

Lab: Create a Blog

LAB OVERVIEW

In this lab you will code a food blog called Om Nom Nom. This lab will cover topics including: HTML, CSS Basics, but also layout and advanced layout concepts including: floats, position and flexbox.

Take a moment to look over the *mockup.pdf* for the blog. You can see it is a fairly simple two column layout with articles about restaurants on the left and a sidebar on the right side.

Creating a complex webpage like this is not easy, so we'll do it in steps. We will follow the general steps in the order below.

- HTML Text
- HTML Images
- HTML Layout
- CSS Text
- CSS Backgrounds
- CSS Boxmodel
- CSS Layout

You do not have to follow the method in this lab, this lab is really just discussing layout techniques. How you decide to code your webpage is up to you as long as it follows the layout shown in mockup.pdf

Note that your HTML / CSS will most likely not match the Mockup.pdf 100%, that is okay. It does need to be pixel perfect, but should match the general layout and style of the mockup.

BEGIN CODING

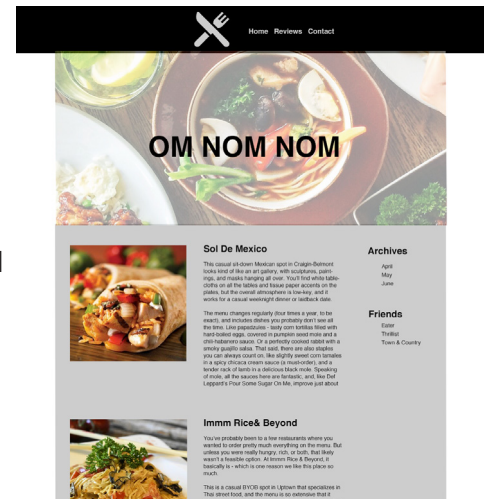
Start by creating a new HTML document. Add in all the standard HTML basic page tags including: `<!doctype html>`, `<head>`, `<title>`, and `<body>`.

Save it into the same folder that stores all of the photos you'll need for this assignment. Save as *foodblog.html*.

LAYOUT THE TEXT / HYPERLINKS

Use *website-text.txt* to assist you in laying out the text for the webpage.

The name of the food blog, Om Nom Nom, should go into a primary heading.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Om Nom Nom</title>
6 </head>
7 <body>
8
9 </body>
10 </html>
```

Each food blog title should be a secondary headings

Archives and Friends should be secondary headings and their content should be displayed as unordered lists

Save and preview. Compare your webpage with the image on the right.

LAYOUT IMAGES

Now that we've laid out text let's start layout out the images. Add in the images including the logo and the image for each restaurant review. The logo should come BEFORE the other navigational options and the restaurant food images should come BEFORE the name of the restaurant.

Save and preview: Compare your webpage with the image on the right.

CREATE HTML STRUCTURE

Now that we have text and images laid out, let's begin to organize our page a bit more with HTML structure.

Let's begin by dividing our page into three major categories `<header>`, `<main>` and `<footer>`. Remember that the `<header>` contains all the elements that typically go on the top of every webpage.

For this website the header elements including the:

- logo
- navigation bar

Next let's lay out the `<main>`

Remember that the `<main>` contains the main part of the webpage. For this design the main will include:

- Site Title: 'Om Nom Nom'
- food articles
- sidebar elements (Archives and Friends)

Finally, let's lay out the `<footer>`. Put the legal information inside of the `<footer>`. We will wait on the advertisement information at the moment

[Home](#) [Review](#) [Cities](#) [Contact](#)

Om Nom Nom

Sol De Mexico

This casual sit-down Mexican spot in Craigin-Belmont looks kind of like an art gallery, with on all the tables and tissue paper accents on the plates, but the overall atmosphere is low-key

The menu changes regularly (four times a year, to be exact), and includes dishes you probabl boiled eggs, covered in pumpkin seed mole and a chili-habanero sauce. Or a perfectly cooke always count on, like slightly sweet corn tamales in a spicy chicaca cream sauce (a must-ord the sauces here are fantastic, and, like Def Leppard's Pour Some Sugar On Me, improve just

While the sauces are reliably delicious, not everything is a hit. For example, you'll frequentl the enchiladas, or overcooked carne asada and salmon that both really need the moisture fror bones in the shredded-chicken-filled masa boats. But as long as you focus on what this place good meal.

Chicago Cut



Om Nom Nom



Sol De Mexico

This casual sit-down Mexican spot in Craigin-Belmont looks kind of like an art gallery, with sculptur on all the tables and tissue paper accents on the plates, but the overall atmosphere is low-key, and it w

The menu changes regularly (four times a year, to be exact), and includes dishes you probabl don't s

