ADVANCED HTML/CSS

CoderDojo

SETTING UP YOUR SITE

Card 1 of 6

I'm Learning: HTML

Produced by CoderDojo Foundation // @CoderDojo

1 If you've finished the **Beginner** and **Intermediate HTML & CSS Sushi Cards**, then you already know a lot about how to write in those two languages. These cards are going to show you how to use all those pieces to put together a professional looking website, like the one you can see at dojo.soy/a-html-fin.

2 Over the next few cards you'll see how to:

- Lay out the skeleton of a website
- Setup a simple page template
- Build an animated menu for your site
- Create an awesome looking header
- Create a gallery of your coding projects

3 To start with, you'll need the skeleton of your website. Create a new **directory** called **advanced_site** to put your website in.

Setup a couple of other **directories** inside that one:

- **css** — Where you'll put your style sheets. Very useful to keep them organised if you have several of them for things like different themes, page-specific styles, etc.
- **img** — Where you'll put any images you need to put on your website, like a logo, photos, or screenshots of your projects.

CoderDojo

SETTING UP YOUR SITE

Card 1 of 6

I'm Learning: HTML

Produced by CoderDojo Foundation // @CoderDojo

4 Then create a few files to work from:

In the **advanced_site** directory (the **root directory** of your website) create empty files called:

- **template.html** — The template page that you'll be copying into all the new pages you create.
- **index.html** — The homepage of your website.
- **projects.html** — The page you'll list all your coding projects on.
- **about.html** — A page where you'll include a little information about yourself.

In the **advanced_site/css** directory create an empty file called:

- **style.css** — The CSS file you're going to put all your styles into.

Copying templates

Copy-pasting template code is not the way professional web developers would do it, but you'll need to learn **JavaScript** or some other programming language like **Python** or **Ruby** before you can learn the even cooler way to do web page templates.

5 You've got all the files in place now. In the rest of the cards you'll be filling them with the code that makes up your website! Putting together a skeleton like this is a good way to start your big website projects. It helps you remember all the pieces you meant to include later, when you're deep into the code.

ADVANCED HTML/CSS

CoderDojo

SITE SKELETON

Card 2 of 6

I'm Learning: HTML

Produced by CoderDojo Foundation // @CoderDojo

1 On this card, you're going to setup your template page. First, open **template.html** and create the outline of a standard **html** file:

```
<!DOCTYPE html>
<html>
    <head>
    </head>

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

2 Now, include your **stylesheet** in the **head** section, and add a place to put the titles of you page. Mark it with a **comment**.

```
 <head>
    <title><!-- Page title goes here --></title>
    <link rel="stylesheet" type="text/css"
href="css/style.css"/>
 </head>
```

3 Next, create the three **div** tags that almost every webpage has. Give them classes to match their roles (e.g. **class="header"**):

- First a **header** div: Contains the menu and website title
- Then a **content** div: Where the main content of the page (like your gallery of projects) goes.
- Finally, a **footer** div: Contains less important or obvious links to things like a "contact me" page.

CoderDojo

SITE SKELETON

Card 2 of 6

I'm Learning: HTML

Produced by CoderDojo Foundation // @CoderDojo

4

You should come up with the basic styles to use across your website: Fonts, text sizes, spacing, colours, etc. and put them in your **style.css** file.

You already know enough CSS from the Beginner and Intermediate cards to write your own styles, but here are some that you can play around with. Notice that I try to make **headings** and **body text** noticeably different.

