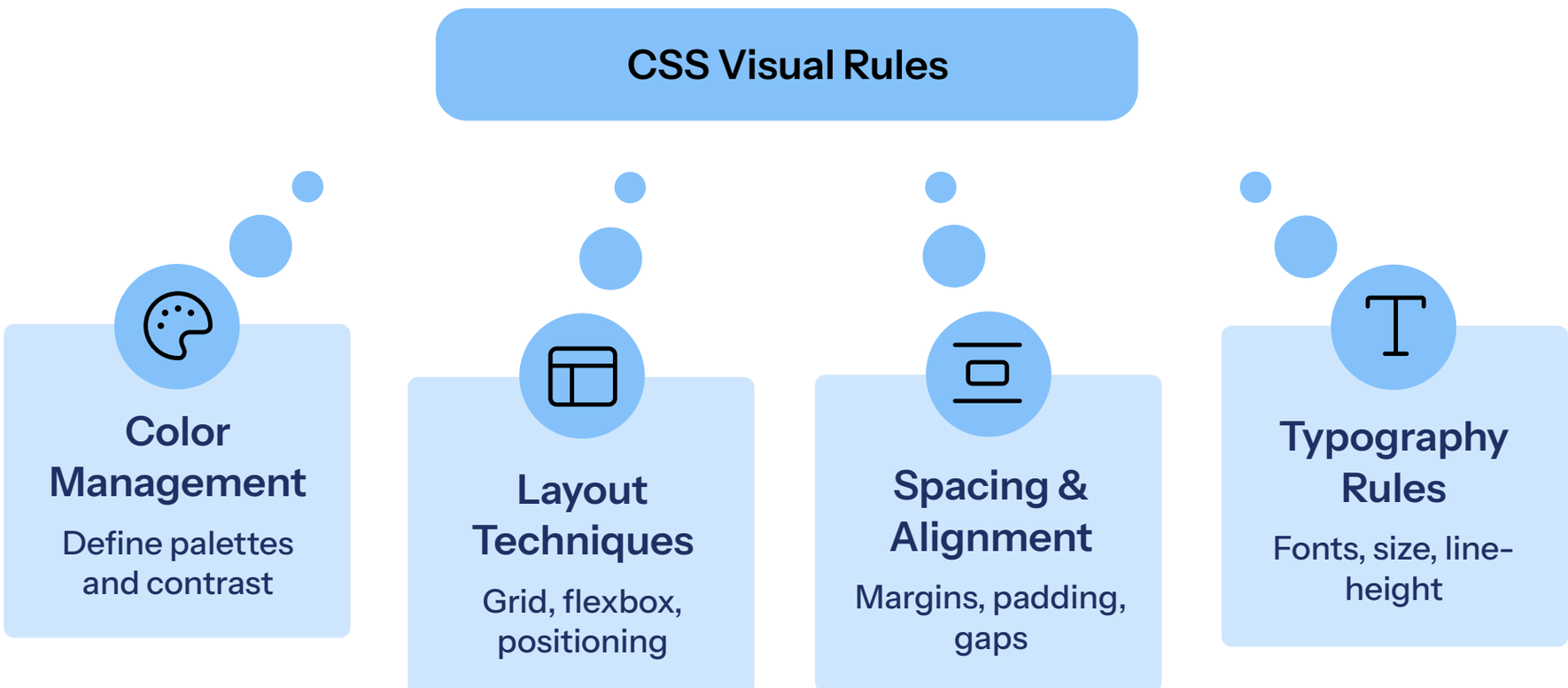
CSS : Visual Rules

Lec-2

Visual Rules in CSS

- Visual rules in CSS are fundamental directives that **govern the aesthetic presentation and arrangement of content on a webpage**, transforming raw HTML into a rich and interactive user experience.
- They are crucial for modern web development as **they control every aspect of an element's appearance, including its color scheme, spatial layout, internal and external spacing, and typographic styling.**
- These categories—color, layout, spacing, and typography—do not operate in isolation; instead, **they work synergistically to create cohesive and visually harmonious designs.**



Units in CSS

- **Units in CSS** are of 2 types:

1. **Absolute Units:** Always fixed irrespective of devices and screen measurements.

