

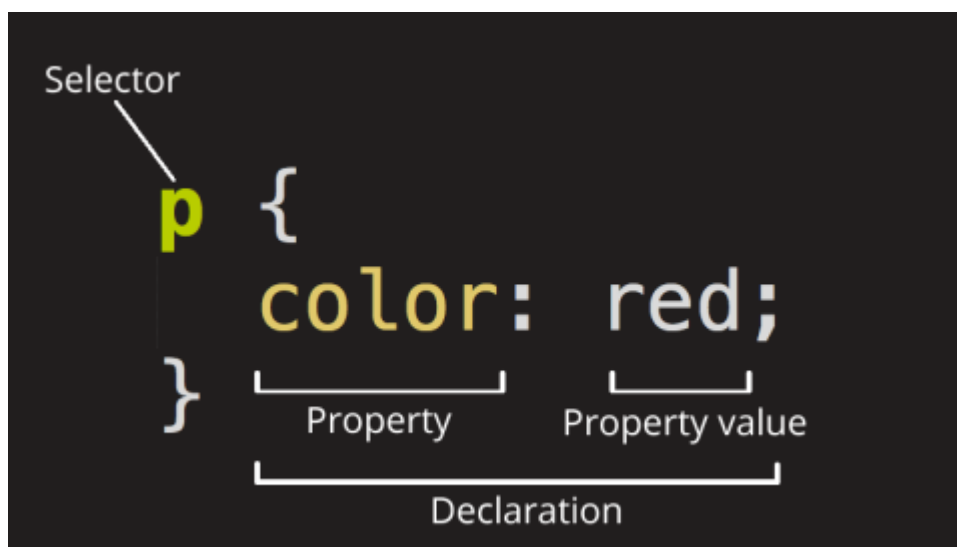
# CSS (Cascading Style Sheets)

## 1. What is CSS?

Cascading Style Sheets (CSS) is a fundamental technology used in web development to control the formatting and layout of HTML documents. While HTML provides the structure and content of a web page, CSS allows developers to apply styles, such as colors, fonts, spacing, and positioning, to create visually appealing and user-friendly interfaces.

## 2. Selectors and declarations

CSS is built around selectors and declarations.



**Selectors** specify which HTML elements the styles should target. They can be as simple as selecting an element by its name (e.g., `p` for all paragraph elements) or as targeting elements based on classes, IDs, attributes.

**Declarations** define the style properties and their values that should be applied to the selected elements. A declaration consists of a property (e.g., `color`) and a value (e.g., `blue`), separated by a colon and enclosed in curly braces. For example, `p { color: blue; }` changes the text color of all paragraphs to blue.

Let's see it in a very simple example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Blue Paragraph</title>
  <style>
    p {
      color: blue;
    }
  </style>
```

```
</head>
<body>
  <p>This is a blue paragraph.</p>
  <p>This is also a blue paragraph.</p>
</body>
</html>
```

### 3. Defining CSS styles

CSS offers three main methods for applying styles to HTML documents: **inline**, **embedded**, and **external CSS**. Each method has its own use cases, benefits, and limitations.

Choosing the right method depends on the project requirements. Inline CSS is useful for quick fixes or unique styling, embedded CSS works well for single-page designs, and external CSS is best for larger projects with multiple pages, as it promotes code reusability and maintainability.

#### Inline

Inline CSS applies styles directly to individual HTML elements using their `style` attribute. It is defined within the opening tag of an element, such as a `<p>` or `<h1>` tag. For example:

```
<p style="color: red; font-size: 16px;">This is a uniquely styled paragraph
with red color and a font size of 16 pixels.</p>
<p>This paragraph has the default formatting, no CSS is applied to it.</p>
```

Inline CSS is useful for making quick style changes to a specific element without affecting other parts of the page. However, it is not ideal often, as these styles are not easily reusable in other parts of the page.

#### Embedded

Embedded CSS, also known as internal CSS, is used within the `<style>` tag in the `<head>` section of an HTML document. This method allows you to apply styles to multiple elements on the page without creating a separate stylesheet file. For example:

```
<head>
  <style>
    h1 {
      color: blue;
    }

    .bold {
      font-weight: bold;
    }
  </style>
```

```
</head>
<body>
  <h1>Blue Heading</h1>
  <p>This is an <span class="bold">example</span> paragraph. </p>
</body>
```

Embedded CSS is beneficial for styling a single page because it keeps all the CSS rules in one place within the document. However, it is less efficient when working with multiple pages, as styles have to be repeated in each document.

## External

External CSS is defined in a separate .css file and linked to the HTML document using the <link> element. This file contains all the CSS rules for the website and can be referenced by multiple HTML files. For example, an HTML document can link to an external stylesheet as follows:

```
<head>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
```

An external stylesheet might include rules like:

```
p {
  color: blue;
  line-height: 1.5;
}
```

External CSS is ideal for websites because it allows for a consistent style across multiple pages, simplifies maintenance, and improves load times by enabling browsers to cache the stylesheet.

## 4. CSS comments

Comments in CSS are used to add notes, explanations, or temporarily disable code without affecting the styles applied to a webpage. They are not displayed in the browser and do not impact the page's layout or design.

