

Front End Overview

Create a simple website with html and CSS

1. Plan your layout The first step of any website is always to know what you want on it and (vaguely) how you want it to look. So, the first step is to do a rough sketch - either on paper or on the computer, depending on which you find easier.

Remember, it doesn't have to look good. Here's mine:

(wire frame picture)

As you can see, it's very rough. The lines aren't straight and nothing is even but I can still see how the site is going to look and what sections I need to have.

In this layout, I have a header (navigation bar), three sections and a footer.

2. Get the 'boilerplate code' set up Now, it's time to get the basic code that you have at the start of any website (this is commonly called the boilerplate).

Do this by:

1. Creating a new folder on your computer for the website
2. Create new empty index.html and style.css files inside
3. Add the basic 'boilerplate code' to your index.html file:

Finally, open up your index.html in a web browser to check everything's working:

3. Create the elements in your layout Now it's time to create the layout/section elements that you planned in step 1!

The best way to do this is by using semantic elements:

,
,
, and
.

Here is the HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title> Boberick the llama</title>
    <link rel="stylesheet" href="style.css">
```

```

</head>
<body>
  <header>

  </header>
  <main>
    <section id="hero">

    </section>
    <section id="about">

    </section>
    <section id="contact">

    </section>
  </main>
  <footer>

  </footer>
</body>
</html>

```

Note that we are giving the <section>s ids, so we can refer to them later.

If you reload the page, you will see there is nothing there. This is because we are just creating the sections of the page, not the actual stuff in them.

4. Filling in the HTML content

Once you have the sections of the page, it's time to fill them up! If you know what content you are going to be using, put that in. If not, put in some dummy text and replace it with the actual content later.

Here is the HTML after filling in some content:

```

<!DOCTYPE html>
<html>
  <head>
    <title>Boberick the llama</title>
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <header>
      
      <nav>
        <ul>

```

```

        <li><a href="#hero">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#contact">Contact</a></li>
    </ul>
</nav>
</header>
<main>
    <section id="hero">
        <div class="section-inner">
            
            <h1>Hi, I'm Boberick the llama.</h1>
        </div>
    </section>
    <section id="about">
        <div class="section-inner">
            <h2>About me</h2>
            <p>I'm a really awesome llama. Every day I wake up, munch on some grass, do some coding and then go back to sleep.</p>
            <h3>Achievements</h3>
            <ul>
                <li>Bachelor of photogenic posing, 2010</li>
                <li>Llamaness certification from the Llama Institute, 2014</li>
                <li>I coded a website, 2017</li>
            </ul>
        </div>
    </section>
    <section id="contact">
        <div class="section-inner">
            <h2>Contact me</h2>
            <p>You can find me on:</p>
            <ul>
                <li><a href="https://twitter.com/llama">Twitter</a></li>
                <li><a href="https://www.reddit.com/user/llama">Reddit</a></li>
                <li><a href="https://www.instagram.com/llamasporfavor/">Instagram</a></li>
            </ul>
            <p>Or, you can <a href="mailto:llama@codetheweb.blog">send me an email</a>.</p>
        </div>
    </section>
</main>
<footer>

```

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

If you reload the page, you will see that we now have some content!

5. Adding some basic layout CSS Once we're done with our HTML, it's time to move on to CSS! The first and most important part to focus on first is to get it looking like our layout - then we can focus on the details.

This means that we need to focus on properties like width, height, margin, padding, position, and display. Also, we need to make sure the images are the right size so that they don't obliterate the page.

Here is the CSS that we will add to our style.css:

```
body {
    margin: 0;
    margin-top: 50px;
}

header {
    display: flex;
    position: fixed;
    top: 0;
    left: 0;
    right: 0;
    height: 50px;
    line-height: 50px;
    background-color: #eee;
}

header * {
    display: inline;
    height: 50px;
}

header ul {
    padding: 0;
}

header li {
    margin-left: 20px;
}

section {
    height: 100vh;
    border: 1px solid black;
```

```

    display: flex;
    justify-content: center;
    align-items: center;
    text-align: center;
}

