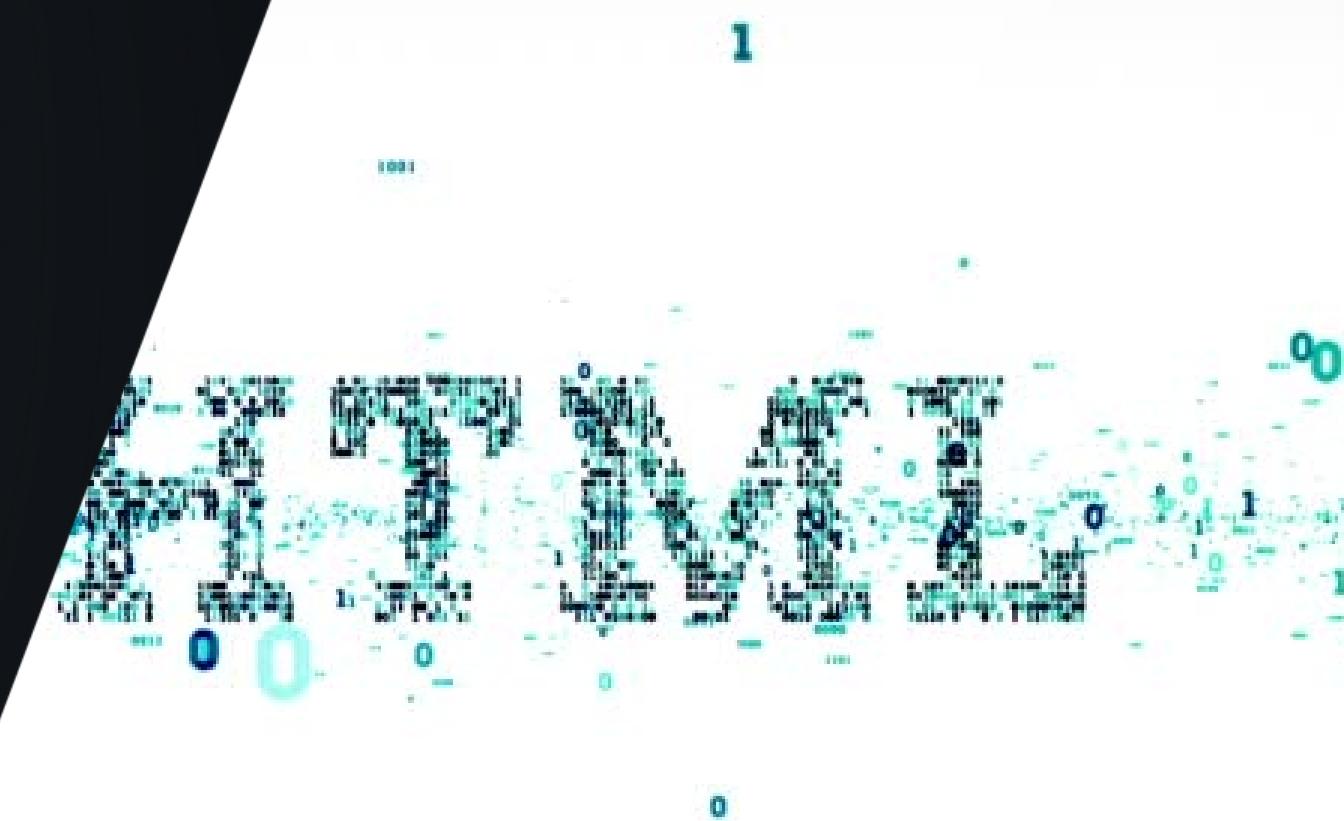


HTML



HTML Introduction

- HTML stands for Hypertext Markup Language.
- Used to design web pages using a markup language.
- Combination of Hypertext and Markup language.