```
<header>
  
    <a href="#">Home</a>
    <a href="#">Review</a>
    <a href="#">Cities</a>
    <a href="#">Contact</a>
  </nav>
</header>
```

```
<footer>
  Copyright Om Nom Nom 2020
</footer>
```

ADDITIONAL HTML5 CONTAINER STRUCTURE

Now that we have a basic structure, let's add additional HTML5 containers to assist the layout.

Put all of the navigational links into a `<nav>`

Put each article (including the restaurant photo) into an `<article>` tag.

Put the sidebar information (Archives and Friends) into a `<aside>`

Organize all of the `<article>` by putting them in a `<section>` with an id of `'#foodarticles.'`

Put the heading 1 into a `<section>` tag and add an id called `#splash`

Organize all of the elements in the webpage by putting them into a `<div>` with the id of `'#container'`.

Save and preview. It should look something like the image on the right.

```
<section id="foodarticles">
  <article>
    
    <h2>Sol De Mexico</h2>
    <p>
      This casual sit-down Mexican spot in Craig-Belmont looks kind of like an art gallery, with sculptures, paintings, and masks hanging all over. You'll find white tablecloths on the tables and tissue paper accents on the tables, but the overall atmosphere is low-key, and it's casual weeknight dinner or laidback date.
```

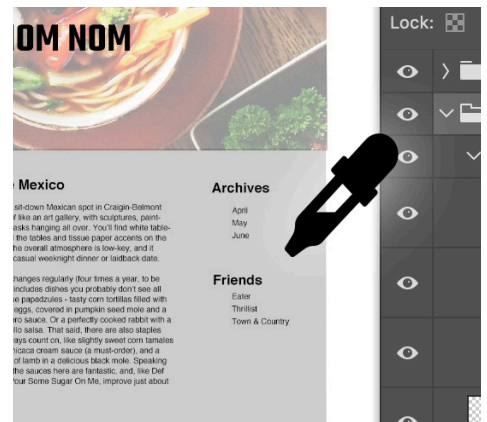
```
<div id="container">
  <header>
    <nav>
      
      <a href="#">Home</a>
      <a href="#">Review</a>
      <a href="#">Cities</a>
      <a href="#">Contact</a>
    </nav>
    <h1>Om Nom Nom</h1>
  </header>
  <main>
    <section id="foodarticles">
      <article>
        
        <h2>Sol De Mexico</h2>
        <p>
          This casual sit-down Mexican spot in Craig-Belmont looks kind of like an art gallery, with sculptures, paintings, and masks hanging all over. You'll find white tablecloths on the tables and tissue paper accents on the tables, but the overall atmosphere is low-key, and it's casual weeknight dinner or laidback date.
      </article>
    </section>
  </main>
  <aside>
    <h3>Archives</h3>
    <ul>
      <li>April</li>
      <li>May</li>
      <li>June</li>
    </ul>
    <h3>Friends</h3>
    <ul>
      <li>Eater</li>
      <li>Thrillist</li>
      <li>Town & Country</li>
    </ul>
  </aside>
</div>
```

ADD STYLE

Let's begin adding styles to this webpage. In your coding program create a stylesheet and name it `'style.css'`. Make sure to save it to the same folder as images and HTML file.

Take a moment to open *Mockup.psd* into Adobe Photoshop. Use the tools in Adobe Photoshop such as the Text Tool, Character / Paragraph panel and eyedropper tool to get a better sense of the typography and colors for the project.

You'll need this information later when we add CSS code.



STYLIZE THE TYPOGRAPHY

Since we are going to have a lot of CSS for this lab, let's organize our CSS document into sections. It will make it easier to read and edit later on.

Begin by writing the following into your CSS document:

```
*/ === Typography === */
```

Let's set up the base text. Remember that with CSS parent tags typically pass properties to their child tags. So when we can use this to our advantage. If we apply a CSS typographical setting to the `<body>` it will apply to everything inside the `<body>`, or basically everything in the webpage.

Apply the following:

```
body {  
  font-family: 'Helvetica', 'Arial', sans-serif;  
  font-size: 14pt;  
  line-height: 18pt;  
  color: #000000;  
}
```

Use *Mockup.psd* to assist you with setting the typographical properties for

- Navigation links
- Heading 1
- Heading 2
- Footer

We can also use CSS compounds to assist us with being more specific when styling a webpage. For instance, although there are navigation links and links in the sidebar, they are styled differently. We could use a class to assist with this, or use CSS compounds.

We would write:

