Omnifood Project FAQs — READ BEFORE YOU START!

Frequently Asked Questions — Omnifood Project

There are some questions that students keep asking me in the course Q&A. So I decided to address some of them in this document.

***Please read this list of questions carefully before you move on with the course. If you have one of these questions while you're building the project, you can come back here and read the answer.***

1. My code is broken and I can’t move forward in the course. What should I do?
2. Brackets: Live preview not working. What to do?
3. Brackets: Quick edit not working. What to do?
4. Brackets: What are those weird JSLint warnings?
5. How does the vertical centering in the header work?
6. My icons are not working. What can I do?
7. The background image looks different in the course, it doesn’t move. Why?
8. My background images looks different on iOS devices than in the course. Why?
9. Autoprefixer is not working. What should I do?
10. My website is not loading correctly on a server. What’s wrong?
11. The responsive navigation disappears after resizing the window. Why is that?
12. There are some other problems going on with the navigation. How can I fix these?
13. My jQuery code is not working at all. What can I do?
14. The map is not working. What can I do?
15. Where is the code for the bonus lectures?
16. When do you use px and when do you use %? I’m confused by this.
17. What’s the difference between relative and absolute positioning?
18. What’s the main difference between classes and IDs?
19. Why did you use the figure and ul elements for the meals section instead of just using plain img elements?
20. What’s the difference between the body selector and the universal selector (\*)?
21. Why does the order in which we place the scripts matter?
22. How and why does the clearfix hack work?
23. Can I build an app/shopping cart/store with this course?
24. What’s the difference between Bootstrap and the grid system used in the course?
25. What’s your opinion about Wordpress, Wix, etc?

And here go all the answers

**1. My code is broken and I can’t move forward in the course. What should I do?**

Don’t panic, this is completely normal :)

Start by spending some time trying to solve the problem on your own. This is very important for your learning process! To find errors, you can compare your code with the downloadable code for each section.

If you *really* can’t solve the problem on your own, please post your code to a new codepen (<http://codepen.io/pen/>), hit save, copy your link and start a new post in the Q&A.

Please explain your problem in a very specific way: that will guarantee you get a good answer! You can even post a screenshot of the problem (not the code!).

**2. Brackets: Live preview not working. What to do?**

It's very difficult for me to find out what's wrong with Brackets on your computer. Sometimes it's enough to just restart Brackets or the Chrome window.

If that doesn't help, please try the official Brackets troubleshoot guide on github:

<https://github.com/adobe/brackets/wiki/Troubleshooting#livedev>

You can also just ignore Live Preview and open the index.html file directly.

**3. Brackets: Quick edit not working. What to do?**

I have actually stopped using the quick edit functionality long ago, because it never really works as expected, so don't worry about it. You don't need to use quick edit at all. Just add the code to your CSS stylesheet wherever you see fit.

**4. Brackets: What are those weird JSLint warnings?**

These errors that Brackets shows you are because it expects pure JavaScript code, and not jQuery code, as we are using here. In other words, Brackets thinks we have errors in our JavaScript code, because it doesn't seem to understand that we're using the jQuery framework.

**5. How does the vertical centering in the header work?**

Please see the following answer about this question here:

<https://www.udemy.com/design-and-develop-a-killer-website-with-html5-and-css3/learn/v4/questions/851800>

**6. My icons are not working. What can I do?**

Sometimes using the icons on the local computer doesn't work very reliably. Please use the following icons instead:

<link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/2.0.1/css/ionicons.css">

**7. The background image looks different in the course, it doesn’t move. Why?**

I show you how to do that effect later in the course, it appeared in one video by mistake because I produced one after all the other videos.

But here's how you get that effect with the background image:

background-attachment: fixed;

**8. My background images looks different on iOS devices than in the course. Why?**

The reason why it doesn't work is that the property:

background-attachment: fixed;

doesn't work on iOS devices. What we can do in this situation is to target iOS devices using media queries, and apply the following rule:

background-attachment: scroll;

Please take a look at this resource: <https://css-tricks.com/snippets/css/media-queries-for-standard-devices/>

Alternatively, you could just ignore background-attachment: fixed;  altogether for smaller screen widths. That should be easier.

