



INM120

Introduction to Adaptive Web Design



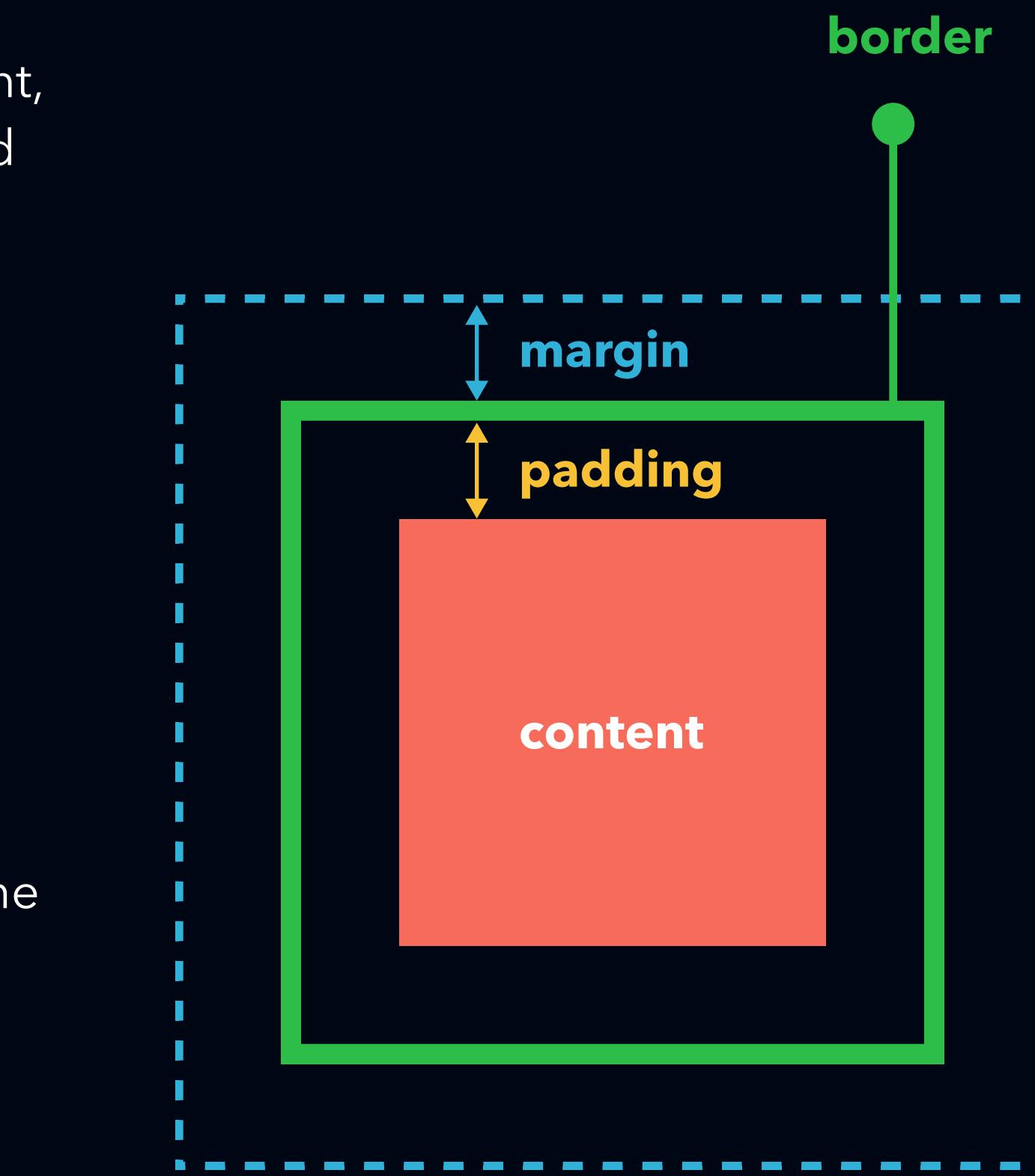
CSS REVIEW

Let's take a moment to revisit what we've learned about CSS before diving into more exciting topics.