```
* {
    box-sizing: border-box;
  }

body {
  font-family: "Times New Roman", serif;
  margin: 10px 0 0 0;
  padding: 0 10%;
}

h1, h2, h3, h4, h5, h6 {
  background-color: #49B749;
  color: #ffffff;
  font-family: Helvetica, Arial, sans-serif;
  padding: 0 0 10px 10px;
  width:100%;
}
```

CSS Properties

In this code I've used a number of CSS **properties** you've seen before. However, there are dozens of them. You'll learn them as you need them. You can look them up on a website like dojo.soy/cssprops

CoderDojo

BASIC MENU

Card 3 of 6

I'm Learning: HTML

Produced by CoderDojo Foundation // @CoderDojo

1

The next few cards will show you how to make an awesome menu for your website by applying some clever styling to **hyperlinks**. This is a cooler version of the menu you made with the Beginner Sushi Cards. Start by opening **template.html** and adding a list of links to the **header** div tag, like this:

```
<div class="header">
 <nav>
     <ul>
         <li><a href="index.html">Home</a></li>
         <li><a href="about.html">About Me</a></li>
         <li><a href="projects.html">Projects</a>
</li>
     </ul>
 </nav>
</div>
```

Nav Tags

The **nav** tag tells the browser (and tools used by people who have visual impairments) that this piece of the page is the **navigation**, used to get around the website.

2

Load **template.html** in your browser. It doesn't look like much, does it? You need to make that list look cool! You can do this, of course, by adding some **classes** and some **CSS**. Add a **menu** class to the **ul** tag and a **menu-button** class to each of the **a** tags.

CoderDojo

BASIC MENU

Card 3 of 6

I'm Learning: HTML

Produced by CoderDojo Foundation // @CoderDojo

3 Now, you're going to update the **CSS** in **style.css** bit by bit, so you can watch your changes.

First, make the list arrange itself like a menu:

```
.menu {
  margin: 5px auto 10px auto;
  list-style:none;
  text-align: center;
  display: flex;
  justify-content: space-around;
}

.menu > li {
  display: inline-block;
}
```

Flex

The **display: flex** in **.menu** is pretty powerful. The details of using **flex** to lay out pages would take a whole Sushi Series on its own, but you can learn more at dojo.soy/a-html-flex.

4 You've got a menu that connects your pages and it's in the right place, you can make it cooler! You've already got the classes in place, so now you just need to add some more **CSS**. You'll find it on the next card!

CoderDojo

AWESOME MENU

Card 4 of 6

I'm Learning: HTML

Produced by CoderDojo Foundation // @CoderDojo

1 Open **style.css** and add the following (or change it around, you know what you're doing by now!):

```
.menu-button {
  background-color: #0093D5;
  border: 1px solid #CCCCCC;
  border-radius: 5px;
  color: #ffffff;
  display: inline-block;
  font-family: "Helvetica Neue", Helvetica, Arial,
sans-serif;
  margin: 5px;
  padding: 5px 10px;
  text-align: center;
  text-decoration:none;
 }
```

Font Choice

Notice that the same fonts are used here as for the headings on the rest of the site. This helps the navigation stand out.

2 You're going to put in a transition. First, tell **.menu-button** how to handle it, so add this line to the **.menu-button** class in your **style.css**:

```
transition: all 0.2s ease-out;
```

This tells it to take 0.2 seconds and to slow down near the end.

CoderDojo

AWESOME MENU

Card 4 of 6

I'm Learning: HTML

Produced by CoderDojo Foundation // @CoderDojo

3 Now you just need to use the **:hover** selector to change what the button looks like when a user puts their mouse cursor over it, like this:

```
.menu-button:hover {
    box-shadow: 0 2px 2px rgba(0,0,0,0.2);
    transform: translateY(-2px);
}
```

This gives the button a shadow and moves it on the **Y-axis** (up/down) by 2 pixels.

Play with it!

Try changing the **hover** behaviour. Play with size, colour and time!

4 Now you've got a menu! Add a title (**h1**) and logo (**img**), if you like, to your **header**, above the menu. Then add "Made by [your name here]" to the **footer**. The template's done! Time to build some pages with it!

5 Save **template.html** and then copy its contents. Paste that into **index.html**, **about.html** and **projects.html** and save all of them.

6 Go to **index.html** and add a few paragraphs welcoming people to your site. Maybe tell them where they can learn to build one of their own! Now you've got a website!

CoderDojo

FANCY HEADINGS

Card 5 of 6

I'm Learning: HTML

Produced by CoderDojo Foundation // @CoderDojo

1 Next, you're going to update your **about.html** page to include a picture of your online avatar (if you don't have one, just grab something cool off the web — personally, I like to use kitten pictures).

Make sure the picture is square so you turn it into a circle using just HTML and CSS! Save it into **img** as **profile_pic** with the correct **extension** (.jpg, .png, etc. depending on its format).

2 On the **about.html** page, add these bits of HTML to your **content** div:

