**Typography**

In this lesson, we’ll focus on *typography*, the art of arranging text on a page. In particular, we’ll look at how to style fonts with CSS to make them legible and appealing and how to add external fonts to your web pages.

Some of the most important information a user will see on a web page will be textual. Styling text to make page content accessible and engaging can significantly improve user experience. Let’s begin!

**Instructions**

In the browser, we have a blog site with fonts of different sizes and styles. In the following exercises, we’ll learn how to manipulate fonts to create engaging interfaces.

Once you have an idea of the general layout of the page, proceed to the next exercise.

# Font Family

If you’ve ever used a formatted word processor, chances are that you probably also used a feature that allowed you change the “type of font” you were typing in. The phrase “type of font” refers to the technical term [typeface](https://en.wikipedia.org/wiki/Typeface), or font family.

To change the typeface of text on your web page, you can use the font-family property.

h1 {

font-family: Garamond;

}

In the example above, the font family for all main heading elements has been set to Garamond.

When setting typefaces on a web page, keep the following points in mind:

1. The font specified in a stylesheet must be installed on a user’s computer in order for that font to display when a user visit the web page. We’ll learn how to work around this issue in a later exercise.
2. You’ve probably noticed that we haven’t been specifying a typeface in previous exercises of this course. How exactly does the browser know what typeface to use when displaying the web page? The default typeface for many browsers is [Times New Roman](https://en.wikipedia.org/wiki/Times_New_Roman). You may be familiar with this typeface if you have ever used a formatted word processor.
3. It’s a good practice to limit the number of typefaces used on a web page to 2 or 3.
4. When the name of a typeface consists of more than one word, it must be enclosed in double quotes (otherwise it will not be recognized), like so:

h1 {

font-family: "Courier New";

}

**Instructions**

**1.**

In **style.css**, change the font family of h1 and h2 elements to Georgia.

**2.**

Next, change the font family of the paragraph to Helvetica.

1 {

  color: white;

  font-family:Georgia;

  font-size: 28px;

}

h2 {

  font-size: 24px;

  font-weight: 700;

  font-family:Georgia;

  line-height: 2.5;

}

p {

  margin: 16px 0;

  font-family: Helvetica;

}

# Font Weight

You’ve probably noticed **bold** text in websites you use, especially in news or text-heavy sites. It’s common to bold important headings or keywords. In CSS, we can style bold text with the font-weight property.

If we want to bold text in a web page, we can set the font-weight to bold.

p {

font-weight: bold;

}

If we want to ensure that text is not bold, we can set the font-weight to normal.

p {

font-weight: normal;

}

By default, the font-weight of most text elements is set to normal. Some elements, like headers, have built-in bold styling. A good approach is to check to see if the the text element has any default styling, and use the font-weight property accordingly.

**Instructions**

**1.**

Taking a look at the web page in the browser, you’ll notice a “Banner” section, the blue section right below the navigation menu.

In **style.css**, in the banner section of the stylesheets, set the font weight of the paragraph within class banner to bold. You’ll notice that the letters in the paragraph in the web page thicken.

banner p {

  border-top: 1px solid #fff;

  border-bottom: 1px solid #fff;

  padding: 10px;

  color: #ffffff;

  font-weight: bold;

}

**Font Weight II**

The font-weight property can also be assigned a number value to style text on a numeric scale ranging from 100 to 900. Valid values are multiples of 100 within this range such as 200 or 500.

When using numeric weights, there are a number of default font weights that we can use:

1. 400 is the default font-weight of most text.
2. 700 signifies a bold font-weight.
3. 300 signifies a light font-weight.

Let’s take a look at an example of how numeric fonts are used.

header {

font-weight: 800;

}

footer {

font-weight: 200;

}

Here, the header would appear as a deep bold, while the footer would appear rather light.

It’s important to note that not all fonts can be assigned a numeric font-weight. You can look up the font you are using to see which font-weight values are available.

**Instructions**

**1.**

The header section of the webpage is where the navigation menu is. It lives at the very top of the page.

In **style.css**, in the header section, set the font weight of the element with class header to 900. You’ll notice the list elements in the navigation thicken.

.header {

  font-family: "Arial", sans-serif;

  font-size: 14px;

  font-weight: 900;

  line-height: 1.25;

  background-color: #fff;

  position: fixed;

  top: 0;

  width: 100%;

}

# Font Style

You can also italicize text with the font-style property.

h3 {

font-style: italic;

}

The italic value causes text to appear in italics. The font-style property also has a normal value which is the default.