Eg: **mm, em, inch (in), pixels(px), points(pt), picas(pc)**

2. **Relative Units:** Not fixed, depends or varies according to device and screen measurements.

Eg: **vh** → view port hieght, **vw** → view port width, **%** → depends on parent element

1 em = 10 mm

1 in = 2.54 em

1 px = 1/96 of 1 in

1 pt = 1/72 of 1 px

1 pc = 12 pt

CSS Typography

- Typography in CSS **refers to the styling and arrangement of text on a webpage using various CSS properties.** These properties **allow control over how text appears,** its size, weight, spacing, and overall layout.
- CSS Properties for Typography are of two types:

1. **Text Properties**
2. **Font Properties**

Text Properties

1. text-align:

- No matter how much styling is applied to the text, the text appears always on the left side of the container in which it resides.
- To align the text, we use **text-align** property.
- The property will align the text of that element or parent which holds it.
- The values can be given to this property are ad follows:

Value	Meaning
text-align: left;	Aligns text to the left side of the browser.
text-align: center;	Aligns text to the center of the browser.
text-align: right;	Aligns text to the right side of the browser.
text-align: justify;	Spaces out text in order to align with right and left side of the parent element or browser.

Text Properties

2. text-transform:

- Text can also be styled to appear in either all **uppercase** or all **lowercase** with this property.
- It takes two values:

(i) text-transform: Uppercase;

(ii) text-transform: Lowercase;

3. letter-spacing:

- The **letter-spacing** property is **used to set the horizontal spacing** between the individual character in an element.
- Sometimes, it helps with the **readability** of certain fonts or styles.
- The **letter-spacing** property takes length values in units such as **px** or **em/rem.**
- Eg:

```
p{
    letter-spacing: 2px;
 }
```

- Above eg will add 2px of space between the letters in paragraph element.

```
p.demo{
        letter-spacing: -0.3em;
   }
```

- Above eg will reduce 0.3em of space between the letters of paragraph element with class **demo**.

Text Properties

4. word-spacing:

- **word-spacing** is the property used to set space between words.
- It helps enhance the readability of bolded or enlarged text.
- The **word-spacing** property also takes length values in units as **px** or **em/rem.**
- **Eg:**

```
h1{
   word-spacing: 0.5em;
 }
```

- Above eg will add 0.5em of spacing between the content of h1 element.

For **word-spacing** using **em** values are recommended because the spacing can be set based on the size of the font.

5. line-height:

- 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** or a **length value** in **px,%** or **em/rem.**
- **Eg:**

```
p{
   line-height: 1.4;
 }
```

Generally, the unitless values are preferred for **line-height** property, since it is responsive based on the current font size, changing the font size will automatically readjust the line height if declared unitlessly.

Font Properties

1. font-family:

- **Font** or **Typeface** of an element can be changed using **font-family** property,
- Eg:

```
h1{
   font-family: Arial;
 }
```

- By above shown example we can add **single word values of font family**.
- Eg:

```
h1{
   font-family: 'Times New Roman';
 }
```

- By above shown example we can add **multi word values font family.**
- Eg:

```
h1{
   font-family: Calson, Georgia, 'Times New Roman';
 }
```