HTML

Basic Web Page

● ● ● p1.html

```
1 <html>
2 <head>
3   <title>Page Title</title>
4 </head>
5 <body>
6   <h1>This is a Heading</h1>
7   <p>This is a paragraph.</p>
8 </body>
9 </html>
```

HTML

Text Formatting

● ● ● p2.html

```
1 <p>This is <b>bold text</b>.</p>
2 <p>This is <strong>strongly important text</strong>.</p>
3 <p>This is <i>italic text</i>.</p>
4 <p>This is <em>emphasized text</em>.</p>
5 <p>This is <mark>highlighted text</mark>.</p>
6 <p>This is <code>computer code</code>.</p>
7 <p>This is <small>smaller text</small>.</p>
8 <p>This is <sub>subscript</sub> and <sup>superscript</sup> text.</p>
9 <p>This is <del>deleted text</del>.</p>
10 <p>This is <ins>inserted text</ins>.</p>
```

HTML

Headings

● ● ● p3.html

```
1 <h1>Heading 1</h1>
2 <h2>Heading 2</h2>
3 <h3>Heading 3</h3>
4 <h4>Heading 4</h4>
5 <h5>Heading 5</h5>
6 <h6>Heading 6</h6>
```

HTML

Tables

● ● ● p4.html

```
1 <table>
2   <tr>
3     <th>Company</th>
4     <th>Contact</th>
5     <th>Country</th>
6   </tr>
7   <tr>
8     <td>Alfreds Futterkiste</td>
9     <td>Maria Anders</td>
10    <td>Germany</td>
11  </tr>
12  <tr>
13    <td>Centro comercial Moctezuma</td>
14    <td>Francisco Chang</td>
15    <td>Mexico</td>
16  </tr>
17 </table>
```

HTML

Unordered Ordered List

● ● ● p5.html

```
1 <ul>
2   <li>Coffee</li>
3   <li>Tea</li>
4   <li>Milk</li>
5 </ul>
6
7 <ol>
8   <li>Coffee</li>
9   <li>Tea</li>
10  <li>Milk</li>
11 </ol>
```

HTML

Form Elements

p6.html

```
1  <form>
2      <label>First name:</label><br>
3      <input type="text" id="fname" name="fname" value="John"><br>
4
5      <label>Last name:</label><br>
6      <input type="text" id="lname" name="lname" value="Doe"><br><br>
7
8      <label>Choose a car:</label>
9      <select id="cars" name="cars">
10         <option value="volvo">Volvo</option>
11         <option value="audi">Audi</option>
12     </select> <br>
13
14     <textarea name="message" rows="10" cols="30">
15         The cat was playing in the garden.
16     </textarea><br>
17
18     <input type="radio" id="html" name="fav_language" value="HTML">
19     <label>HTML</label><br>
20
21     <input type="radio" id="css" name="fav_language" value="CSS">
22     <label>CSS</label><br>
23
24     <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
25     <label> I have a bike</label><br>
26
27     <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
28     <label> I have a car</label><br>
29
30
31     <input type="submit" value="Submit">
32 </form>
```

HTML

Images

● ● ● p7.html

```
1 
2 
```

HTML

Video

● ● ● p8.html

```
1 <video width="400" controls>
2   <source src="mov_bbb.mp4" type="video/mp4">
3   <source src="mov_bbb.ogg" type="video/ogg">
4   Your browser does not support HTML video.
5 </video>
```

HTML

Audio

● ● ● p9.html

```
1 <audio controls>
2   <source src="horse.ogg" type="audio/ogg">
3   <source src="horse.mp3" type="audio/mpeg">
4 </audio>
```

HTML

YouTube Videos

● ● ● p10.html

```
1 <iframe  
2      width="420"  
3      height="345"  
4      src="https://www.youtube.com/embed/tgbNymZ7vqY">  
5 </iframe>
```

HTML

Meta Tags

● ● ● p11.html

```
1 <meta charset="utf-8" />
2 <link rel="icon" href="/logo192.png" />
3 <meta name="viewport" content="width=device-width,initial-scale=1" />
4 <meta name="description" content="">
5 <meta name="keywords" content="">
6 <meta name="author" content="Rabbil Hasan">
7 <meta property="og:site_name" content="Learn With Rabbil">
8 <meta property="og:url" content="https://rabbil.com" />
9 <meta property="og:title" content="Learn Confidently with Rabbil Hasan" />
10 <meta property="og:description" content="" />
11 <meta property="og:image" content="/indexseo.png" />
12 <link rel="manifest" href="/manifest.json" />
13 <title>Learn Confidently</title>
```