#hero .profile-img {
    width: 300px;
}

footer {
    text-align: center;
    padding: 50px;
}

```

Here, we are only adding styles to make our overall layout look similar - not the individual content. We make sure that the sections are set to 100% viewport height, make the header have a fixed position, position the items in the header, and more. We also use flexboxes to center the content in our sections.

This is the result: (img)

6. Add more specific styles Once the basic framework of the site is done, we can add more specific styles.

Now we can make our website look good!

Here's our CSS:

```

body {
    margin: 0;
    margin-top: 50px;
    font-family: sans-serif; /* Add this line */
}

header {
    display: flex;
    position: fixed;
    top: 0;
    left: 0;
    right: 0;
    height: 50px;
    line-height: 50px;
    background-color: #eee;
}

```

```
header * {
    display: inline;
    height: 50px;
}

header ul {
    padding: 0;
}

header li {
    margin-left: 20px;
}

section {
    height: 100vh;
    border: 1px solid black;
    display: flex;
    justify-content: center;
    align-items: center;
    text-align: center;
}

#hero .profile-img {
    width: 300px;
    border-radius: 50%; /* Add this line */
}

footer {
    text-align: center;
    padding: 50px;
}

/* Add everything below here */

#hero h1 {
    font-size: 3em;
}

section h2 {
    font-size: 2.5em;
}

section h3 {
    font-size: 1.5em;
}
```

```
header a {
  text-decoration: none;
  color: black;
}
```

As you can see, we've made the title bigger, rounded our image (using border-radius), and changed the font. We've also removed some default styling from the header links.

Here's the result:

7. Add colors and backgrounds Yay, we're on the home stretch now! It's time to add the finishing touches to our website - colors and backgrounds!

This is what will make our site look really awesome.

Here is the CSS:

```
body {
  margin: 0;
  margin-top: 50px;
  font-family: sans-serif;
}

header {
  display: flex;
  position: fixed;
  top: 0;
  left: 0;
  right: 0;
  height: 50px;
  line-height: 50px;
  background-color: #eee;
}

header * {
  display: inline;
  height: 50px;
}

header ul {
  padding: 0;
}

header li {
  margin-left: 20px;
}
```

```
section {
  height: 100vh;
  border: 1px solid black;
  display: flex;
  justify-content: center;
  align-items: center;
  text-align: center;

  background-size: cover; /* Add this line */
  background-position: center center; /* Add this line */
  background-repeat: no-repeat; /* Add this line */
  background-attachment: fixed; /* Add this line */
}

#hero .profile-img {
  width: 300px;
  border-radius: 50%;
}

footer {
  text-align: center;
  padding: 50px;
}

#hero h1 {
  font-size: 3em;
}

section h2 {
  font-size: 2.5em;
}

section h3 {
  font-size: 1.5em;
}

header a {
  text-decoration: none;
  color: black;
}

/* Add everything below here */

#hero {
```



```
background-image: linear-
gradient(rgba(255,255,255,0.75),rgba(255,255,255,0.75)),
url('https://codetheweb.blog/assets/img/posts/steps-to-creating-a-
website/field.jpg');
}

#about {
background-image: linear-
gradient(rgba(255,255,255,0.75),rgba(255,255,255,0.75)),
url('https://codetheweb.blog/assets/img/posts/steps-to-creating-a-
website/beach.jpg');
}

#contact {
background-image: linear-
gradient(rgba(255,255,255,0.75),rgba(255,255,255,0.75)),
url('https://codetheweb.blog/assets/img/posts/steps-to-creating-a-
website/canyon.jpg');
}
```

As you can see, we've added some general background styles to the section elements, as well as adding a background-image to each section individually.

The reason for the linear-gradient(rgba(255,255,255,0.75),rgba(255,255,255,0.75)), before the url('image.jpg') is because otherwise the text is hard to read - so we add a semi-transparent white overlay ontop. I wrote a bit more about that here.

8. Celebrate!

References:

1. <https://codetheweb.blog/2017/12/10/steps-to-creating-a-website/>