```
<div class="picture-heading">
    <h2>Who am I?</h2>
    <img src="img/profile_pic.[extension]" />
</div>
```

3 Add the following to **style.css** to setup the basics of your cool new heading.

```
.picture-heading {
    height: 150px;
    margin: 10px 0;
    position: relative;
}
```

Here you're setting **position: relative** so you can set the **img** and **h2** tags to have **position: absolute**, which lets you place them one on top of the other, with the right CSS!

CoderDojo

FANCY HEADINGS

Card 5 of 6

I'm Learning: HTML

Produced by CoderDojo Foundation // @CoderDojo

4 Next, set the properties on the image inside the **picture-heading**. You use **.picture-heading img** (**img** tags inside something with the **picture-heading** class) as the identifier for this:

```
.picture-heading img {
height: 150px;
border-radius: 50%;
z-index: 40;
position: absolute;
left: 20px;
border: 2px solid #ddd;
}
```

There are a few interesting bits this time:

- Setting the **border-radius** to 50% makes the picture round
- The **z-index** controls what appears over it. Higher ones go on top.
- Setting the position to **absolute** and then using the **left** property, to move it in 20 pixels from the left of the **.picture-heading**.

5 Finally, you need to set the properties on the heading itself.

```
.picture-heading h2 {
  position: absolute;
  line-height: 100px;
  height: 100px;
  padding: 0 0 0 190px;
  width: 100%;
}
```

ADVANCED HTML/CSS

CoderDojo

LIGHTBOX

Card 6 of 6

I'm Learning: HTML

Produced by CoderDojo Foundation // @CoderDojo

1 Your website is really coming together! It's starting to look like something a pro coder would build! One last trick that you'll have seen on loads of websites. It's called a **lightbox**: you click on an image, or button, or anything really, and the screen dims and something else (often a bigger version of that image) appears.

2 You're going to build this on your projects page, so you can copy it for however many projects you have. Open **projects.html** and in the **content div** add the following (feel free to use other images if you want!):

```
 <a href="#box1">
   <img src="http://placehold.it/150x150text=thumb1"
class="thumbnail">
 </a>

 <a href="#_" class="lightbox" id="box1">
     <h3>Project Name</h3>
     <img src="http://placehold.it/400x300?
text=box1">
     <p>Project description</p>
 </a>
```

A **thumbnail** is the name given to the small image that is clicked on to show the larger image. You're just going to use one for now, to understand how this works, but you can use a whole bunch of them later, maybe in a **table**, to show off all your projects!

CoderDojo

LIGHTBOX

Card 6 of 6

I'm Learning: HTML

Produced by CoderDojo Foundation // @CoderDojo

3 Now for the **CSS**. There are a few clever bits in here, that I'll explain afterwards. As usual, this all goes in **style.css**:

```
.lightbox{
    background: rgba(0,0,0,0.8);
    color: #fff;
    height: 100%;
    left: 0;
    position: fixed;
    text-align: center;
    text-decoration: none;
    top: 0;
    visibility: hidden;
    width: 100%;
    z-index: 999;
}

.lightbox:target {
    outline: none;
    visibility: visible;
}
```

- The **lightbox** is hidden most of the time, by **visibility: hidden**
- It has **position: fixed**, which means it will stay in place even if you scroll the page. With its width and height, it takes over the whole page.
- Turn off the **text-decoration** to avoid underlining everything inside the **a** tag.
- The **lightbox:target** class only applies when the **lightbox** was the **target** of the last **hyperlink** clicked. So clicking anywhere will switch the **visibility** back to **hidden** and hide the **lightbox**!