```
/* Formatting rules for paragraphs */
p {
  color: black; /* Set the text color to black */
  font-size: 16pt; /* Set font size to 16 pts */
}
```

## 5. Selecting elements, classes and IDs

CSS selectors are patterns used to select and style HTML elements. Three common types of selectors are **element**, **class**, and **ID** selectors, each serving a unique purpose and offering different levels of specificity.

An **element selector** targets all instances of a specific HTML element on a webpage. It is defined by simply using the element's name, such as `p`, `h1`, or `div`. For example, the selector `p` will apply styles to every `<p>` tag in the document. In this example, all `<p>` elements on the page will be displayed with a font size of 16px and black text color:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p {
      font-size: 16px;
      color: black;
    }
  </style>
</head>
<body>
  <p>This is a paragraph styled using an element selector.</p>
  <p>Another paragraph with the same styles.</p>
</body>
</html>
```

A **class selector** targets one or more elements that have the same `class` attribute. Class selectors are defined using a period ( `.` ) followed by the class name. In this example, both the `<p>` and `div` elements with the class `highlight` will have a yellow background and bold font weight. Since classes can be reused across multiple elements, class selectors offer greater flexibility in styling.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .highlight {
      background-color: yellow;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <p class="highlight">This paragraph is styled using a class
  selector.</p>
  <p>This paragraph is not highlighted.</p>
  <div class="highlight">This div is also styled using the same class
  selector.</div>
</body>
</html>
```

An **ID selector** targets a single, unique HTML element by referencing its `id` attribute. ID selectors are denoted using the `#` symbol followed by the ID name. Here, only the `<h1>` element with `id="main-header"` is styled, while other elements remain unaffected. This makes ID selectors ideal for targeting only specific elements.

```

<!DOCTYPE html>
<html>
<head>
  <style>
    #main-header {
      font-size: 24px;
      text-align: center;
    }
  </style>
</head>
<body>
  <h1 id="main-header">This is a header styled using an ID selector.</h1>
  <p>This paragraph is not affected by the ID selector.</p>
</body>
</html>

```

In summary, element selectors are used for broad styling, ID selectors offer targeted styling for unique elements, and class selectors provide versatility by enabling the styling of multiple elements at once. Understanding these selectors and how to use them together is key to writing efficient and maintainable CSS.

Element selectors are marked with the name of the member, class selectors are marked with a `.` and the name of the class, and ID selectors are marked with a `#` and the identifier:

Selector type	HTML example	CSS selector
Element selector	<code>&lt;p&gt;Test&lt;/p&gt;</code>	<code>p</code>
Class selector	<code>&lt;p class="highlight"&gt;Test&lt;/p&gt;</code>	<code>.highlight</code>
ID selector	<code>&lt;p id="main-text"&gt;Test&lt;/p&gt;</code>	<code>#main-text</code>

## 6. Cascading

By default, when multiple rules target the same selector, the later rule will override the previous one:

```

<head>
  <style>
    p {
      color: red;
    }

    p {
      color: green;
    }
  </style>
</head>
<body>
  <p>This paragraph will be green.</p>
</body>

```

## 7. Role of span and div elements

The **span** and **div** elements are fundamental components in HTML used for grouping and organizing content. They serve as containers that can be styled using CSS.

The span element is an **inline container**, meaning it only takes up as much width as its content. It is typically used to style or manipulate small portions of text. In this example, the span is used to color only the word “blue” without affecting the rest of the text. It is perfect for applying specific styles or behaviors to parts of a sentence, such as changing text color or font weight:

```
<p>This is a <span style="color: blue;">blue</span> word in a paragraph.</p>
```

The div element is a **block-level container**, meaning it takes up the full width of its parent element by default and creates a line break before and after it. It is commonly used to group larger sections of content, such as paragraphs, images, or other block-level elements. For example:

```
<div style="background-color: lightgrey;">
  <h1>Section Title</h1>
  <p>This paragraph is inside a div container.</p>
</div>
```

## 8. Background styling

Background styling in CSS allows developers to customize the appearance of an element's background, enhancing the overall visual appeal of a webpage. There are several properties associated with background styling that can be used to control color, images, gradients, and positioning. For example:

```
body {
  background-color: #f0f0f0;
}

#header {
  background-image: url('header.jpg');
}
```

## 9. Margin, padding and border

The margin is the outer space around an element, while the padding is the inner space between the content and the border. The border defines the outline of the element. This is called the “box model.”