HTML

Meta Tags For No Cache

● ● ● p12.html

```
1 <meta http-equiv="cache-control" content="max-age=0" />
2 <meta http-equiv="cache-control" content="no-cache" />
3 <meta http-equiv="expires" content="0" />
4 <meta http-equiv="expires" content="Tue, 01 Jan 1980 1:00:00 GMT" />
5 <meta http-equiv="pragma" content="no-cache" />
6
```

HTML

Time, Mark

● ● ● p13.html

```
1
2 <p>Written by Doctor Who on <time datetime="2052-11-21">Thursday 21st November 2052</time>. </p>
3 <mark>RABBIL HASAN</mark>
4
```

HTML

Datalist Autocomplete
suggestion

● ● ● p14.html

```
1
2 <input name="country" list="country_name">
3 <datalist id="country_name">
4   <option value="Afghanistan">
5   <option value="Albania">
6   <option value="Algeria">
7   <option value="Andorra">
8   <option value="Armenia">
9   <option value="Australia">
10  <option value="Austria">
11  <option value="Azerbaijan">
12  <!-- etc. -->
13 </datalist>
```

● ● ● p15.html

HTML

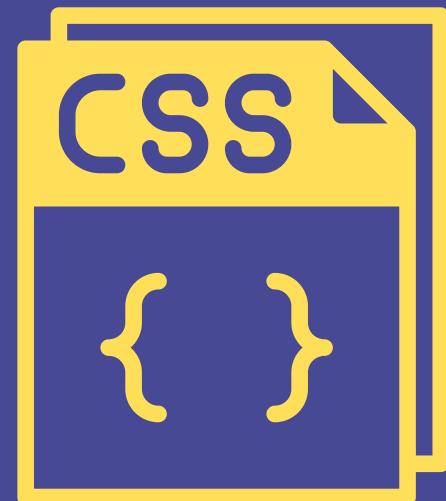
Emojis

```
1 <p style="font-size:48px">
2   &#128507;
3   &#128508;
4   &#128509;
5   &#128510;
6   &#128511;
7   &#128512;
8   &#128513;
9   &#128514;
10  &#128515;
11  &#128516;
12  &#128517;
13 </p>
```

CSS Fundamental



CSS is the language we use to style an HTML document



What is CSS

-  CSS STANDS FOR CASCADING STYLE SHEET.
-  CSS IS USED TO DESIGN HTML TAGS
-  CSS IS A WIDELY USED LANGUAGE ON THE WEB

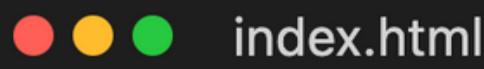
How to add CSS

CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

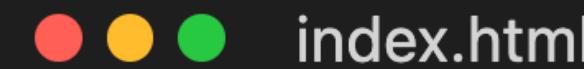
1. Inline CSS
2. Internal CSS
3. External CSS



```
14 <p style="color:blue">Hello CSS</p>
```



```
4 <style>
5   p{color:blue}
6 </style>
```



```
8 <link rel="stylesheet" type="text/css" href="style.css">
```