**Instructions**

**1.**

The web page features three sections, “Garamond”, “Helvetica”, and “Space Mono”. Each of these sections includes a line with the name of the font creator, such as “Claude Garamond”.

Let’s italicize the creator’s name on each of these cards. In **style.css**, in the font card section, set the font style of .font-card .creator to italic.

Notice the font change from normal to italic on the web page.

.font-card .creator {

  font-size: 20px;

  line-height: 1.25;

  font-style: italic;

}

# Word Spacing

You can also increase the spacing between words in a body of text, technically known as word spacing.

To do so, you can use the word-spacing property:

h1 {

word-spacing: 0.3em;

}

The default amount of space between words is usually 0.25em. In the example above, the word spacing is set to 0.3em, which represents an increase of only .05em in word spacing.

It’s not common to increase the spacing between words, but it may help enhance the readability of bolded or enlarged text. Note, again, that the preferred unit is ems.

**Instructions**

**1.**

In **style.css**, set the word spacing of the h2 elements to 0.05em.

h2 {

  font-size: 24px;

  font-weight: 700;

  font-family: Georgia;

  word-spacing: 0.05em;

  word-spacing: 0.3em;

}

# Letter Spacing

You’ve learned how to increase the spacing between lines of text and words, but it’s possible to get even more detailed: increasing the spacing between individual letters.

The technical term for adjusting the spacing between letters is called tracking. Tracking can be adjusted with the letter-spacing property in CSS.

h1 {

letter-spacing: 0.3em;

}

Like word spacing, it’s not common to increase the tracking in text, but sometimes it enhances the readability of uppercase text.

**Instructions**

**1.**

In **style.css**, set the letter spacing of the h2 elements to 0.02em.

h2 {

  font-size: 24px;

  font-weight: 700;

  font-family: Georgia;

  word-spacing: 0.05em;

  word-spacing: 0.3em;

  letter-spacing: 0.02em;

}

# Text Transformation

Text can also be styled to appear in either all uppercase or lowercase with the text-transform property.

h1 {

text-transform: uppercase;

}

The code in the example above formats all <h1> elements to appear in uppercase, regardless of the case used for the heading within the HTML code. Alternatively, the lowercase value could be used to format text in all lowercase.

Since text can be directly typed in all uppercase or lowercase within an HTML file, what is the point of a CSS rule that allows you to format [letter case](https://en.wikipedia.org/wiki/Letter_case)?

Depending on the type of content a web page displays, it may make sense to always style a specific element in all uppercase or lowercase letters. For example, a website that reports breaking news may decide to format all <h1> heading elements such that they always appear in all uppercase, as in the example above. It would also avoid uppercase text in the HTML file, which could make code difficult to read.

**Instructions**

**1.**

In **style.css**, transform the text in the main heading (h1) to appear uppercase.

h1 {

  color: white;

  font-family:Georgia;

  font-size: 28px;

  text-transform: uppercase;

}

**Text Alignment**

No matter how much styling is applied to text (typeface, size, weight, etc.), text always appears on the left side of the browser.

To move, or align, text, we can use the text-align property.

h1 {

text-align: right;

}

The text-align property can be set to one of the following three values:

1. left - aligns text to the left hand side of the browser.
2. center - centers text.
3. right - aligns text to the right hand side of the browser.

Later in the course, you’ll learn exactly how the browser positions HTML elements by default, which will help you understand how the browser “aligns” text, since “align” is a relative term. For now, it’s enough to know that text can be moved to the left, center, or right side of the web page.

**Instructions**

**1.**

In **style.css**, set the text-align property of the main heading (h1) so that it appears on the left.

**2.**

On second thought, the heading looks better in the center. Reset the heading so that it is aligned to the center.

1 {

  color: white;

  font-family:Georgia;

  font-size: 28px;

  text-transform: uppercase;

  text-align: Left;

}

# Line Height Anatomy

Another property that we can set for text is line-height. This property modifies the leading of text.

The diagram to the right helps illustrate exactly what the terms “leading” and “line height” mean.



**Line Height**

We often modify line-height to make text on a web page easier to read. When text is styled to appear larger, the vertical spacing between lines of text can decrease, creating text that is difficult to read, particularly in paragraphs.

We can use the line-height property to set how tall we want the line containing our text to be, regardless of the height of the text. Line heights can take one of several values:

1. A unitless number, such as 1.2. This number is an absolute value that will compute the line height as a ratio of the font size.
2. A number specified by unit, such as 12px. This number can be any valid CSS unit, such as pixels, percents, ems, or rems.

