</a></li>
      <li><a href="investor.html">Investor Relations</a></li>
      <li><a href="privacy.html">Privacy Policy</a></li>
</ul>
```

Putting It All Together

Check out the source code of *jetspeedHTML.html* in the “Answers” folder for the combined answers to Exercise 1 to 4.

Chapter 2: CSS

Congratulations! You have now completed the content of your webpage. If you load your page now, you'll probably find that it looks hideous. No worries, we are going to apply some CSS magic to convert the website into something more visually appealing. All the CSS code should be added between the `<style type="text/css">... </style>` tags. After each exercise, I encourage you to refresh the page in your browser to see the effects of the CSS rules you added.

Exercise 1: Styling the `body` element

First, let's select the `body` element by writing

```
body{

}
```

Now we need to add three CSS rules to the `body` element.

1. By default, most browsers add a small margin to the `body` element. If you look at the page that you have now, you can see that there's a small gap between the text and the edge of the browser. To remove that gap, we write `margin: 0;` to set the margins of the `body` element to 0px.
2. In addition, we'll also set the font family for the entire page. This demo site uses the `Tahoma, Geneva, sans-serif` font family. Try setting this property yourself.
3. Finally, let's set the minimum width of the `body` element to 960px. This is because we'll be setting the width of the `container` class to 960px later. Hence, we do not want the width of the `body` element to be less than that of the `container` class. To set minimum width, you use the `min-width` property. There is also a corresponding `max-width` property for setting maximum width.

Now refresh the page and observe the difference.

Exercise 2: Styling the `container` class

Now let's style the `container` class. First select the `container` class. Then add the following rules:

1. Set `width` to 960px.
2. Center align the container class using the `margin` property. (Refer to Chapter 4 in the book if you have forgotten how to do it)
3. Set the `overflow` property to `auto`. This is to inform the browser to add scroll bars if the content in the container ever exceeds its width.

Note: If you want to see the boundaries of the `container` class, simply add the line `border: 1px solid black;` to its rules. This is a handy trick to help us visualize the boxes that we are working on.

Exercise 3: Styling the `header` and `nav` element

Exercise 3.1

Let's style the `header` element now. Select the `header` element and add the following rules:

1. First, we want to center align the “Jetspeed Travel” logo. Suprisingly, to center align an image, we use the property `text-align: center;`. There is no separate `image-align` property. Add this rule to the `header` element.
2. Next, we want the header to spread across the full width of the page. Set the `width` property to 100%.

Done? Refresh your page to see the effects of the CSS rules above.

Exercise 3.2

The next element we'll style is the `nav` element. Select the `nav` element and add the following rules:

1. Change the `font-weight` to bold.
2. Use the image blacktile.jpg as the `background-image`.
3. Set the `background-repeat` property to repeat horizontally.

Exercise 3.3

Next, let's style the unordered list in the `<nav>` element. Select the `ul` child element and add the following rules to it.

1. Use the `list-style-type` property to remove the black dot beside the list items.
2. Set `margin` to 0px.
3. Set `padding` to 20px for top and bottom and 0px for left and right.

You can experiment with different margin and padding values, but I find that these values work well for me.

Exercise 3.4

Now, let's style the individual list items. Select the `li` child element and add the following rules:

1. Change the `display` to inline so that the items are displayed horizontally.
2. Change `margin` to 20px.
3. Use the `text-transform` property to change all text to uppercase. You do that by writing `text-transform: uppercase;`

Notice that we set the margins for both the `ul` and `li` elements? If you want to know where the respective margins are, simply add a 1px border to the respective elements. You can use different border colors to differentiate between the borders. Remember that `margin` refers to the space outside the border.

Exercise 3.5

The next element we'll style is the `<a>` element within the `<nav>` section. Select the element and add the following rules:

1. Use the `text-decoration` property to remove the underline.
2. Change the font color to white.

Exercise 3.6

Finally, we want to add an underline when we **hover** over the hyperlinks in the `<nav>` section.

1. Use the `border-bottom` property to add a 5px solid lightgray border to the links.

Done?

Congrats! We've come a long way. Now refresh your page. The top section should look identical to the demo page now. If it doesn't, compare your code with the “jetspeed.html” code provided in the “Answers” folder.

Exercise 4: Styling the `main` element

Exercise 4.1

Now, let's move on to the `main` element. First, select the `main` element and add the following rules:

1. Use the `text-align` property to center align all text.
2. Change the color of all text to #505E67.
3. Add a margin of 40px to the bottom of the `main` box. This margin will ensure that the `footer` is not too close to the `main` element.