CSS BOX MODEL

The Box Model is a fundamental concept in web design and CSS that describes how elements on a web page are **rendered** and how their **dimensions** are calculated. It visualizes every HTML element as a rectangular box with four core components:

- **Content:** The actual content of the HTML element, such as text, images, or other media, is contained within the innermost part of the box.
- **Padding:** The **space between the content** and **the element's border** is known as padding. Padding provides spacing and separation between the content and the border.
- **Border:** The border is a **visible or invisible boundary** that surrounds the padding and content. It defines **the edges** of the element.
- **Margin:** The margin is **the outermost layer** of the box, creating **space between the border** of the element and **adjacent elements** on the page. Margins are used to control the layout and spacing between elements.



SETTING ORDER

the order in which you set the border, padding, and margin properties within a style rule can have a significant impact on the layout and appearance of elements. The order typically follows the pattern of **top**, **right**, **bottom**, and **left**.

```
p { padding: 10px};
```

10px padding all 4 sides

```
p { padding: 10px 20px };
```

10px top & bottom - 20px right & left

```
p { padding: 10px 20px 30px };
```

10px top - 20px right & left - 30px bottom

```
p { padding: 10px 20px 30px 40px };
```

10px top - 20px right - 30px bottom - 10px left



BORDER STYLES

You can apply various properties to control the appearance and behaviour of borders around HTML elements.

1. **border-width:** Specifies the **width** of the border.
2. **border-style:** Determines the **style** of the border.
Common values include **solid**, **dotted**, **dashed**, **double**, **groove**, **ridge**, **inset**, and **outset**.
3. **border-color:** Sets the **color** of the border.
4. **border:** A shorthand property that allows you to **set all border properties** (**width**, **style**, and **color**) in **one declaration**.
5. **border-radius:** Defines **rounded corners** for an element.

GOOD TO KNOW:

box-sizing: border-box; is a CSS property that controls how the **total width and height** of an element are calculated. When you apply `box-sizing: border-box;` to an element, it includes the element's padding and border in its total width and height, rather than adding them to the dimensions as extra space.

CSS COLOURS

Colors are specified using various methods, including **color names**, **hexadecimal** values, **RGB** values, **HSL** values, and more.

- **Color Names:** A set of predefined color names that you can use directly.
- **Hexadecimal Values:** Hexadecimal color codes represent colors using a combination of **six hexadecimal digits** (0-9 and A-F), preceded by a hash symbol (#).
- **RGB Values:** RGB (Red, Green, Blue) color values specify colors using three integers representing the amounts of red, green, and blue, respectively. Each component can have a value between 0 and 255.
- **RGBA Values:** RGBA is an extension of RGB that includes an additional **alpha** component to control **opacity**. Alpha is a value between 0 (completely transparent) and 1 (fully opaque).
- **HSL Values:** HSL (Hue, Saturation, Lightness) color values specify colors based on their hue, saturation, and lightness. **hue** is a degree **value** (0-360), **saturation** is a **percentage** (0-100%), and **lightness** is a **percentage** (0-100%).
- **HSLA Values:** Similar to RGBA, HSLA includes an alpha component for opacity.
- **Color Picker Tools:** Many code editors and design tools offer color picker tools that allow you to visually select and customize colors, providing the corresponding color code.

CSS FONTS

You can control the font of text on your web page using the `font` property. The `font` property allows you to specify various aspects of the font, including the font **family**, font **size**, font **weight**, font **style**, and more.

- **font-family:** Specifies the font family to be used for the text. You can provide a list of font names separated by commas. The browser will use the first available font in the list.

```
font-family: Arial, Helvetica, sans-serif;
```

- **font-size:** Sets the size of the font. You can use various units such as pixels **px**, **ems (em)**, **percentages (%)**, or keywords like `small`, `medium`, `large`, etc.

```
font-size: 16px;
```

- **font-weight:** Determines the **thickness** of the font. You can use values like `normal`, `bold`, `bolder`, or numeric values like `400`, `700`, etc.

```
font-weight: bold;
```

- **font-style:** Defines the style of the font, such as `normal`, `italic`, or `oblique`.

```
font-style: italic;
```