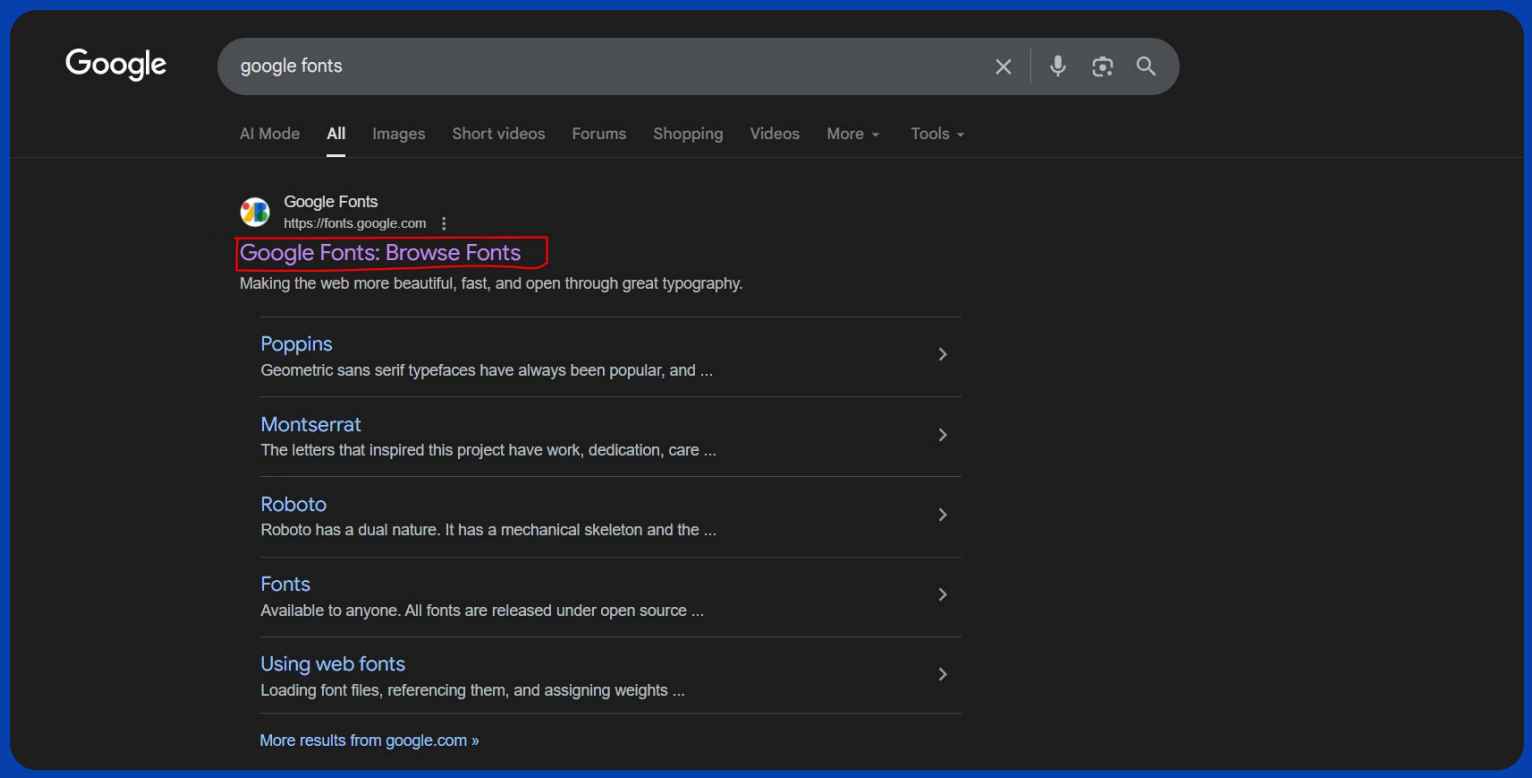
- In the font-stack shown in above example, the browser will first try to use **Calson** font, if that font is not available then it will try to use the next font in given font-stack i.e **Georgia**, and if that font is also not available then it will try to use the next font in the given font-stack i.e **Times New Roman.**
- Thus in above case **Georgia** and **Times New Roman** are the **fallback fonts** for the **Calson** font which is ahead in terms of priority in the above font-stack.
- If all the fonts inside the font-stack is not available then in order to solve this problem, it is recommended to add **Serif** and **Sans Serif** keywords at the end of your font-stack.
- Eg:

```
h1{
   font-family: Calson, Georgia, 'Times New Roman', Serif, 'Sans Serif';
 }
```