**9. Autoprefixer is not working. What should I do?**

You can try to highlight individual sections and prefix only those sections one by one, sometimes that works.

Also, it only works if you have no bugs in your CSS code at all. You can check if you have some here:

<https://jigsaw.w3.org/css-validator/>

**10. My website is not loading correctly on a server. What’s wrong?**

This problem is most likely related with lowercase and uppercase filenames. On Windows or macOS it doesn't matter if you use lowercase and uppercase in filenames. However, on the server, which is typically Linux, it does matter.

For example, if your file is called **s**tyle.css and you include it in your HTML as **S**tyle.css, it won’t work on the server. So please make sure that you include all your files with the exact correct filename.

**11. The responsive navigation disappears after resizing the window. Why is that?**

This is actually a small bug in my code. There is a post in the Q&A that shows you how to solve that issue:

<https://www.udemy.com/design-and-develop-a-killer-website-with-html5-and-css3/learn/v4/questions/821648>

**12. There are some other problems going on with the navigation. How can I fix these?**

I already did that for you! I coded up a new version of the responsive navigation, it fixes all the issues there with the existing one.

Please replace all the navigation-related CSS code in your 767px media query by the one in this fiddle:

<https://jsfiddle.net/d83ykzku/1/>

Please also replace the jQuery code for the navigation with the one you can find in the link above. You can of course change the colors and style :)

**13. My jQuery code is not working at all. What can I do?**

Has the jQuery worked before? Are you sure you included the script.js file correctly, as well as the jQuery library? Are you using the correct class names in both HTML and jQuery code?

If your answer is YES, then post a question like I explained before.

**14. The map is not working. What can I do?**

Please take a look at this post in the Q&A where I solve one of the possible problems with this:

<https://www.udemy.com/design-and-develop-a-killer-website-with-html5-and-css3/learn/v4/questions/1589266>

<https://www.udemy.com/design-and-develop-a-killer-website-with-html5-and-css3/learn/v4/questions/1967866>

**15. Where is the code for the bonus lectures?**

Map lecture:<https://jsfiddle.net/uo3uxo1f/>

PHP contact form lecture:<https://jsfiddle.net/mf0ydjfq/>

**16. When do you use px and when do you use %? I’m confused by this.**

Okay, that is a very good question. So, we use px and % in different situations. We use percentages more for "layout" elements, which means elements that define the layout and that we want to change according to the screen width. Also, percentages are most useful for defining horizontal distances between elements, such as widths  or margin-left  or margin-right , not vertical distances, because responsive web design works with screen widths (horizontal). In other words, everything that we want to change based on the screen width when the screen becomes smaller, should be defined in %.

Also, font-sizes are always defined in %, except for the base font-size  defined for the body element, so that we can easily change font-sizes for smaller screen sizes. More about that later in Section 6.

We use px to define some margins and paddings, and sometimes to define smaller distances and distances that doesn't need to change when the screen changes. For the paddings, please note that we can only use px here thanks to box-sizing: border-box;  , which ensures that the box width (defined in %) always stays the same no matter how much padding we add inside of it.

**17. What’s the difference between relative and absolute positioning?**

Relative positioning simply positions the elements relative to the other elements on the webpage. Elements just appear on the website in the order they are defined in the HTML document. You can use margins to relatively position elements, i.e. to create space between elements.

In absolute positioning, we can use the top , bottom , left  and right  properties to perfectly position an element inside its relatively positioned parent, wherever we want. This gives us a lot of control, which is extremely useful, and you will see examples of this throughout the course. We need to explicitly say that the parent has position: relative, otherwise the browser would not know where to position the absolute element.

**18. What’s the main difference between classes and IDs?**

The main difference is that you can use classes as many times as you want on each website, but IDs only once.

