**Typography**

Typography is the art of arraging text on a page.

1. **Font Family**

The font of an element can be changed using the font-family property.

**Web Safe Fonts**

There is a selection of fonts that will appear the same across all browsers and operating systems. These fonts are referred to as *web safe fonts*. You can check out a complete list of web safe fonts [here](https://www.cssfontstack.com/).

**Fallback Fonts and Font Stacks**

Web safe fonts are good *fallback fonts* that can be used if your preferred font is not available.

h1 {  
  font-family: Caslon, Georgia, 'Times New Roman',serif;  
}

🡪 Georgia will be used if Caslon not available. Times will be used if Georgia not available. These are known as a *font stack*.

**Serif and Sans-Serif**

You may be wondering what features make a font similar to another font. Serif fonts have extra details on the ends of each letter, as opposed to *Sans-Serif* fonts, which do not have the extra details.

Graphical user interface, website

Description automatically generated

serif and sans-serif are also keyword values that can be added as a final fallback font if nothing else in the font stack is available.

1. **Font Weight**

In CSS, the font-weight property controls how bold or thin text appears. It can be specified with keywords or numerical values.

**Keyword Values**

The font-weight property can take any one of these keyword values:

* bold: Bold font weight.
* normal: Normal font weight. This is the default value.
* lighter: One font weight lighter than the element’s parent value.
* bolder: One font weight bolder than the element’s parent value

**Numerical Values**

Numerical values can range from 1 (lightest) to 1000 (boldest).

It’s important to note that not all fonts can be assigned a numeric font weight, and not all numeric font weights are available to all fonts.

1. **Font Style**

You can also italicize text with the font-style property. Default value is normal

h3 {

font-style: italic;

}

1. **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.

1. **Text Layout**

Now you’ll learn about some ways text can be displayed or laid out within the element’s container.

**Letter Spacing**

The letter-spacing property is used to set the horizontal spacing between the individual characters in an element.

The letter-spacing property takes length values in units, such as 2px or 0.5em

**Word Spacing**

You can set the space between words with the word-spacing property.

The word-spacing property also takes length values in units, such as 3px or 0.2em

**Line Height**

A picture containing calendar

Description automatically generated

We can use the line-height property to set how tall we want each line containing our text to be. Line height values can be a unitless number, such as 1.2, or a length value, such as 12px, 5% or 2em.

If the line-height is specified by a unitless number, changing the font size will automatically readjust the line height.

**Text Alignment**

The text-align property aligns text to its parent element.

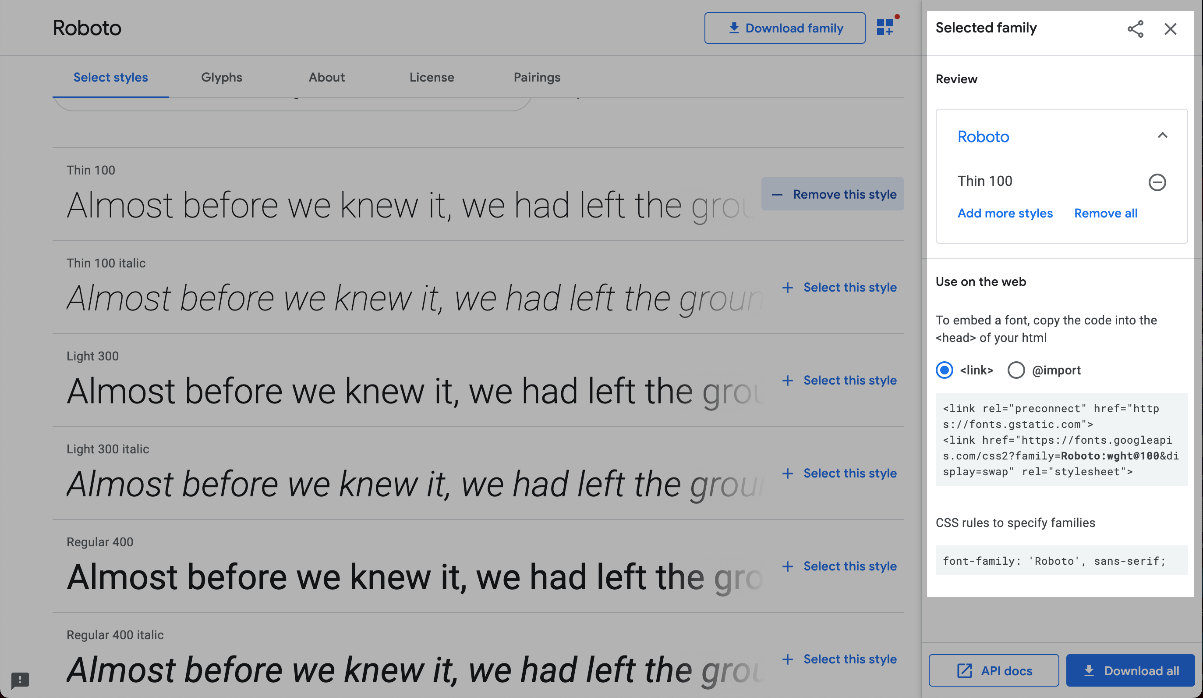
h1 {  
  text-align: right;  
}

1. **Web Fonts**

We previously knew that browsers have safe fonts that we can use. However, we are not limited to these fonts. The number of fonts we can use are actually limitless, since we can link our stylesheet with online web fonts (or downloading and hosting fonts)

**Web Fonts Using <link>**

Online font services, like [Google Fonts](https://fonts.google.com/), make it easy to find and link to fonts from your site. You can browse and select fonts that match the style of your website.



The generated <link> element needs to be added to the <head> element in your HTML document for it to be ready to be used in your CSS.

<head>  
   <!-- Add the link element for Google Fonts along with other metadata -->  
   <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@100&display=swap" rel="stylesheet">  
</head>

🡪 Now we can declare this font from our stylesheet, just like other fonts.

**Web Fonts Using @font-face**

Fonts can also be added using a [@font-face ruleset](https://developer.mozilla.org/en-US/docs/Web/CSS/@font-face) in your CSS stylesheet instead of using a <link> element in your HTML document. As mentioned earlier, fonts can be downloaded just like any other file on the web. They come in a few different file formats, such as:

* OTF (OpenType Font)
* TTF (TrueType Font)
* WOFF (Web Open Font Format)
* WOFF2 (Web Open Font Format 2)

The different formats are a progression of standards for how fonts will work with different browsers, with WOFF2 being the most progressive. It’s a good idea to include TTF, WOFF, and WOFF2 formats with your @font-face rule to ensure compatibility on all browsers.

When you have the files you need, move them to a folder inside your website’s working directory, and you’re ready to use them in a @font-face ruleset

@font-face {

font-family: 'MyParagraphFont';

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

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

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

}

* The @font-face at-rule is used as the selector. It’s recommended to define the @font-face ruleset at the top of your CSS stylesheet.
* Inside the declaration block, the font-family property is used to set a custom name for the downloaded font. The name can be anything you choose, but it must be surrounded by quotation marks. In the example, the font is named 'MyParagraphFont', as this font will be used for all paragraphs.
* The src property contains three values, each specifying the relative path to the font file and its format. In this example, the font files are stored inside a folder named fonts within the working directory.
* Note that the ordering for the different formats is important because our browser will start from the top of the list and search until it finds a font format that it supports. Read more on format prioritization on [CSS-Tricks](https://css-tricks.com/snippets/css/using-font-face-in-css/).

Once the @font-face at-rule is defined, you can use the font in your stylesheet

p {  
  font-family: 'MyParagraphFont', sans-serif;  
}