```
nav a {  
  color: #FFFFFF;  
  text-decoration: none;  
}
```

This would only affect the hyperlinks in the `<nav>` section, instead of the entire document.

WEB FONTS

If you view *mockup.psd* you can see that the title of On Nom Nom is a special font called 'Teko'. Teko is a web font that can be found on Google Fonts.

Go to Google Fonts and embed the `<link>` to the font into your HTML file. Then you can call the font in your CSS.

Save and preview, your webpage should look roughly like the one on the right.

Sol De Mexico

This casual sit-down Mexican spot in Craig-in-Belmont looks kind of all over. You'll find white tablecloths on all the tables and tissue paper and it works for a casual weeknight dinner or laidback date.

The menu changes regularly (four times a year, to be exact), and it's - tasty corn tortillas filled with hard-boiled eggs, covered in pumpkin rabbit with a smoky guajillo salsa. That said, there are also staples: spicy chichaca cream sauce (a must-order), and a tender rack of lamb are fantastic, and, like Def Leppard's Pour Some Sugar On Me, im

While the sauces are reliably delicious, not everything is a hit. For instance, fish. Like bland diced chicken breast in the enchiladas, or overcooked from their sauces. We've also had the unnerving experience of finding as long as you focus on what this place does really well (i.e. moles



Om Nom Nom



STYLIZE THE BACKGROUNDS

Add the following code to your CSS document:

```
*/ === Backgrounds === */
```

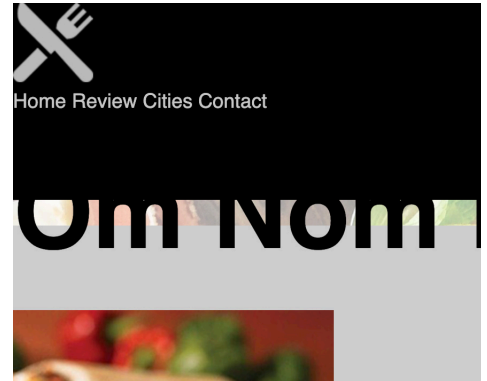
Use *Mockup.psd* to assist you with setting the background colors:

- <header>
- <main>
- <body>
- <footer>

Sample the color using the eyedropper tool and use *background-color* to set the colors in your CSS.

Let's set the background image for the title of the blog, 'On Nom Nom'. To do this use *background-image* and *background-size:cover* to apply to the *#splash*. Don't worry that it looks a bit different than the mockup as we'll add dimensions later.

Save and preview. Your webpage should look roughly like the one on the right.



SETUP BOX MODEL PROPERTIES

Add the following code to your CSS document:

```
*/ === Box Model === */
```

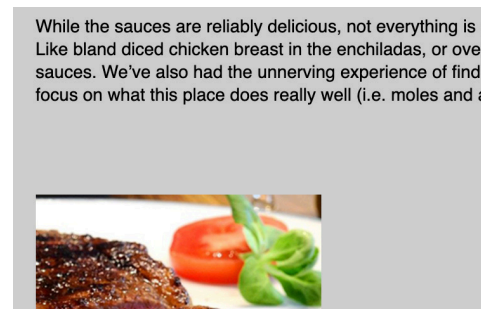
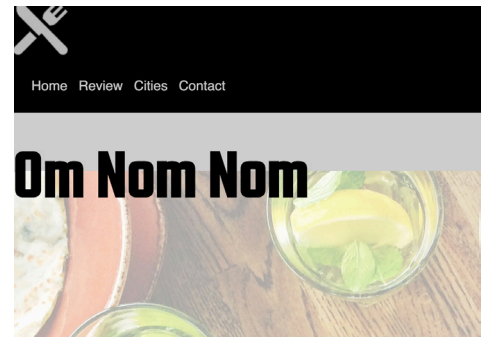
This is where we'll add box model elements such as width,height, padding and margins to our page.

Add a *height* of 400px to *#splash*

Add *padding* to the <nav>, <footer> and *#foodarticles*

Add a bit of *margin-bottom* each <article> to separate the articles from one another

Use *margin-right* side of each navigational links to separate the links more from each other.



ADD LAYOUT TO YOUR WEBPAGE

Okay, now that we have a general setup, let's begin to layout items in our page. This means creating columns and generally moving large page blocks arounds.

Add the following code to your CSS document:

```
*/ === Layout === */
```

FLOATS

A float is a way to move an item to left or right of a container. When something is floated the material below it will move up and begin to flow around it.

Create a class called *.floatleft* and set it the following:

```
.floatleft {  
    float: left;  
}
```

Then apply it to your restaurant images. Save and preview and you'll see that the images float the left side of the page and text flows around them. To add a bit of space add *.margin-right* and *margin-bottom* to the *.floatleft*.

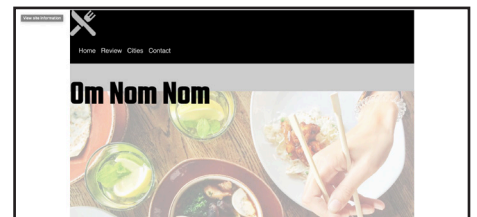
Save and preview. It should roughly like the image on the right.

CENTERING THE WEBSITE

Let's try to center the content to the middle of the page. The easiest way would be to set a width for the entire webpage, or the *#container* and then use *margin-left:auto*, *margin-right:auto* to center it.

Set the width of *#container* to a width of 1000px and use *margin:auto*.

Save and preview



CSS GRIDS

Let's lay out the columns for food articles and the sidebar. This will be a two column layout. There are different techniques to achieve this, but we will use CSS grids, as it is one of the easiest methods.

As discussed in class examples and previous labs, CSS grids allows you to turn a HTML tag into a grid of rows and columns. Once a tag has been transformed into a grid, it's direct children become the columns.

In this lab we'll begin with the `<main>` tag. If you look at *mockup.pdf* you can see a two column layout with the `#splash` area at the top as one column. You can also see that the articles column on the left is much wider than the column on the right. The `#foodarticles` column is roughly 3/4ths of the width and the sidebar is 1/4th of the width of the design.

Begin by setting `<main>` to a CSS grid

```
main {  
  display: grid;  
}
```

Next use *grid-template-columns* to create columns. The first should be 75% of the width and the second should be 25% of the width.

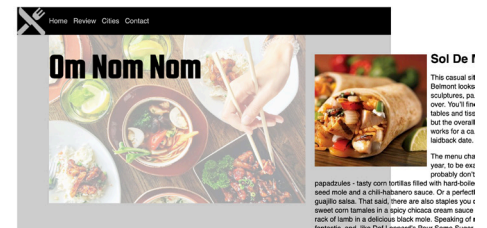
Save and your layout should look roughly like the image on the right. You'll notice that `#splash` is being displayed on the left, the `#foodarticles` is being displayed on the right and sidebar below on a new row. The reason for this is we have three items, but only two columns. To fix this we need the `#splash` to expand to take up the entire row.

To expand the `#splash` we will use *grid-column-start* and *grid-column-end*. The `#splash` needs to start at the far left or marker 1 and go to the far right or marker 3.

```
#splash {  
  grid-column-start: 1;  
  grid-column-end: 3;  
}
```

Now that the layout is correct you can apply `margin-left` to the aside to move the content over a bit. I recommend 35px.

Save and your image should look like the image on the right.



FLEXBOX

One thing that has always been difficult for developers was finding a way to center something horizontally AND vertically using CSS. A new method has come out referred to as flexbox. Flexbox allows developers to place items at the top, center or bottom, as well as the left, right, and center of a container.

In our lab we want the name of the website, Om Nom Nom, to be in the vertical and horizontal center of the page. To do this we'll use flexbox.

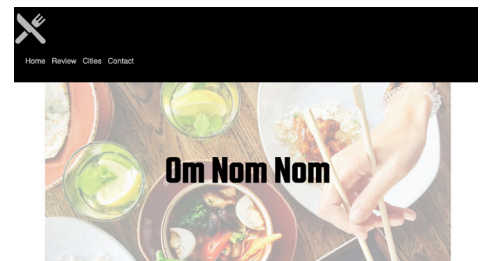
Begin by turning your #splash into a flexbox.

```
#splash {  
  display: flex;  
}
```

You can then use the properties *align-items* and *justify-content* to align the text to the vertical and horizontal center.

```
#splash {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

Save and preview.

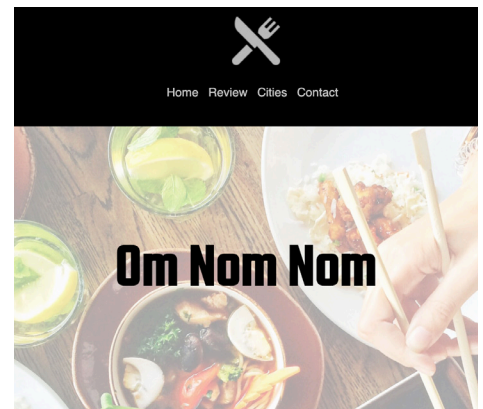


FINISHING TOUCHES

Finally, let's center the text for the `<header>` and for the `<footer>` we can use *text-align:center* for that.

Save and your webpage should look roughly like the image on the right.

Great job!



SUBMISSION

Please turn all the related files to iLearn as a zipped folder. Please upload your files to your github hub account and submit the url to the html page.