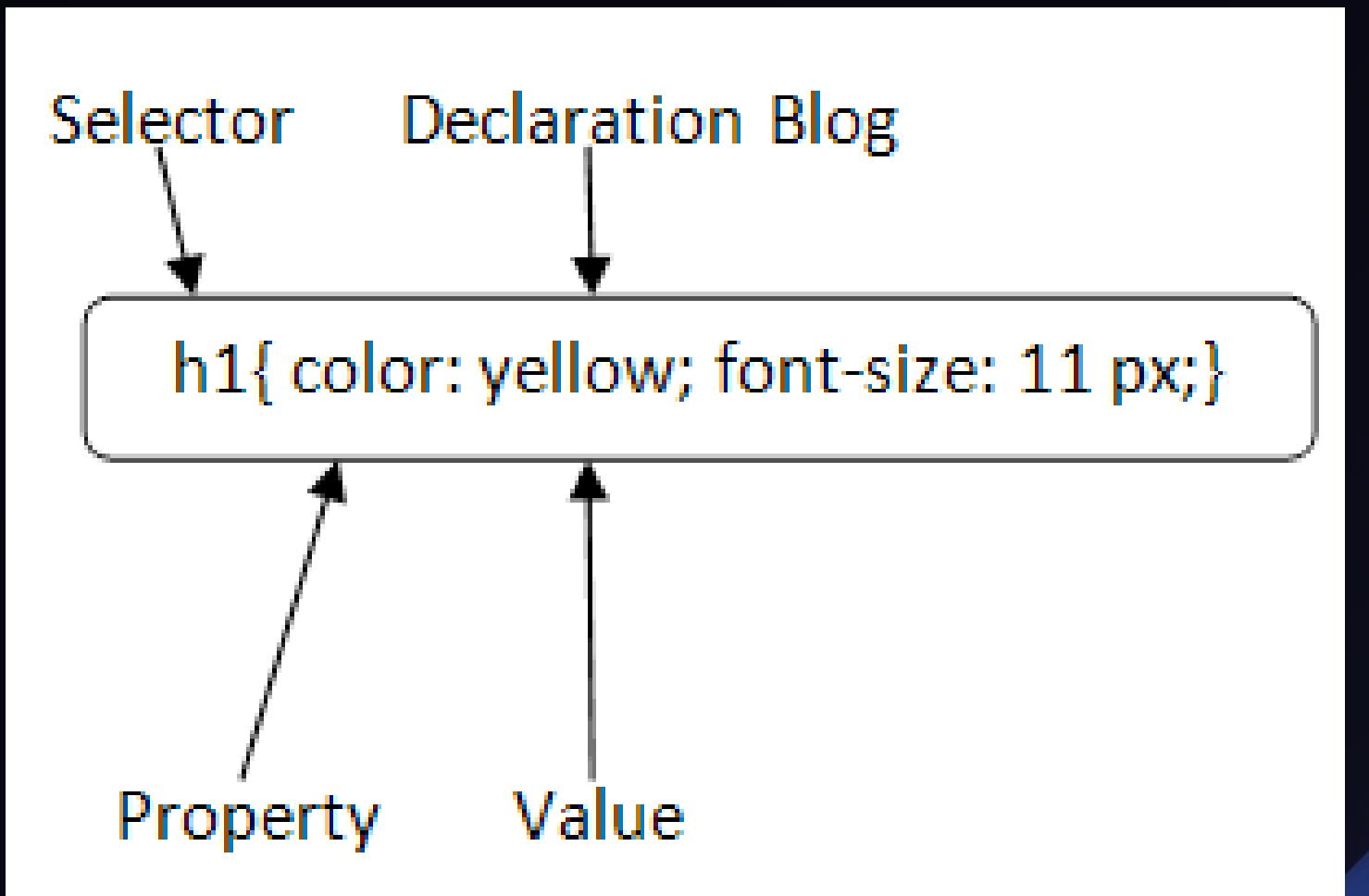
CSS Syntax

Selector: Selector indicates the HTML element you want to style. It could be any tag like `<h1>`, `<title>` etc.

Declaration Block: The declaration block can contain one or more declarations separated by a semicolon.

Property: A Property is a type of attribute of HTML element.

Value: Values are assigned to CSS properties.



CSS Selector



CSS ELEMENT SELECTOR



CSS ID SELECTOR



CSS CLASS SELECTOR



CSS UNIVERSAL SELECTOR



CSS GROUP SELECTOR

CSS SELECTOR

CSS ELEMENT SELECTOR

```
● ● ● index.html  
  
11 p{  
12   text-align: center;  
13   color: blue;  
14 }
```

CSS CLASS SELECTOR

```
● ● ● index.html  
  
16 .center {  
17   text-align: center;  
18   color: blue;  
19 }
```

CSS ID SELECTOR

```
● ● ● index.html  
  
6 #para1 {  
7   text-align: center;  
8   color: blue;  
9 }  
10
```

CSS UNIVERSAL SELECTOR

```
● ● ● index.html  
  
21 * {  
22   color: green;  
23   font-size: 20px;  
24 }
```

CSS GROUP SELECTOR

```
● ● ● index.html  
  
26 h1, h2, p {  
27   text-align: center;  
28   color: blue;  
29 }
```