CSS FONTS

- **line-height:** Sets the height of a line and influences the **spacing between lines** of text.
- **text-transform:** Controls the **capitalization** of text. Values like `uppercase`, `lowercase`, and `capitalize` can be used to change the text case.
- **text-decoration:** Adds **decorative** styling to text, such as **underlines**, **overlines**, or **strikethroughs**.
- **color:** Sets the **color** of the text. You can use color names, hexadecimal values, RGB values, or other valid color representations.

```
p {  
    font-family: Arial, Helvetica, sans-serif;  
    font-size: 25px;  
    font-weight: bold;  
    font-style: italic;  
    line-height: 6;  
    text-transform: uppercase;  
    text-decoration: underline;  
    color: #F7C035;  
}
```

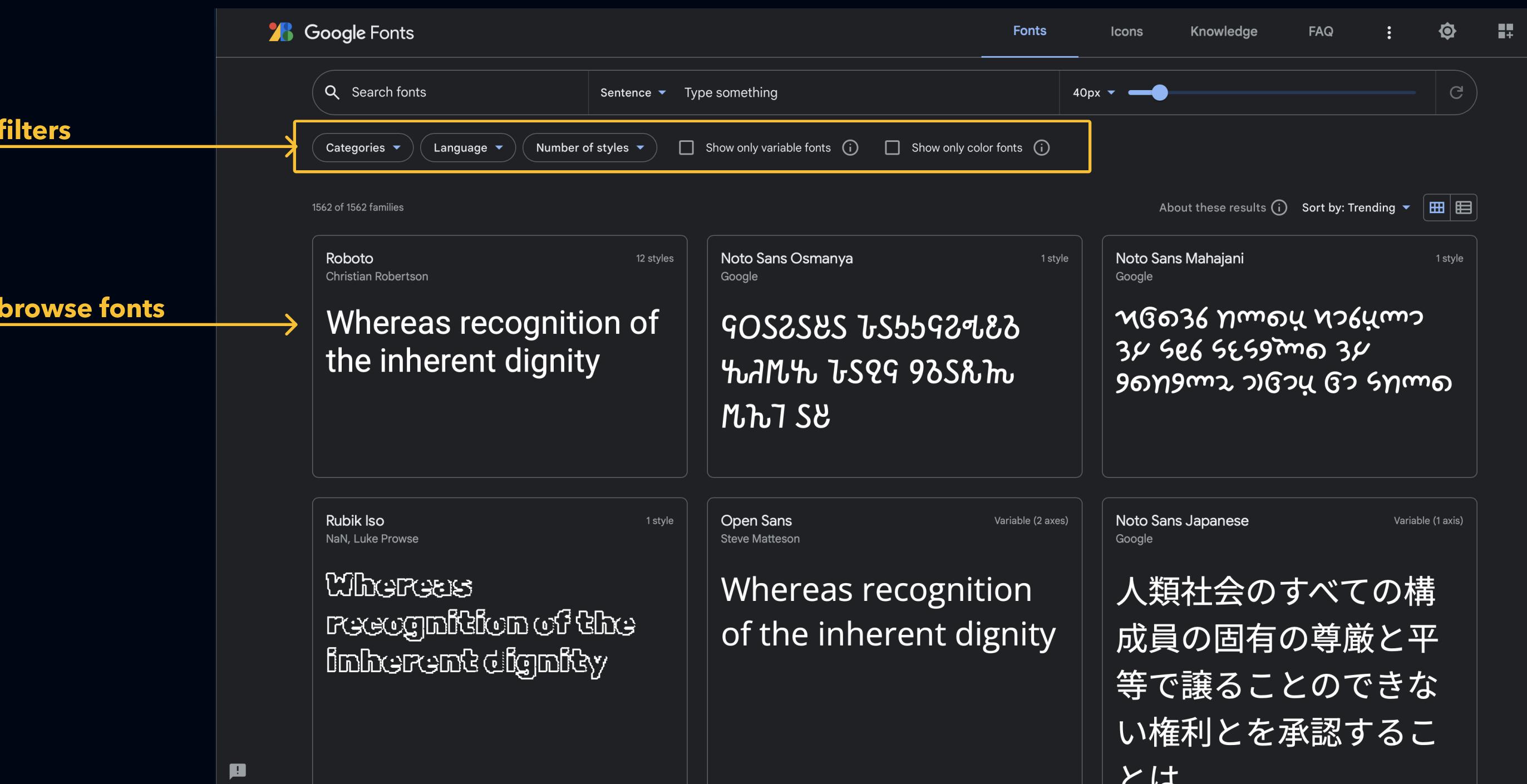


**THIS IS THE RESULT
OF THE PARAGRAPH**

GOOGLE FONTS

To add Google Fonts to your project, follow these steps:

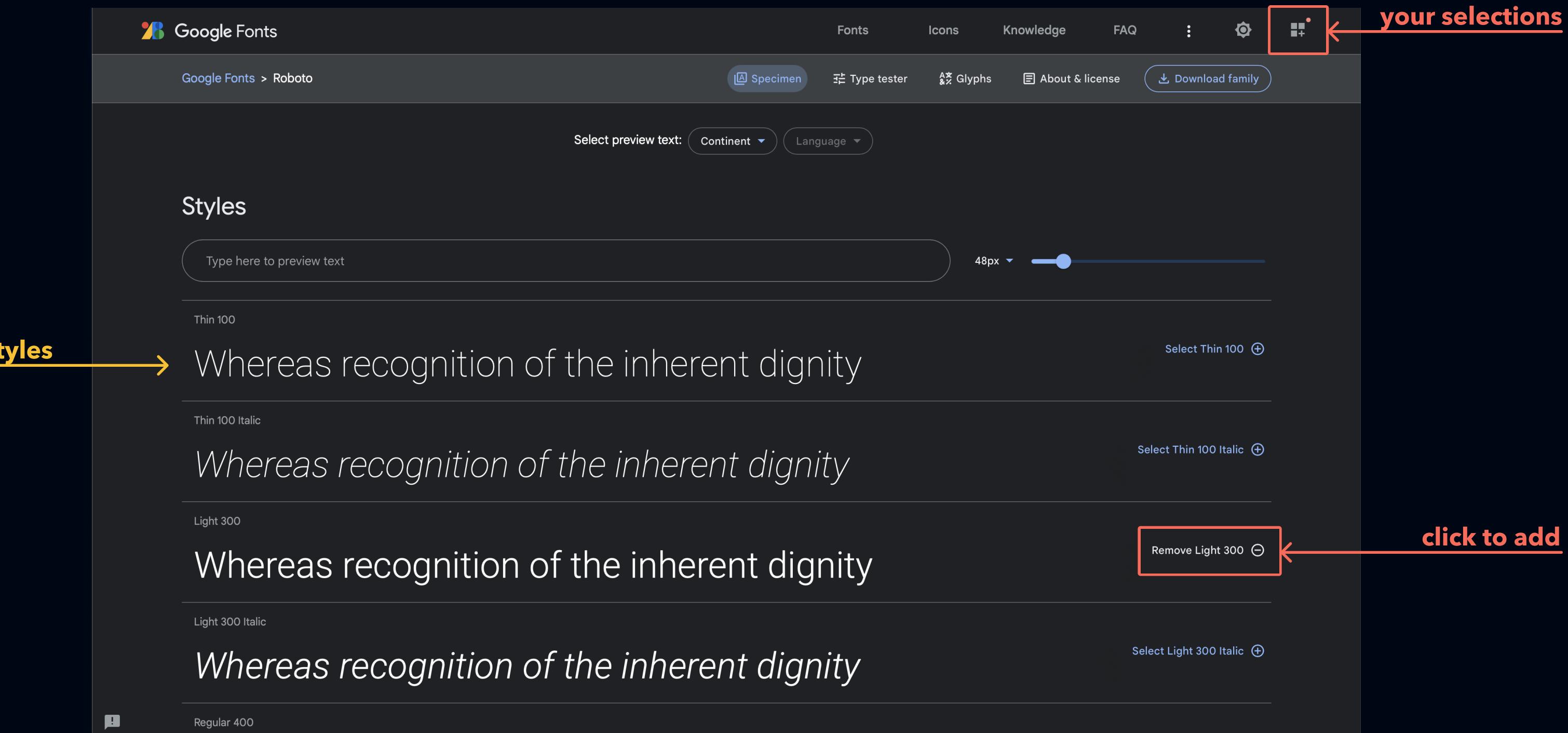
Select Google Fonts: Visit the [Google Fonts website](#), where you can browse and choose from a wide range of fonts. You can search for specific fonts or browse through categories. Click on the fonts you want to use to select them.