Generally, the unitless ratio value is the preferred method, since it is responsive and based exclusively on the current font size. In other words, if we change the font size, a unitless line-height would automatically readjust, whereas the pixel value would remain static.

p {

line-height: 1.4;

}

**Instructions**

**1.**

Let’s change the line height of the text within the banner.

In **style.css**, set the line height of the paragraph element within the banner class to 1.4. You’ll notice the line height increase.

.banner p {

  border-top: 1px solid #fff;

  border-bottom: 1px solid #fff;

  padding: 10px;

  color: #ffffff;

  font-weight: bold;

  line-height: 1.4;

}

**Serif and Sans Serif**

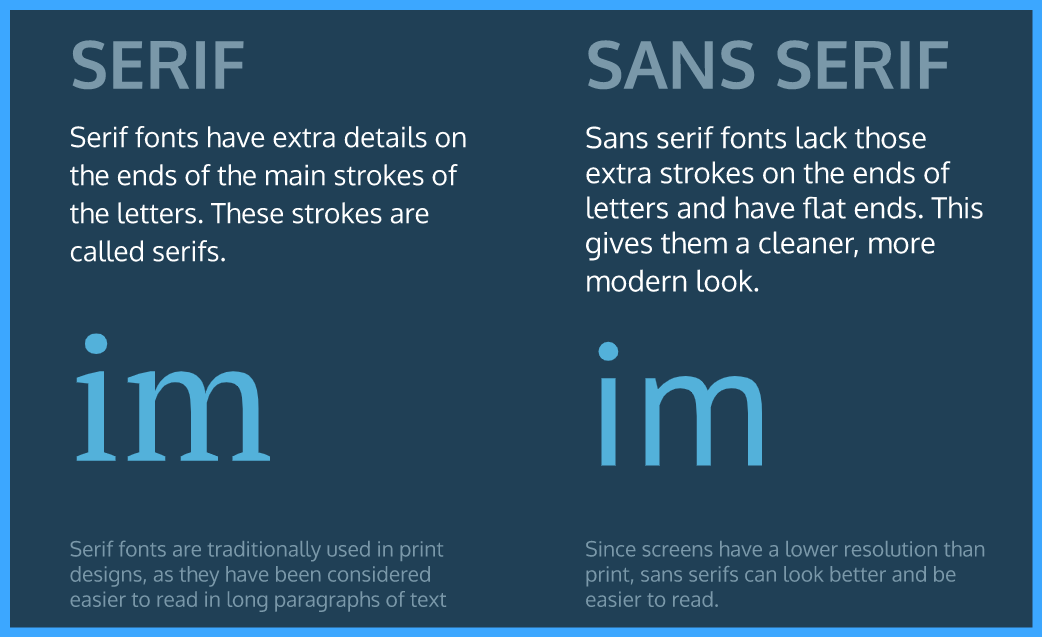
You’ve learned a lot of properties to modify text on a web page!

In the next exercise, you’ll set some text to be *serif* and some text to be *sans-serif*. What exactly do these words mean?

1. Serif — fonts that have extra details on the ends of each letter. Examples include fonts like Times New Roman or Georgia, among others.
2. Sans-Serif — fonts that do not have extra details on the ends of each letter. Instead, letters have straight, flat edges, like Arial or Helvetica.

**Instructions**

Notice the difference between serif fonts and sans-serif fonts in the diagram to the right. When you’re done, proceed to the next exercise.



**Fallback Fonts**

What happens when a stylesheet requires a font that is not installed on a user’s computer? Most computers have a small set of typefaces pre-installed. This small set includes serif fonts like Times New Roman and sans-serif fonts like Arial.

These pre-installed fonts serve as *fallback fonts* if the stylesheet specifies a font which is not installed on a user’s computer.

To use fallback fonts, the following syntax is required:

h1 {

font-family: "Garamond", "Times", serif;

}

The CSS rule above says:

1. Use the Garamond font for all <h1> elements on the web page.
2. If Garamond is not available, use the Times font.
3. If Garamond and Times are not available, use any serif font pre-installed on the user’s computer.

The fonts specified after Garamond are the fallback fonts (Times, serif). Fallback fonts help ensure a consistent experience for the diverse audience of users that visit a site.

**Instructions**

**1.**

Below the banner, the web page is made of three main font sections, Serif, Sans-Serif and Monospace. Each of these sections shows an example font, Garamond, Helvetica, and Space Mono, respectively.

