# Step 1

We've provided a basic HTML boilerplate for you.

Create an h1 element within your body element and give it the text Nutrition Facts.

# Step 2

Below your h1 element, add a p element with the text 8 servings per container.

# Step 3

Add a second p element with the text Serving size 2/3 cup (55g).

# Step 4

Within your head element, add a link element with the rel attribute set to stylesheet and the href attribute set to https://fonts.googleapis.com/css?family=Open+Sans:400,700,800.

This will import the Open Sans font family, with the font weight values 400, 700, and 800.

Also add a link element to link your styles.css file.

# Step 5

Create a body selector and give it a font-family set to Open Sans with a fallback of sans-serif.

Remember that fonts with spaces in the name must be wrapped in quotes for CSS.

# Step 6

The font is a bit small. Create an html selector and set the font to have a size of 16px.

# Step 7

Wrap your h1 and p elements in a div element. Give that div a class attribute set to label.

# Step 8

Borders can be used to group and prioritize content.

Create a .label selector and give it a border set to 2px solid black.

# Step 9

Good use of white space can bring focus to the important elements of your page, and help guide your user's eyes through your text.

Give your .label selector a width property set to 270px.

# tep 10

Give your .label selector a margin property set to 20px auto, and a padding property set to 0 7px.

# Step 11

If you inspect your .label element with your browser's developer tools, you may notice that it's actually 288 pixels wide instead of 270. This is because, by default, the browser includes the border and padding when determining an element's size.

To solve this, reset the box model by creating a \* selector and giving it a box-sizing property of border-box.

# Step 12

Remember that the use of h1, h2, and similar tags determine the semantic structure of your HTML. However, you can adjust the CSS of these elements to control the visual flow and hierarchy.

Create an h1 rule and set the font-weight property to 800. This will make your h1 text bolder.

# Step 13

Give your h1 selector a text-align property of center.

# Step 14

Fine-tune the placement of your h1 by giving it a top and bottom margin of -4px and a left and right margin of 0.

# Step 15

Create a p selector and remove all margins.

# Step 16

Lines can help separate and group important content, especially when space is limited.

Create a div element below your h1 element, and give it a class attribute set to divider.

# Step 17

Create a selector for your new .divider and set the border-bottom property to 1px solid #888989. Also give it a top and bottom margin of 2px. It should not have any left or right margin.

# Step 18

The letter-spacing property can be used to adjust the space between each character of text in an element.

Give your h1 selector a letter-spacing property set to 0.15px to space them out a bit more.

# Step 19

Nutrition labels have a lot of bold text to draw attention to important information. Rather than targeting each element that needs to be bold, it is more efficient to use a class to apply the bold styling to every element.

Give your second p element a class attribute set to bold.

# Step 20

Your new class does not have any styling yet. Create a .bold selector and give it a font-weight property set to 800 to make the text bold.

Go ahead and remove the font-weight property from your h1 selector as well.

# Step 21

Give your h1 element a class attribute set to bold. This will make the text bold again.

# Step 22

Horizontal spacing between equally important elements can increase the readability of your text.

Wrap the text 2/3 cup (55g) in a span element, and give it a class attribute set to right.

# Step 23

The float property is used to place an element on the left or right of its container, allowing other content (such as text) to wrap around it.

Create a new .right selector and set the float property to right.

# Step 24

Wrap everything within the .label element in a new header element.

# Step 25

Now update your h1 selector to be header h1 to specifically target your h1 element within your new header.

# Step 26

Create a new div element below your header element, and give it a class attribute set to divider lg.

# Step 27

Create a new .lg selector and give it a height property set to 10px. Also create an .lg, .md selector and set the background-color property to black.

# Step 28

You may notice there is still a small border at the bottom of your .lg element. To reset this, give your .lg, .md selector a border property set to 0.

# Step 29

Create a new div below your .lg element and give it a class attribute set to calories-info.

# Step 30

Within your .calories-info element, create a p element. Give that p element a class attribute set to bold sm-text, and the text Amount per serving.

# Step 31

The rem unit stands for root em, and is relative to the font size of the html element.

Create an .sm-text selector and set the font-size to 0.85rem, which would calculate to be roughly 13.6px (remember that you set your html to have a font-size of 16px).

# Step 32

Below your .sm-text element, create a new h1 element with the text Calories 230. Wrap the 230 portion of the text in a span element with the class set to right.

# Step 33

Create a new .calories-info h1 selector setting the top and bottom margin to -5px, and the left and right margin to -2px.

# Step 34

Create a .calories-info span selector and set the font-size to 1.2em.

# Step 35

The larger font size of the number 230 is causing it to overflow. Give the .calories-info h1 an overflow property set to hidden to avoid this.

# Step 36

Typography is often more art than science. You may have to tweak things like alignment until it looks correct.