CSS Background Properties

● ● ● style.css

```
1 #MyBackground{  
2     background-color: #b0d4de;  
3     background-image: url("gradient_bg.png");  
4     background-repeat: repeat-x;  
5     background-attachment: fixed;  
6     background-position: center;  
7 }  
8
```

CSS Border Properties

● ● ● style.css

```
1 p.none {border-style: none;}  
2 p.dotted {border-style: dotted;}  
3 p.dashed {border-style: dashed;}  
4 p.solid {border-style: solid;}  
5 p.double {border-style: double;}  
6 p.groove {border-style: groove;}  
7 p.ridge {border-style: ridge;}  
8 p.inset {border-style: inset;}  
9 p.outset {border-style: outset;}  
10 p.hidden {border-style: hidden;}  
11
```

CSS border-radius Properties

● ● ● style.css

```
1 #one {  
2     border-radius: 90px;  
3     background: lightgreen;  
4 }  
5 #two {  
6     border-radius: 25% 10%;  
7     background: orange;  
8 }  
9 #three {  
10    border-radius: 35px 10em 10%;  
11    background: cyan;  
12 }  
13 #four {  
14    border-radius: 50px 50% 50cm 50em;  
15    background: lightblue;  
16 }
```

CSS Display Properties

● ● ● style.css

```
1  p{  
2      display: inline-block;  
3  }  
4  p{  
5      display:block;  
6  }  
7  p{  
8      display: run-in;  
9  }  
10 p{  
11     display: none;  
12 }
```

CSS Cursor Properties

● ● ● index.html

```
8      <div style = "cursor:alias">alias Value</div>
9      <div style = "cursor:auto">auto Value</div>
10     <div style = "cursor:all-scroll">all-scroll value</div>
11     <div style = "cursor:col-resize">col-resize value</div>
12     <div style = "cursor:crosshair">Crosshair</div>
13     <div style = "cursor:default">Default value</div>
14     <div style = "cursor:copy">copy value</div>
15     <div style = "cursor:pointer">Pointer</div>
16     <div style = "cursor:move">Move</div>
17     <div style = "cursor:e-resize">e-resize</div>
18     <div style = "cursor:ew-resize">ew-resize</div>
19     <div style = "cursor:ne-resize">ne-resize</div>
20     <div style = "cursor:nw-resize">nw-resize</div>
```

Snipped

CSS Cursor Properties

● ● ● index.html

```
21 <div style = "cursor:n-resize">n-resize</div>
22 <div style = "cursor:se-resize">se-resize</div>
23 <div style = "cursor:sw-resize">sw-resize</div>
24 <div style = "cursor:s-resize">s-resize</div>
25 <div style = "cursor:w-resize">w-resize</div>
26 <div style = "cursor:text">text</div>
27 <div style = "cursor:wait">wait</div>
28 <div style = "cursor:help">help</div>
29 <div style = "cursor:progress">Progress</div>
30 <div style = "cursor:no-drop">no-drop</div>
31 <div style = "cursor:not-allowed">not-allowed</div>
32 <div style = "cursor:vertical-text">vertical-text</div>
33 <div style = "cursor:zoom-in">Zoom-in</div>
34 <div style = "cursor:zoom-out">Zoom-out</div>
```

CSS Float Properties

● ● ● style.css

```
1 #img1{  
2     float: right;  
3 }  
4 #img2{  
5     float: left;  
6 }  
7 #img3{  
8     float: initial;  
9 }  
10 #img4{  
11     float: inherit;  
12 }  
13
```

CSS Font Properties

● ● ● style.css

```
1 #MyFont{  
2     font-size: 40px;  
3     font-variant: small-caps;  
4     font-weight:400;  
5     color: red;  
6     font-family: 'Courier New', Courier, monospace;  
7     font-style: italic;  
8 }
```

CSS Hover Properties

● ● ● style.css

```
1 .my-button{  
2     background-color: red;  
3 }  
4 .my-button:hover{  
5     background-color: green;  
6 }  
7
```

CSS Important

● ● ● style.css

```
1 element {  
2     font-size: 14px !important;  
3     color: blue !important;  
4 }
```