```
div {  
  margin: 10px;  
  padding: 20px;  
  border: 1px solid black;  
  border-radius: 5px;  
}
```

## 10. Setting width and height

The width and height properties allow you to set the dimensions of an element.

```
img {  
  width: 100px;  
  height: 100px;  
}
```

## 11. Text styling

There are several properties available for text formatting.

```
p {  
  font-family: Arial;  
  font-size: 16px;  
  font-weight: bold;  
  font-style: italic;  
  
  text-decoration: underline;  
  text-align: justify;  
}
```

## 12. Exercise

Create the following webpage by using HTML and CSS. Use inline CSS for formatting the text, and also apply class-based and id-based selectors!

Feel free to use online resources (e.g. [W3Schools CSS tutorial](https://www.w3schools.com/css/)) as help.

# John von Neumann

*"Young man, in mathematics you don't understand things. You just get used to them."  
John von Neumann*

**John von Neumann** (born **János Lajos Neumann**) (Budapest, December 28, 1903 - Washington, February 8, 1957) was a Hungarian-born mathematician. In addition to his theoretical research in quantum mechanics, he became famous for laying the foundations of the digital computer.

John von Neumann



**Born:** December 28, 1903

**Died:** February 8, 1957 (aged 53)

**Known for:** Mathematician, computer scientist, physicist, professor

**Citizenship:** Hungarian, American

**Education:**

- Budapest-Fasori Evangelical Gymnasium
- Friedrich Wilhelm University
- Royal Hungarian Pázmány Péter University
- Swiss Federal Institute of Technology Zurich
- University of Göttingen

Neumann made tremendous contributions to mathematics, quantum mechanics, quantum theory, game theory, economics, and computer science. Several concepts and principles are associated with his name, such as the *Neumann principle* in quantum mechanics, the *Neumann stability theory* in game theory, and the *Neumann process* in numerical analysis.

John von Neumann was one of the most significant figures in computer science. He was among the first to recognize the potential of digital computers. Neumann developed fundamental concepts and architectures that have become foundational in the design and operation of modern computers. He is known for the "*von Neumann architecture*", which is a computer architecture where programs and data are stored in the same memory.

John von Neumann was an incredibly versatile scientist, and his research had a profound impact on computer science, mathematics, and other scientific fields. His work and legacy continue to influence modern scientific and technological developments.

Raw text:

John von Neumann

“Young man, in mathematics you don't understand things. You just get used to them.”

John von Neumann

John von Neumann (born János Lajos Neumann) (Budapest, December 28, 1903 - Washington, February 8, 1957) was a Hungarian-born mathematician. In addition to his theoretical research in quantum mechanics, he became famous for laying the foundations of the digital computer.

John von Neumann

Image:

<https://upload.wikimedia.org/wikipedia/commons/thumb/d/d6/JohnvonNeumann-LosAlamos.jpg/462px-JohnvonNeumann-LosAlamos.jpg>

Born: December 28, 1903

Died: February 8, 1957 (aged 53)

Known for: Mathematician, computer scientist, physicist, professor

Citizenship: Hungarian, American

Education:

- Budapest-Fasori Evangelical Gymnasium
- Friedrich Wilhelm University
- Royal Hungarian Pázmány Péter University
- Swiss Federal Institute of Technology Zurich
- University of Göttingen

Neumann made tremendous contributions to mathematics, quantum mechanics, quantum theory, game theory, economics, and computer science. Several concepts and principles are associated with his name, such as the Neumann principle in quantum mechanics, the Neumann stability theory in game theory, and the Neumann process in numerical analysis.

John von Neumann was one of the most significant figures in computer science. He was among the first to recognize the potential of digital computers. Neumann developed fundamental concepts and architectures that have become foundational in the design and operation of modern computers. He is known for the "von Neumann architecture", which is a computer architecture where programs and data are stored in the same memory.

John von Neumann was an incredibly versatile scientist, and his research had a profound impact on computer science, mathematics, and other scientific fields. His work and legacy continue to influence modern scientific and technological developments.

From:

<https://edu.iit.uni-miskolc.hu/> - Institute of Information Science - University of Miskolc

Permanent link:

[https://edu.iit.uni-miskolc.hu/tanszek:oktatas:techcomm:cascading\\_style\\_sheets?rev=1727775038](https://edu.iit.uni-miskolc.hu/tanszek:oktatas:techcomm:cascading_style_sheets?rev=1727775038)

Last update: **2024/10/01 09:30**