Exercise 4.2

Next, let's style the <`H1`> tags in the `main` section. If we do not specify any styling rules, the browser's default rules will be applied. To style the <`H1`> tags our way, select the `H1` child element and add the following rules:

1. Change `font-weight` to bold.
2. Change `letter-spacing` to 0.5px
3. Change `font-size` to 2.2em
4. Add a top margin of 50px
5. Add a bottom border of 1px, dashed and lightgray in color.

Exercise 4.3

Now, let's style the `<img>` tags in the `main` section. Select the `<img>` child element and add the following rules:

1. Change the `width` to 70%. This sets the width of all images inside the `main` section to 70% of the width of the `container` class.
2. Set `padding` to 2px
3. Add a solid, lightgray border of 1px.

Exercise 4.4

Next, we are going to style the feedback boxes for “What Customers Are Saying”. Select the `customerBox` ***class***. The rules for the `customerBox` class will apply to both the left and right aligned boxes. Add the following rules:

1. Set `border` to 2px in width, lightgray in color and outset in style.
2. Set `width` to 60%
3. Set `margin-bottom` to 20px.
4. Set a `border-radius` of 10px for all four corners.

Exercise 4.5

Next, we have to style the `<H1>` tags inside the customerBox class.

Note that since all <H1> tags inside the `customerBox` class are also inside the `main` element, they'll inherit the properties of `main h1 {...}` unless we specifically overwrite them. Let's overwrite some rules. Select the <`h1`> child element and add the following rules:

1. Set `font-size` to 1.2em
2. Set `margin-top` to 0px
3. Set `padding` to 10px

Exercise 4.6

Next, let's style the `<p>` elements inside the `customerBox` class. Select the `<p>` child element and add the following rule:

1. Set a `padding` of 10px to the left and right edge.

Exercise 4.7

Now, we'll style the `<h2>` tags inside the `customerBox` class. This tag is used to display the customer's name at the bottom right edge of the box. Select the <`h2`> child element and add the following rules:

1. Set `font-size` to 1em

2. Set `font-weight` to `normal`
3. Set `font-style` to `italic`
4. Set `text-align` to `right`
5. Set `padding-top` to 5px
6. Set `padding-right` to 10px
7. Add a top border of 1px, dashed and lightgray in color.

Exercise 4.8

Next, we need to float the `customerBox` left and right alternatively.

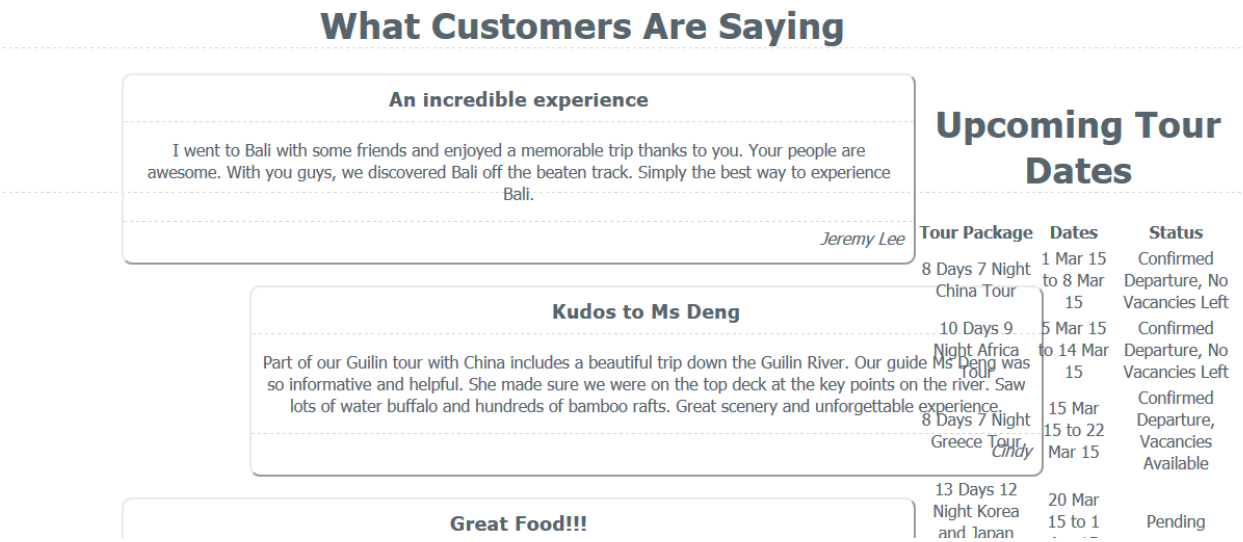
Select the `floatLeft` class and add the following rules:

1. Set `margin-left` to 15%
2. Use the `float` property to float the box left.

Now select the `floatRight` class and add the following rules:

1. Set `margin-right` to 15%
2. Use the `float` property to float the box right.

Now refresh the page in your browser window. See something similar to the page below?



