**Learn Typograghy by building a nutrition label**

Typography is the art of styling your text to be easily readable and suit its purpose.

In this course, you'll use typography to build a nutrition label webpage. You'll learn how to style text, adjust line height, and position your text using CSS.

What is **Typograghy:**

Typography is the art and technique of arranging type to make written language legible, readable, and visually appealing when displayed. It involves selecting typefaces, point sizes, line lengths, line-spacing (leading), and letter-spacing (tracking), and adjusting the space between pairs of letters (kerning). Typography encompasses both the aesthetic and functional aspects of type, playing a crucial role in the design and communication of printed and digital materials. It includes the following key elements:

1. \*\*Typefaces and Fonts\*\*: A typeface is a set of characters with a common design, while a font is a specific weight, style, and size within a typeface family.

2. \*\*Hierarchy and Emphasis\*\*: Using different sizes, weights, and styles to create a visual hierarchy that guides the reader's attention and emphasizes important information.

3. \*\*Alignment and Spacing\*\*: Proper alignment (left, right, center, or justified) and spacing (between lines and characters) to ensure readability and aesthetic harmony.

4. \*\*Readability and Legibility\*\*: Ensuring text is easy to read and understand, which involves selecting appropriate typefaces and organizing text in a clear and structured manner.

5. \*\*Visual Balance\*\*: Creating a harmonious and aesthetically pleasing layout by balancing various typographic elements.

Typography is essential in various fields such as graphic design, web design, advertising, and publishing, as it significantly impacts how information is perceived and understood.

We've provided a basic HTML boilerplate for you.

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

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

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

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.

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.

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

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

Borders can be used to group and prioritize content.

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

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

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

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.

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.

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

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.

Create a p selector and remove all margins.

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.

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.

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.

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.

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

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

Wrap the text 2/3 cup (55g) in a span element.

Now we can add the horizontal spacing using flex. In your p selector, add a display property set to flex and a justify-content property set to space-between.

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

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

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

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

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

Note: the medium(medium) class will be utilized later for the thinner bars of the nutrition label.

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

Within your .calories-info element, create a div element. Give that div element a class attribute set to left-container. Within the newly created div element, create a h2 element with the text Amount per serving. Give the h2 element a class attribute set to bold small-text.

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

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

Create a .calories-info h2 selector and remove all margins.

Below your .small-text element, create a new p element with the text Calories. Also below the .left-container element, create a new span element with the text 230.

Create a new .calories-info selector and give it a display property set to flex. Also give it a justify-content property set to space-between and align-items property set to flex-end.

Create a new .left-container p selector setting the top and bottom margin to -5px, and the left and right margin to -2px. Also set the font-size to 2em and font-weight to 700.

Create a .calories-info span selector, set its font-size to 2.4em and font-weight to 700.

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 set to -7px -2px. This will shift your 230 text into place.

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

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

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