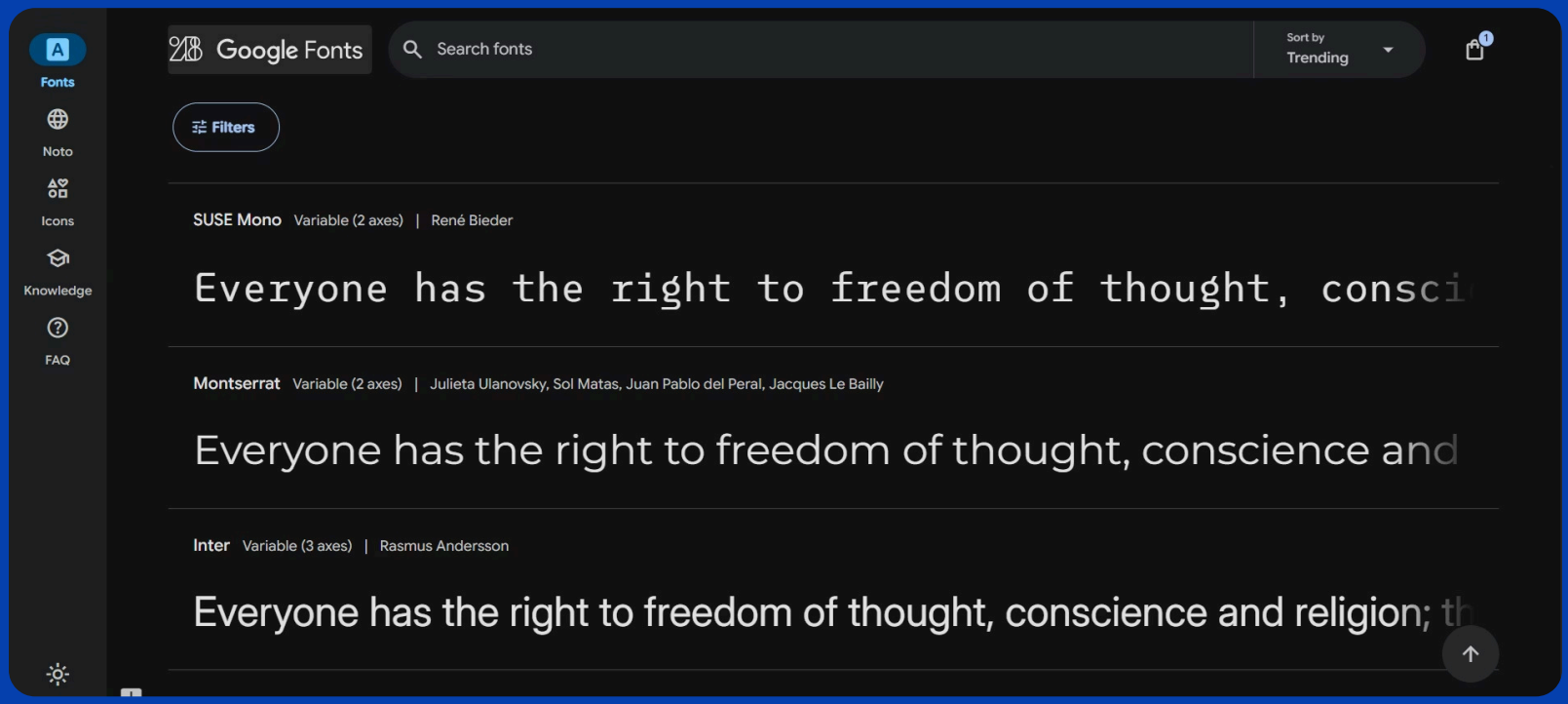
- **Serif** and **Sans Serif** are the font-families which appear same across all browsers and operating systems. These fonts are referred as **Web Safe fonts.**

How to link Google Fonts with our website

- Open any Browser of your choice and type **"Google Fonts"** in the search bar and click on first link.



- Select the font of your choice from the font library getting displayed on the screen.