The reason for this ugly misalignment is because of the `float` property. Recall we mentioned in Chapter 5 of the book that we need to clear our floats? To do that, add the following CSS

declaration:

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

Then add the HTML code `<div class="clearfloat"></div>` just before the line `<h1>Upcoming Tour Dates</h1>`. Now refresh the page again. The page looks better now right? If you do not get the desired layout, check the names of the classes. Recall that ids and classes are case sensitive in CSS.

Exercise 4.9

Now let us style the `table` inside the `main` element. Select the `<table>` child element and add the following rules:

1. Set `border` to 1px, solid and lightgray in color
2. Center align the table using the `margin` property
3. Add a `padding` of 5px
4. Set `border-radius` to 10px

Note: In this example, as the entire website only has one table, it doesn't matter whether you select `main table {...}` or just `table {...}`. In my code, I choose to be more precise and use `main table` as my selector, instead of just `table`.

Exercise 4.10

Next, we shall style the table cells using the `<th>` and `<td>` elements. Since we are going to apply the same style to both elements, we can select them together. Select both elements and add the following rules:

1. Add a `padding` of 20px;
2. Set `border` to 1px, solid and lighgray in color

Exercise 4.11

Now we add additional style for the `<th>` cells. Select the `<th>` element and add the following

rules:

1. Change background color to #505E67
2. Change text color to white
3. Set font size to 1.2em

Exercise 4.12

Finally we want to set the background color to be different for odd rows.

1. Use the `nth-child(odd)` selector to change the `background-color` to #E6E6E6. You can refer to Chapter 10 in the book for an example of how to do this.

Congratulations! We have completed the styling of the `main` section. Next, let us move on to the `footer` section, which is relatively easy to style.

Exercise 5: Styling the `footer` element

Exercise 5.1

First, select the `footer` element and add the following rules:

1. Set `text-align` to center
2. Set `width` to 100%
3. Change background color to #232244
4. Change text color to white

Exercise 5.2

Next, we'll style the `<a>` element within the `<footer>` section. Select the element and add the following rules:

1. Change text color to white

Exercise 5.3

Finally we'll style the unordered list in the `footer`. This list shares the styling rules as the list in the `nav` element. Hence, all we have to do is to add the selector `footer ul` to the selector in

Exercise 3.3.

Similarly, for list items in the unordered list, the `nav` element and the `footer` element share the same styling rules. Hence you just have to add the selector `footer li` to the selector in Exercise 3.4.

Done?

Good. Your website should look almost identical to the demo site now, except for one small difference. Notice something missing on your site? That's right, the “Email Us” link at the bottom right corner is missing. Let's add that now.

Exercise 6: Adding the “Email Us” box

Exercise 6.1

First, we need to add the box to the HTML code. To do that, let's add a `div` with `id=”contactUs”` to our HTML code, just before the `<header>` tag. Inside this `div` element, add the line `<a href="mailto:jamie@learncodingfast.com">Email Us</a>`. This creates a hyperlink that launches the email program when clicked.

Refresh the website now. Notice the “Email Us” link at the top of the page? Try clicking on it. If you have an email program installed, the program should launch.

Exercise 6.2

Now we need to style the #`contactUs` element so that it has a blue background and is positioned correctly. To do that, select the element and add the following rules:

1. Set width to 150px
2. Set height to 30px
3. Set background color to #26244B
4. Center align the text
5. Add a top padding of 10px
6. Set border-radius to 5px

To position the `div` and keep it fixed at the bottom right corner, we need to use fixed positioning and position it 0px from the bottom and 0px from the right. To do that, add the following rules:

1. Set position to fixed
2. Set bottom to 0
3. Set right to 0

Exercise 6.3

Finally, we need to change the hyperlink color and remove the underline. Select the `<a>` child element and add the following rules:

1. Change text color to white
2. Remove the underline

That's it! You have successfully coded your first website using HTML and CSS. Congratulations! Now what's next?

Chapter 3: What's next

Modifying Wordpress Themes

If you have a blog and use Wordpress, you can practice your CSS skills by modifying the Wordpress theme on your site. However, be aware that reading codes written by other programmers can be confusing, especially if the programmer does not comment his/her code. Nonetheless, you should find that modifying things like the background color, font size etc is a relatively simple task.