We’ll focus on styling the Garamond section here. In **style.css**, in the “Font Card” section, create the selector .garamond .sample .text.

**2.**

Using the selector you just created, set the font to Garamond and add serif as a fallback font.

**3.**

In **style.css**, below the last selector you made, create the selector .helvetica .sample .text.

**4.**

Using the selector you just created, set the font to Helvetica and add sans-serif as a fallback font.

.garamond .sample .text {

  font-family: Garamond, serif;

}

# Linking Fonts I

With the number of fonts available with modern typography, it is unrealistic to expect users to have all fonts installed on their computers. New fonts are often centralized in directories made available for public use. We refer to these fonts as non-user fonts.

[Google Fonts](https://fonts.google.com/) is one such directory of thousands of open-source fonts, available for free use. Google Fonts gives us a way to retrieve the link for a single font, multiple fonts, or multiple fonts with the font-weight and font-style properties.

We’ll show you where to add this link in the next exercise.

**Instructions**

1. View the video in this exercise.

The video demonstrates how to retrieve the link for a single font, multiple fonts, and multiple fonts of various numeric font weights and font styles.

2. Once you have viewed the video, repeat these steps in the browser on your desktop computer, selecting the Space Mono font specifying both the regular (400) and bold (700) font-weights.

When you are ready, click ‘Next’ to continue.

# Linking Fonts II

When we have the link to the font of our choice, we can add the font to the <head> section of the HTML document, using the <link> tag and the href.

Let’s take a look at a few examples:

1. A single linked font, using Droid Serif as an example:

<head>

<link href="https://fonts.googleapis.com/css?family=Droid+Serif" type="text/css" rel="stylesheet">

</head>

2. Multiple linked fonts, using the Droid Serif and Playfair Display fonts as an example:

<head>

<link href="https://fonts.googleapis.com/css?family=Droid+Serif|Playfair+Display" type="text/css" rel="stylesheet">

</head>

3. Multiple linked fonts, along with weights and styles. Here Droid Serif has font weights of 400, 700, and 700i, while Playfair Display has font weights of 400, 700, and 900i:

<head>

<link href="https://fonts.googleapis.com/css?family=Droid+Serif:400,700,700i|Playfair+Display:400,700,900i" rel="stylesheet">

</head>

Once a font is linked, we can create CSS selectors to target elements, just as we do with other fonts.

**Instructions**

**1.**

In the previous exercise, you surfaced the link to the Space Mono font specifying both the normal (400) and bold (700) font weights.

In **index.html** include this font in the project using the <link> tag.

Hint

The URL to reference in the link tag is:

https://fonts.googleapis.com/css?family=Space+Mono:400,700

If you’ve directly copied and pasted from Google fonts it may contain an additional [query string parameter](https://en.wikipedia.org/wiki/Query_string#Structure), so your full link maybe be:

https://fonts.googleapis.com/css?family=Space+Mono:400,700&display=swap

That’s ok too, both links work!

**2.**

In **style.css**, in the “Font Card” section, create the selector .space .sample .text.

Hint

The URL to reference in the link tag is:

https://fonts.googleapis.com/css?family=Space+Mono:400,700

If you’ve directly copied and pasted from Google fonts it may contain an additional [query string parameter](https://en.wikipedia.org/wiki/Query_string#Structure), so your full link maybe be:

https://fonts.googleapis.com/css?family=Space+Mono:400,700&display=swap

That’s ok too, both links work!

**3.**

Using the selector you just created, set the font to "Space Mono" and add monospace as a fallback font.

.space .sample .text {

  font-family: "Space Mono", monospace;

}

# Font-Face I

There are other ways to link non-user fonts that don’t require the use of the <link> tag in the HTML document. CSS offers a way to import fonts directly into stylesheets with the @font-face property.

To load fonts with the @font-face property:

1. Instead of using the font’s link in the HTML document, enter the link into the URL bar in the browser.
2. The browser will load the CSS rules. You will need to focus on the rules that are directly labeled as /\* latin \*/. Some of the latin rules are on separate lines. You will need each of these.
3. Copy each of the CSS rules labeled latin, and paste the rules from the browser to the top of **style.css**.

It is important to stress the need to copy the @font-face rules to the top of the stylesheet for the font to load correctly in the project.

**Instructions**

1. View the video in this exercise.

The video demonstrates how to enter the Space Mono font link you previously retrieved into the browser. The browser in turn displays the rules you will use for the font.