GOOGLE FONTS

To add Google Fonts to your project, follow these steps:

Select Google Fonts: Visit the [Google Fonts website](#), where you can browse and choose from a wide range of fonts. You can search for specific fonts or browse through categories. Click on the fonts you want to use to select them.



GOOGLE FONTS

To add Google Fonts to your project, follow these steps:

Select Google Fonts: Visit the [Google Fonts website](#), where you can browse and choose from a wide range of fonts. You can search for specific fonts or browse through categories. Click on the fonts you want to use to select them.

The screenshot shows the Google Fonts interface for the Roboto font family. On the left, under 'Styles', there are preview sections for 'Thin 100' (disabled), 'Thin 100 Italic' (disabled), 'Light 300' (selected), and 'Regular 400'. Each section contains a text input field with placeholder text 'Whereas recognition of the inherent dignity' and a 'Select' button. On the right, the 'Selected family' panel shows 'Roboto' and 'Light 300' selected. It includes options to 'Add more styles', 'Remove all', and 'Review'. Below this, the 'Use on the web' section provides code snippets for embedding the font via '' or '@import'. The 'CSS rules to specify families' section contains the CSS rule 'font-family: 'Roboto', sans-serif;'. Red annotations with arrows point to various parts of the interface:

- A red box surrounds the 'Review' section with the text "review your selections".
- A red box surrounds the 'Remove all' button in the 'Selected family' panel with the text "remove from collection".
- A red box surrounds the radio buttons for '' and '@import' with the text "methods to add to your project".
- A red box surrounds the copy icon in the 'Use on the web' section with the text "copy for HTML".
- A red box surrounds the copy icon in the 'CSS rules to specify families' section with the text "copy for CSS".

CSS UNITS & MEASUREMENTS

In CSS, there are various units and measurements you can use to define the size and spacing of elements on a web page.

- **Pixels (px):** The most commonly used unit, a pixel **represents a single dot on a screen**. It provides a **fixed** and **precise** measurement.
- **Ems (em):** The **em** unit is **relative to the font size of the parent element**. For example, if the parent element has a font size of **16px**, **1em** is equivalent to **16px**. It allows for more flexible and scalable designs.
- **Percentages (%):** Percentages are relative units often **used for responsive design**. They are **based on the parent element's size** or the container's size.
- **Viewport Measurements (vw & vh):** These units are **relative to the viewport's width and height**, respectively. They are **useful for creating responsive layouts** based on the screen size.
- **Points (pt):** These unit is typically used **for print styles** and are **based on typographic measurements**.
- **Rem (root em):** Similar to em, but it's relative to the font size of the **root element** (usually the `<html>` element). It's a more consistent unit for scaling text across the entire page.

CSS POSITIONS

In CSS, the **position** property is used to control the **positioning** of elements within a web page. Here are the most common values:

- **static:** This is the **default value**. Elements with position: static; are displayed in their **normal order** in the **document flow**. They are not affected by the top, right, bottom, or left properties.
- **relative:** Elements with position: relative; are positioned **relative to their normal position** in the document flow. You **can use** the top, right, bottom, or left properties to adjust their position from where they would normally appear.
- **absolute:** Elements with position: absolute; are **positioned relative to their closest positioned ancestor**. If there is no positioned ancestor, they are positioned relative to the **initial containing block (usually the viewport)**. These elements are removed from the normal document flow, and other elements can **overlap** them.
- **fixed:** Elements with position: fixed; are **positioned relative to the viewport, regardless of scrolling**. These elements stay in the same position even if you scroll the page. Commonly used for fixed navigation bars or elements.
- **sticky:** Elements with position: sticky; are **initially in the normal flow**, but they **become sticky** based on their position when scrolling reaches a specified point. You can use top, right, bottom, or left properties to determine the sticky point.