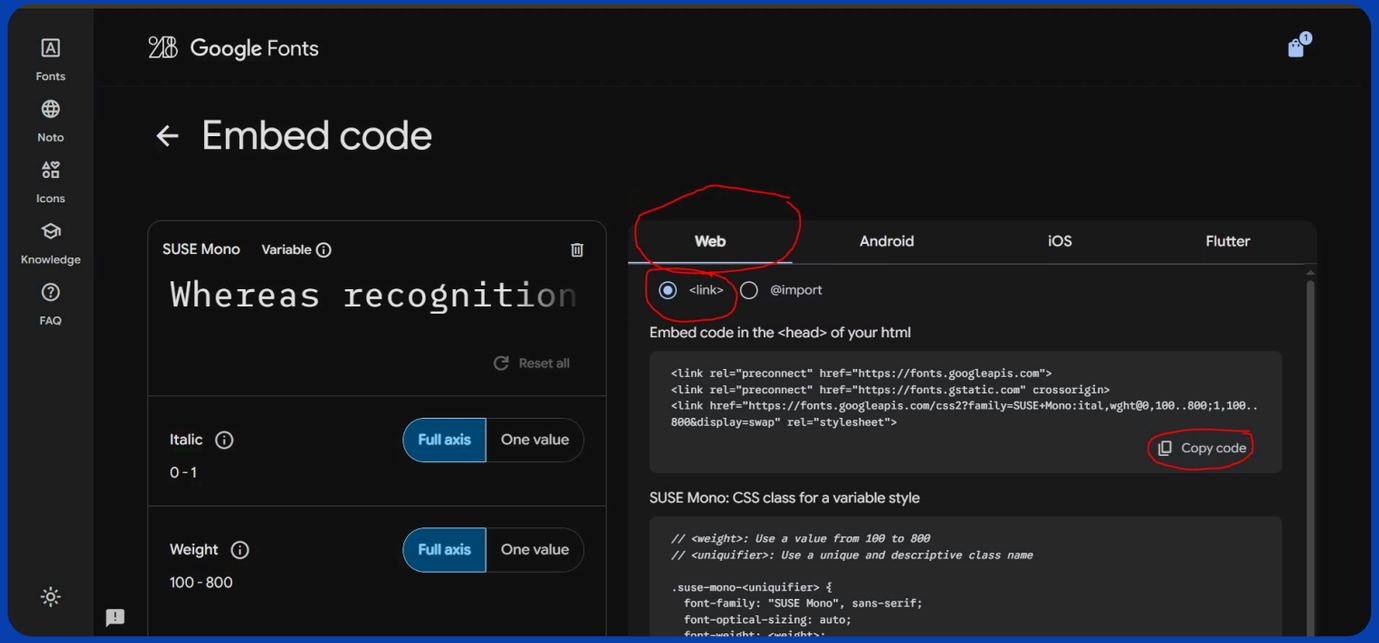
How to link Google Fonts with our website

- Click on **"Get Font"**

- Click on **"Get Embed Code"**

How to link Google Fonts with our website

- Under **Web** tab > click on the radio button **<link>** and then click on **copy code** button.