This can be a problem. Imagine that you use an ID for something you think you will only use once, but later you decide you want to reuse that style. Then you'd have to go back and change from ID to class, and you’d also have to update your CSS. That's why you should only use classes and no IDs. I agree that this may not be the case with this exact project, but it will help you in the future.

The important issue here is code maintenance. It's easier to maintain code which has just classes and not a mixture of classes and IDs. That's why I have chosen to never use IDs and recommend students to do the same.

**19. Why did you use the figure and ul elements for the meals section instead of just using plain img elements? (This logic applies to all questions of this type)**

Please understand that in this course I had to mix a real website with teaching things to students. That's why in some circumstances, like having the figures just with an image and no caption, I do things just to show you new elements or stuff like that. That's why I used the figure element. This gives you a foundation so that you can now go ahead and add a <figcaption>   to it.

As of the unordered list, this is just to make it more semantic. These images are, in fact, like a list of meals with an image, right? And if we feel like it's a list, then we should express that with HTML, even if it doesn't look any different in the final website. That's called semantic HTML, and it’s important to use it.

**20. What’s the difference between the body selector and the universal selector (\*)?**

The universal selector simply selects each and every element on the webpage and applies the styles to all of them, unless we apply a different style somewhere else.

The body and html are different. We apply styles to the body/html elements that will be inherited by other elements. These are most commonly styles related to text, like the font-family, font-size or font-weight properties, because these properties get inherited by other elements. This means that once we define them on a parent element, all of the child elements will also get these properties automatically. And that's why we define these right on the html and body elements, because they are the parent elements of all the other elements.

**21. Why does the order in which we place the scripts matter?**

Our own script.js file should always be the last one. That's because of the dependencies we have in our code. For example, with jQuery, we use it in our script.js file., so it's a dependency. And we need to load that dependency before we can use it, of course. And that's why the jQuery should be loaded first. And the same of course applied to other external scripts.

**22. How and why does the clearfix hack work?**

The .clearfix class has the pseudo-element :after  because we want to properties, especially clear: both  (which does the real work of clearing the floats) to happen at the end of the element (after) on which we use the clearfix class. As for the other properties, there is no real explanation, this is just a "hack", which means that it solves the problem in a strange way. This (or in a slightly different version) is how everybody uses the clearfix.

**23. Can I build an app/shopping cart/store with this course?**

So this course only covers HTML and CSS, which are front-end technologies, or client-based technologies, meaning that these run in the user's browser. You use them to build the user interface that the user sees and interacts with.

Now, a shopping cart is somewhat different, because it involves some functionality, some logic, that needs to be programmed. We can't do that with HTML and CSS. This is called back-end development, and it usually involves a server and database, and you need some more advanced coding skills for it.

You can learn some more on this blog post I wrote some time ago:

<http://codingheroes.io/blog/>

**24. What’s the difference between Bootstrap and the grid system used in the course?**

The difference between Bootstrap and this small Grid System is that Bootstrap is an entire framework including thousands of lines of HTML, CSS and JavaScript, which you can use for quickly building a website using pre-defined building blocks. It also includes a sophisticated grid system. Now, the Grid System we use in this course is just a small CSS file which defines the rows and columns of our small fluid grid. If you look at the code, you'll see it's so simple that you could even hand-code it yourself :)

I thinks it's important to learn Bootstrap as well, because it's most likely the most popular framework in the world, but it's equally important to first fully understand the fundamentals of HTML, CSS and responsive web design without any framework.

**25. What’s your opinion about Wordpress, Wix, etc?**

In my opinion, before you start creating websites using one of these tools, you should know how the underlying technologies of web development work, and that's HTML, CSS and also JavaScript (not so much covered in this course, but in a future course).

First, knowing the fundamental building blocks of the web allows you to create 100% custom websites without any themes or constraints. You are not bound to a template or theme that someone else designed. Creativity is the only limit when you really know what you're doing, and that is very empowering.

Second, I think it's far more professional to actually know how stuff works under the surface, then just make a website for a client using some Wordpress theme or wix template.

And third, HTML, CSS and JavaScript are used to build web applications, so if you want to do that, you obviously will have to know them really well.