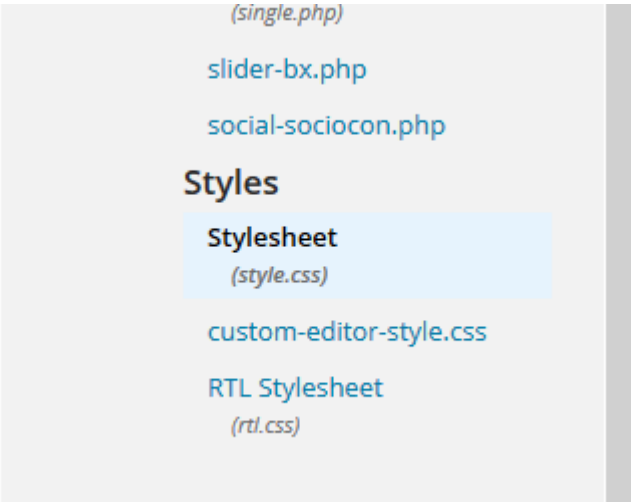
To modify your Wordpress theme, you can do it in two ways.

Method 1: Add your own CSS code

You can add your own CSS code to overwrite the original code. To do that, log in to your admin page. On the left menu, choose Appearance > Edit CSS. You'll see the screen below. Just add your code here.

Method 2: Modify the theme's CSS code

Alternatively, you can choose to modify the theme's CSS code, instead of adding your own code. To do that, select “Editor” instead of “Edit CSS”. Once you are inside the editor, scroll to the bottom of the page and select “style.css” on the right menu.



Most browsers come with developer tools to help us develop websites. We can use these tools to figure out the HTML element that we are trying to modify. Check out the youtube video below for a tutorial on how to use the developer tool in Chrome. There's a similar tool in Firefox which you can call up by right clicking and selecting “Inspect Element”.

https://www.youtube.com/watch?v=dlQrFFwAC_A

Showcasing your work

Do you want to showcase your work to your friends without having to send them all the files that are needed for the site? Or perhaps you are aspiring to be a web programmer and want to build your own web portfolio?

In any case, if you want to have your own website on the internet, you need to get your own domain and get a web hosting package. With a domain and web hosting, you can upload your work onto the internet and share your work to the public simply by giving them the URL of your website.

A domain is the address of your website. For instance, www.google.com is a domain. You can also purchase a domain for your own name, as long as it is not already taken by someone else. A domain costs about $10 USD per year and can be purchased from http://www.godaddy.com. This

is the company that I use to purchase my domain www.learncodingfast.com.

After getting your domain, you need a place to upload and store your html, CSS and image files. This is what a web hosting package is for. I host my website on http://www.hostgator.com and have been very pleased with their services. If you decide to use Hostgator, I recommend getting the "Baby Plan" which gives you unlimited domains. This means you can purchase mulitple domains from Godaddy and host all your sites on Hostgator.

More Courses

Interested in learning more? Here are some resources I recommend:

1) W3Schools CSS3 Course
http://www.w3schools.com/css/css3_intro.asp

If you are interested in learning more advanced CSS properties, W3Schools is an excellent free resource to refer to. This site covers the more fanciful properties newly added in CSS3, such as properties for animation, transition, gradient, shadows etc. Now that you understand how CSS works, you should have no problem self learning these advanced properties.

2) The Complete Web Developer Course
https://www.udemy.com/complete-web-developer-course/

Want to learn other web programming languages such as PHP and Javascript? This course is THE BEST course on udemy for learning web programming. It covers everything from HTML to CSS, Javascript, Jquery, Bootstrap, PHP, SQL and more. I strongly recommend checking it out. It even provides you with one year of free web hosting. Definitely a value-for-money course.

3) Build Your First Website In 1 Week
https://www.udemy.com/build-your-first-website-in-1-week/

This course covers how you can use Bootstrap to quickly create your own **responsive** website in one week. Bootstrap (http://getbootstrap.com/) is a hugely popular and free framework that you can use to create your own website by modifying their CSS templates. A responsive website is one that adapts to the device used to view the site. The layout of the site changes so as to reduce the amount of resizing, panning, and scrolling needed, in order to provide an easy reading and navigation experience.

If you want to learn how to use Bootstrap to create your own responsive website rapidly, I would strongly recommend checking out this course.

Note that some of the links above are affiliate links, which means I get a small commission if you purchase the service/course by clicking on my links. The price you pay is not affected by these commissions. If you found this tutorial useful, I would appreciate it if you purchase through my affiliate links, but only if you really need the product.

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.