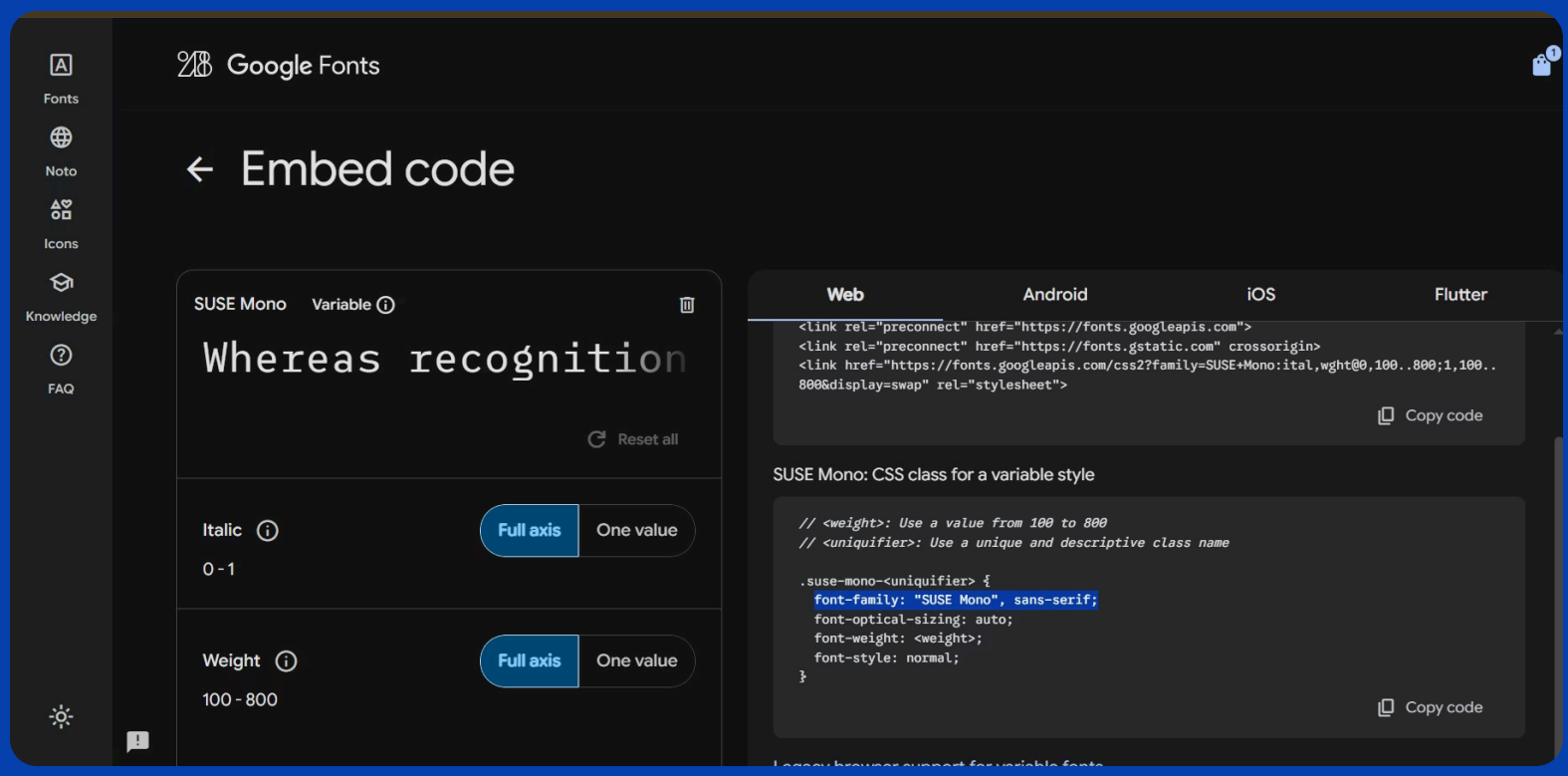


- Come to **VS Code** and paste it inside the **<head>** element of your **HTML boilerplate** code.