2. Once you have viewed the video, repeat these steps in the browser on your desktop computer, with link for the Space Mono font with normal (400) and bold (700) font-weights. You can find the direct link to the font [here](https://fonts.googleapis.com/css?family=Space+Mono:400,700/).

Once your desktop browser displays the rules, click ‘Next’ to continue.

# Font-Face II

We can then use the fonts in the stylesheets as you would use any other font. Let’s practice loading an external font in our stylesheets using the @font-face property, and using the font to style our page.

**Instructions**

**1.**

In **index.html** remove the link to the Space Mono font that you added in the previous exercise.

**2.**

Now, add this same font to **style.css** using the @font-face rules labeled latin.

Recall that in order to do this, we use the same CSS rules returned in the browser for the Space Mono font, and copy the latin @font-face rules to the top of the stylesheet.

If you need help retrieving the link for the appropriate font, click [here](https://fonts.googleapis.com/css?family=Space+Mono:400,700).

Once you add the font, you’ll notice that the elements styled in the Space Mono font retain the correct font styling.

# Font-Face II

We can then use the fonts in the stylesheets as you would use any other font. Let’s practice loading an external font in our stylesheets using the @font-face property, and using the font to style our page.

**Instructions**

**1.**

In **index.html** remove the link to the Space Mono font that you added in the previous exercise.

**2.**

Now, add this same font to **style.css** using the @font-face rules labeled latin.

Recall that in order to do this, we use the same CSS rules returned in the browser for the Space Mono font, and copy the latin @font-face rules to the top of the stylesheet.

If you need help retrieving the link for the appropriate font, click [here](https://fonts.googleapis.com/css?family=Space+Mono:400,700).

Once you add the font, you’ll notice that the elements styled in the Space Mono font retain the correct font styling.

# Font-Face III

While Google Fonts and other resources can broaden font selection, you may wish to use an entirely different font or abstain from using a font from an external service.

We can modify our @font-face rule to use local font files as well. We can supply the user with the desired font family and host it along with our site instead of depending on a different site.

@font-face {

font-family: "Roboto";

src: url(fonts/Roboto.woff2) format('woff2'),

url(fonts/Roboto.woff) format('woff'),

url(fonts/Roboto.tff) format('truetype');

}

Here, you’ll notice:

1. The main difference is the use of a relative filepath instead of a web URL.
2. We add a format for each file to specify which font to use. Different browsers support different font types, so providing multiple font file options will support more browsers.

As of now .woff2 appears to be the way of the future, due to greatly reduced file sizes and improved performance, but many browsers still don’t support it. There are lots of great sources to find fonts to use locally, such as [Font Squirrel](https://www.fontsquirrel.com/).

**Instructions**

**1.**

In the **fonts/** directory, you’ll notice that we have added several local font files.

Let’s change the typography of the banner, using local font files. If you open up the **fonts/** directory using the file navigator in the code editor, you’ll notice that we have added local font files Glegoo-Regular.ttf and Glegoo-Bold.ttf.

At the top of **style.css**, create a selector using the @font-face property and give it the font family Glegoo.

**2.**

Within the @font-face rule, add a src attribute with the relative path of the file Glegoo-Regular.ttf and a format of truetype. Make sure to include this path in the url() parentheses.

Hint

The syntax to include a path and format with a @font-face src looks like this:

@font-face {

src: url("relative-path") format('format');

}

The relative path for Glegoo-Regular.ttf is

../fonts/Glegoo-Regular.ttf

**3.**

Using the selector that targets the paragraph nested within the banner class, add the font family Glegoo and a font size of 20px.

Reload the page to see the changes in the browser.

@font-face {

  font-family: "Glegoo";

  src: url(../fonts/Glegoo-Regular.ttf) format('truetype');

}

# Review

Great job! You learned how to style an important aspect of the user experience, typography.

Let’s review what you’ve learned so far:

* Typography is the art of arranging text on a page.
* Text can appear in any number of weights, with the font-weight property.
* Text can appear in italics with the font-style property.
* The vertical spacing between lines of text can be modified with the line-height property.
* Serif fonts have extra details on the ends of each letter. Sans-Serif fonts do not.
* Fallback fonts are used when a certain font is not installed on a user’s computer.
* Google Fonts provides free fonts that can be used in an HTML file with the <link> tag or the @font-face property.
* Local fonts can be added to a document with the @font-face property and the path to the font’s source.
* The word-spacing property changes how far apart individual words are.
* The letter-spacing property changes how far apart individual letters are.
* The text-align property changes the horizontal alignment of text.