Give your .calories-info span selector a margin-top set to -7px. This will shift your 230 text into place.

# Step 37

Below your .calories-info element, add a div with the class attribute set to divider md.

# Step 38

Create an .md selector and give it a height property of 5px.

# Step 39

Create a new div element below your .md element. Give it a class attribute set to daily-value sm-text. Within this new div, add a p element with the text % Daily Value \*, and set the class attribute to right bold.

# Step 40

The float styling is causing the new p element to be outside of the label's border. Use your existing .divider element as an example to add a new divider after the p element.

# Step 41

Give the .divider selector a clear property set to right. This will clear the float property, pushing the divider and any following content down below the float text.

# Step 42

After your last .divider element, create a p element and give it the text Total Fat 8g 10%. Wrap Total Fat in a span element with the class set to bold. Wrap 10% in another span element with the class set to bold right.

# Step 43

Below your element with the Total Fat text, create a new p element with the text Saturated Fat 1g 5%. Wrap the 5% in a span with the class attribute set to bold right.

# Step 44

This new p element will need to be indented. Give it a class set to indent.

# Step 45

Create a new .indent selector and give it a margin-left property set to 1em.

# Step 46

Create a .daily-value p selector to target all of your p elements in the daily-value section. Give this new selector a border-bottom set to 1px solid #888989.

# Step 47

The bottom borders under your % Daily Value \* and Saturated Fat 1g 5% elements do not extend the full width of the label. Add no-divider to the class for these two elements.

# Step 48

The :not pseudo-selector can be used to select all elements that do not match the given CSS rule.

div:not(#example) {

color: red;}

The above selects all div elements without an id of example.

Modify your .daily-value p selector to exclude the .no-divider elements.

# Step 49

Now you will have to add separate dividers below your .no-divider elements.

Your first .no-divider element has a .divider after it. Create another .divider after your second .no-divider element.

# Step 50

After your last .divider, create another p element with the text Trans Fat 0g. Italicize the word Trans by wrapping it in an i element. Give the new p element the class attribute set to indent no-divider.

# Step 51

Create another .divider after your last p element.

# Step 52

After your last .divider, create a new p element with the text Cholesterol 0mg 0%. Wrap the text Cholesterol in a span element, and give that span element the class attribute set to bold. Wrap the text 0% in another span element, with the class set to bold right.

# Step 53

Below your last p element, create another p element with the text Sodium 160mg 7%. Wrap the text Sodium in a span element with a class attribute set to bold. Wrap the 7% text in another span element with the class set to bold right.

# Step 54

Add another p element with the text Total Carbohydrate 37g 13%. Like before, use span elements to make the text Total Carbohydrate bold, and the text 13% bold and right aligned.

# Step 55

Below your last p element, add another p element with the text Dietary Fiber 4g. Give the p element the class necessary to ***indent*** it and *remove the* ***dividing border***. Then create a divider below that p element.

# Step 56 Passed

Create another p element after your last .divider, and give it the text Total Sugars 12g. Assign that p element the class values necessary to ***indent*** it and ***remove the bottom border***. Then create another .divider below your new p element.

# Step 57

The advantage to creating these dividers is that you can apply specific classes to style them individually. Add dbl-indent to the class for your last .divider.

# Step 58

Create a .dbl-indent selector and give it a *left margin* of 2em.

# Step 59

Below your .dbl-indent element, add a new p element with the text Includes 10g Added Sugars 20%. Your new p element should also be ***double indented***, and have ***no bottom border***. Use a span to make the 20% bold and right aligned.

Then create another divider after that p element.

# Step 60

After your last divider, create another p element with the text Protein 3g. Use the necessary classes to remove the bottom border, and a span to make the Protein bold.

Following this element, create a large divider.

# Step 61

Create another p element below your large divider. Give the p element the text Vitamin D 2mcg 10%. Use a span to make the 10% align to the right, but do not make it bold.

# Step 62

Create another p element, give it the text Calcium 260mg 20%. Align 20% to the right. Below that, create a p element with the text Iron 8mg 45%, aligning the 45% to the right.

# Step 63

Create the final p element for your .daily-value section. Give it the text Potassium 235mg 6%. Align the 6% text to the right, and remove the bottom border of the p element.

# Step 64

Add a medium divider after your .daily-value element. Below that new divider, create a p element with the class attribute set to note.

Give the p element the following text:

\* The % Daily Value (DV) tells you how much a nutrient in a serving of food contributes to a daily diet. 2,000 calories a day is used for general nutrition advice.

# Step 65

Create a .note selector, and set the size of the font to 0.6rem. Also set the top and bottom margins to 5px, removing the left and right margins.

# Step 66

Give the .note selector a left and right padding of 8px, removing the top and bottom padding. Also set the text-indent property to -8px.

With these last changes, your nutrition label is complete!