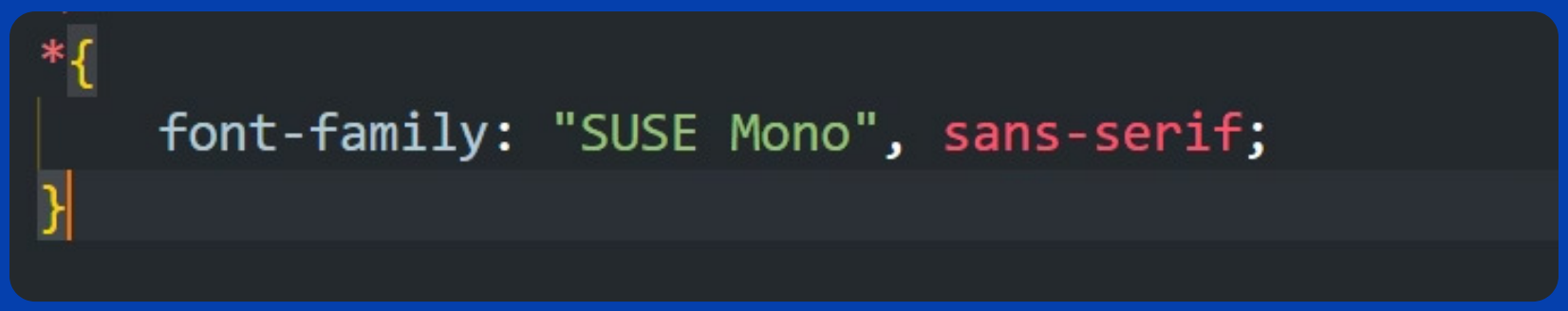
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Google Fonts demo</title>
    <!-- ^ Paste the copied code like this below -->
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
    <link
      href="https://fonts.googleapis.com/css2?family=SUSE+Mono:ital,wght@0,100..800;1,100..800&display=swap"
      rel="stylesheet"
    />
  </head>
  <body></body>
</html>
```

How to link Google Fonts with our website

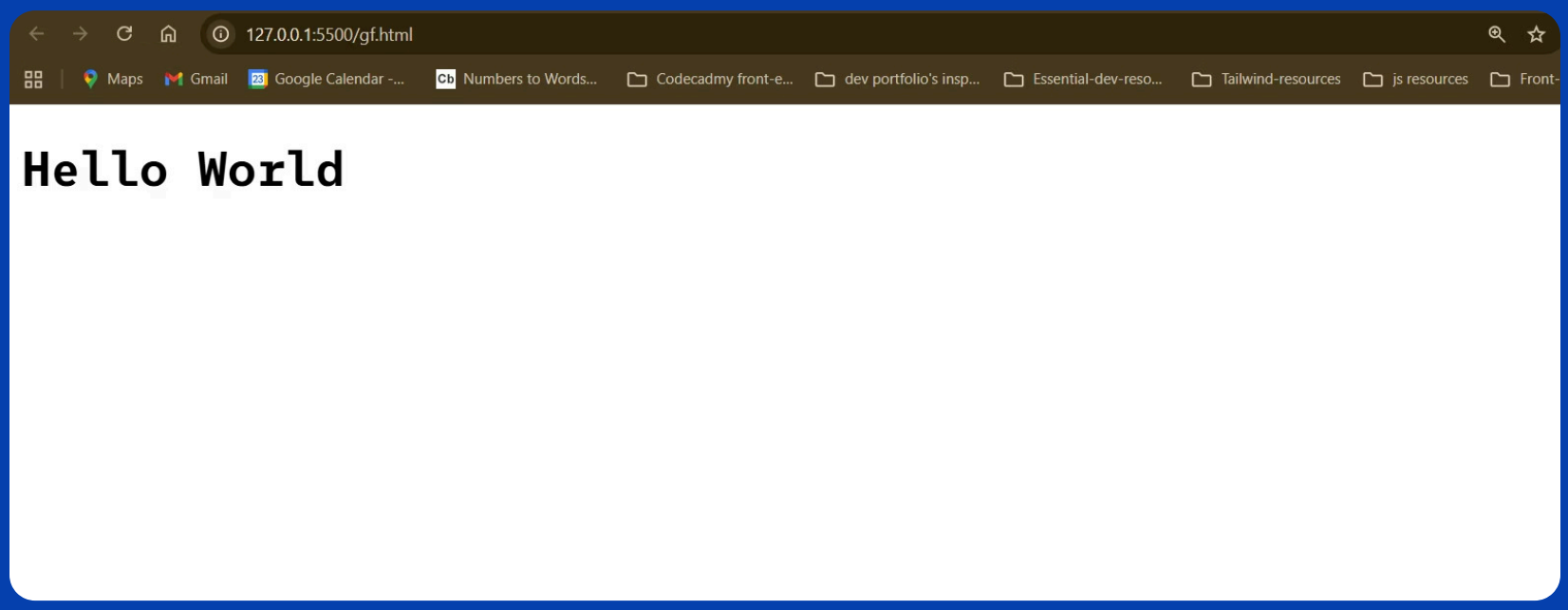
- From here copy the first line of code (selected one) as shown in below screenshot.



- Paste this line **using universal selector** into your CSS file linked to your HTML file or inside **<style>** element (in case of internal CSS)



- Look at the output:



Font Properties

2. **font-size:**

- To change the size of text on your web page, you can use the **font-size** property.
- Eg:

```
p{
   font-size: 1.2rem;
 }
```

- Applying above eg, the **font-size** of the **<p>** element will be set to **1.2rem.**

3. **font-weight:**

- The **font-weight** property controls how **bold** or **thin** the text appears.
- Eg:

```
p{
   font-weight: bold;
 }
```

- Applying above example, all paragraphs on webpage would appear bolded.
- **font-weight** takes the following values:

1. **normal (400)**
2. **lighter**
3. **bold (700)**
4. **bolder**

- It can also take numeric values **1 (lightest)** to **1000 (boldest).**

Font Properties

4. **font-style:**

- The css **font-style** property specifies the font style for a text.
- This property can have one of the following values:

1. **normal** - The text is shown normally.
2. **italic -** The text is shown in italics.
3. **oblique -** The text is "leaning" (oblique is very similar to italic).

- Eg:

```
p{
  font-style: italic;
 }
```

- The above example will make all the paragraphs look italic.

CSS Colors

- Colors in CSS can be described in three different ways:

1. **Named colors**
2. **Hexadecimal colors**
3. **RGB**
4. **HSL**

- Let's study each color in detail

CSS Colors

1. Named Colors:

- These are English words that describe colors, also called as **keyword colors.**
- **Eg:** Red, Blue, Magenta, Hotpink, Biege, etc.

2. Hexadecimal Colors:

- A hex color begins with a **hash character (#)**, which is followed by three or six characters.
- The Characters represent values for **Red, Blue** and **Green.**
- **Eg:** black - #000000 or #000, white- #ffffff or #fff, aqua: #00ffff or #0ff.
- In the example above, you may notice that there are both letters and numbers in the values. This is because the hexadecimal number system has 16 digits (0-15), to represent 10-15, we use A-F.

3. RGB Colors:

- RGB uses decimal numbers and it looks like the below example:
- **Eg:**

```
h1{
    color: rgb(23,45,23);
 }
```

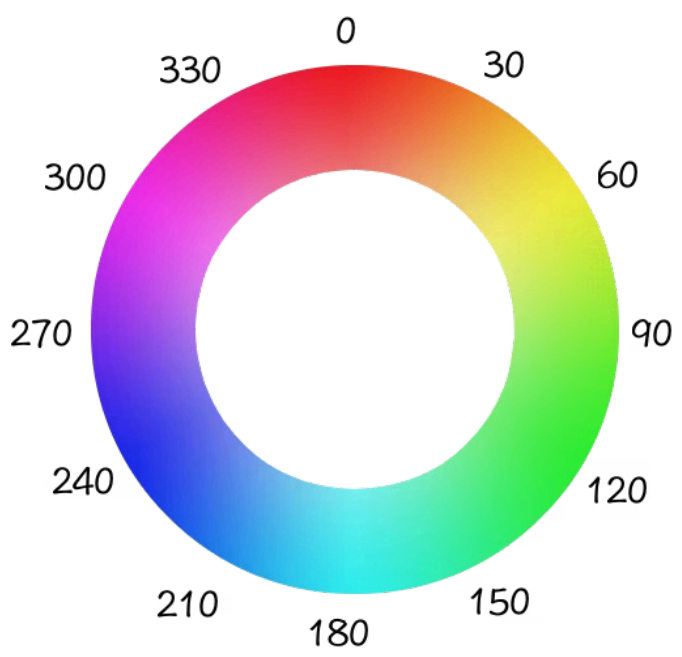
- Each of three values represent a color component and each can have decimal number value from **0 to 255**. The first number represents the amount of **Red**, second represent **green** and third represents **blue**.

4. HSL colors:

- HSL stands for **Hue, Saturation** and **Lightness.**
- The syntax for HSL is similar to the decimal form of RGB, though it differs in important ways. Here's an example:
- **Eg:**

```
h1{
   color: hsl(120, 60%, 70%);
 }
```

- **Hue** is the first number, **it refers to an angle on a color wheel**.
- **Red is 0 degrees and 360 degrees.**
- **Green is 120 degrees.**
- **Blue is 240 degrees.**
- **Saturation** refers to the **intensity or purity of color.**
- The saturation **increases towards 100% as color becomes richer.** The saturation **decreases towards 0% as color becomes grayer.**
- **Lightness** refers to **how light or dark the color is.**
- Halfway or 50% is **normal lightness.**



- Sliding the dimmer up towards 100%, **makes the color lighter, closer to white.**
- Sliding the dimmer down towards 0%, **makes the color darker, closer to black.**

Opacity and Alpha

- Opacity is nothing but **amount of transparency.**
- To use opacity in HSL color scheme, use **hsla** instead of **hsl,** and four values instead of three
- **Eg:**

```
h1{
    color: hsla(34, 100%, 50%, 0.1);
 }
```

- **Alpha (a)** is a decimal number from **0-1**.
- If **a = 0, the color will be completely transparent.**
- If **a = 0.5, the color will be half-transparent.**

Useful Resources:

- https://fonts.google.com/
- https://fonts.adobe.com/
- https://fontsinuse.com/
- https://coolors.co/
- https://cssgradient.io/
- http://colormind.io/bootstrap/
- https://tailwindcss.com/docs/colors