he Box Model: Davie's Burgers

In this project, you will follow step-by-step instructions to fix a fictional restaurant’s website. All of the HTML and most of the CSS is intact, but the box model properties have yet to be set. You’ll use knowledge of height, width, padding, border, and margin to complete this project.

The website’s existing **index.html** and **style.css** files are displayed in the text editor to the right. As you work, use both to see which elements you are selecting and styling. Most of the elements that you’ll need to add styles for already have rule sets in **style.css** to which you can add additional declarations.

In this project, you will follow step-by-step instructions to fix a fictional restaurant’s website. All of the HTML and most of the CSS is intact, but the box model properties have yet to be set. You’ll use knowledge of height, width, padding, border, and margin to complete this project.

The website’s existing **index.html** and **style.css** files are displayed in the text editor to the right. As you work, use both to see which elements you are selecting and styling. Most of the elements that you’ll need to add styles for already have rule sets in **style.css** to which you can add additional declarations.

If you get stuck during this project or would like to see an experienced developer work through it, click “**Get Unstuck**“ to see a **project walkthrough video**.

**Tasks**

**0/8 Complete**

Mark the tasks as complete by checking them off

**1.**

You’ll go through Davie’s Burger’s menu item roughly from top to bottom. Let’s begin with styling the <nav> element containing the logo and navigation elements.

* Set the width of the img to 180 pixels.
* Center the img horizontally using the margin property.

Stuck? Get a hint

**2.**

Set margin of span elements inside the nav to 10 pixels on the top and bottom, and 0 pixels on the left and right.

Stuck? Get a hint

**3.**

Now set some rules for the element with the class content. This element is a container for all the elements not included in <nav>.

* Set the height to 500 pixels.
* Create 10-pixel vertical margins and automatic horizontal margins.
* Set .body elements inside .content to have no vertical margin and automatic horizontal margins so that it is centered.

Stuck? Get a hint

**4.**

With a 500-pixel height for .content, some elements will overflow out of its box if the browser window is made too small.

* Make .content scrollable with the overflow property.
* Resize the browser window so it is very skinny and notice that this section is now scrollable.

Stuck? Get a hint

**5.**

Now it’s time to tackle the .header and its h1 text: ‘BBQ BACON BURGER’.

* Set the height of the .header class to 320 pixels.
* Create a 20-pixel padding for the h1 element inside the .header.
* Set vertical margins to 0 pixels and the horizontal margins to be determined automatically for the same h1 element.

Stuck? Get a hint

**6.**

Now add some box-model formatting to the ‘ORDER NOW’ .button element. As you make each change, make sure to scroll down if necessary to view the effect on the button.

* Set the width to 200 pixels.
* Set the padding to 20 pixels.
* Set the vertical margins to 40 pixels and the horizontal margins to automatic.
* Give the button a 1-pixel, solid, blue border.

Stuck? Get a hint

**7.**

Finally, style the nutrition facts section at the bottom. In the element selected with ul.nutrition:

* Set the padding of the ul.nutrition element to 20 pixels.
* Set the width of li elements within the ul.nutrition to 200 pixels.
* Set a 10-pixel vertical padding and 20-pixel horizontal padding to the same elements.
* Add a 3-pixel bottom margin to the same elements.

Stuck? Get a hint

**8.**

Good job—this menu item is looking great, and the layout and spacing is much more readable. If you’d like, you can continue refining the design or add your own spin!