CSS Line Height

● ● ● style.css

```
1 element {  
2     line-height: 70%;  
3 }
```

CSS Margin

● ● ● style.css

```
1 element {  
2   margin-top: 50px;  
3   margin-bottom: 50px;  
4   margin-right: 100px;  
5   margin-left: 100px;  
6 }
```

CSS Padding

● ● ● style.css

```
1 element {  
2     padding-top: 50px;  
3     padding-right: 100px;  
4     padding-bottom: 150px;  
5     padding-left: 200px;  
6 }
```

CSS Padding

● ● ● style.css

```
1 element {  
2     padding-top: 50px;  
3     padding-right: 100px;  
4     padding-bottom: 150px;  
5     padding-left: 200px;  
6 }
```

CSS Opacity & Filter

● ● ● style.css

```
1 element {  
2     filter: brightness(130%);  
3     filter: blur(2px);  
4     filter: invert(60%);  
5     filter: saturate(40%);  
6     filter: drop-shadow(10px 20px 30px yellow);  
7     filter: contrast(50%);  
8     filter: opacity(40%);  
9     filter: hue-rotate(240deg);  
10    filter: grayscale(80%);  
11    filter: sepia(90%);  
12 }
```

CSS Overflow

● ● ● style.css

```
1 element {  
2     overflow: scroll;  
3     overflow: hidden;  
4     overflow-y: auto;  
5     overflow-x: scroll;  
6 }
```

CSS Width Height

● ● ● style.css

```
1 element {  
2     height: 150px;  
3     width: 150px;  
4 }
```

CSS Position

● ● ● style.css

```
1 p.pos_fixed {  
2     position: fixed;  
3     top: 50px;  
4     right: 5px;  
5     color: blue;  
6 }  
7 h2.pos_left {  
8     position: relative;  
9     left: -30px;  
10 }  
11 h2.pos_right {  
12     position: relative;  
13     left: 30px;  
14 }
```

● ● ● style.css

```
1 h2 {  
2     position: absolute;  
3     left: 150px;  
4     top: 250px;  
5 }
```

CSS Word Wrap

● ● ● style.css

```
1 p.test {  
2     width: 11em;  
3     background-color: #00ffff;  
4     border: 1px solid #000000;  
5     padding:10px;  
6     word-wrap: break-word;  
7 }
```

CSS justify-content

● ● ● style.css

```
1 #flexstart{  
2     display:flex;  
3     justify-content: flex-start;  
4 }  
5  
6 #flexend{  
7     display:flex;  
8     justify-content: flex-end;  
9 }  
10  
11 #cent{  
12     display:flex;  
13     justify-content: center;  
14 }
```

● ● ● style.css

```
16 #evenly{  
17     display:flex;  
18     justify-content: space-evenly;  
19 }  
20  
21 #around{  
22     display:flex;  
23     justify-content: space-around;  
24 }  
25  
26 #between{  
27     display:flex;  
28     justify-content: space-between;  
29 }
```

CSS text decoration

● ● ● style.css

```
1  #p1 {  
2      text-decoration: underline;  
3  }  
4  #p2 {  
5      text-decoration: line-through;  
6  }  
7  #p3 {  
8      text-decoration: overline;  
9  }  
10 #p4 {  
11     text-decoration: overline underline line-through;  
12 }
```

CSS letter-spacing

● ● ● style.css

```
1 #p1 {  
2     letter-spacing: normal;  
3 }  
4 #p2 {  
5     letter-spacing: 7px;  
6 }  
7 #p3 {  
8     letter-spacing: 0.7em;  
9 }  
10 #p4 {  
11     letter-spacing: -1px;  
12 }
```

CSS word-spacing

● ● ● style.css

```
1 #p1 {  
2     word-spacing: normal;  
3 }  
4 #p2 {  
5     word-spacing: 40px;  
6 }  
7 #p3 {  
8     word-spacing: 20px;  
9 }  
10 #p4 {  
11     word-spacing: 1%;  
12 }
```

CSS Variables

● ● ● style.css

```
1  :root {  
2      --bg-color: lightgreen;  
3      --text-color: red;  
4  }  
5  
6  h1 {  
7      color: var(--text-color);  
8  }
```

TERMINOLOGY

Mandatory Knowledge for
web development

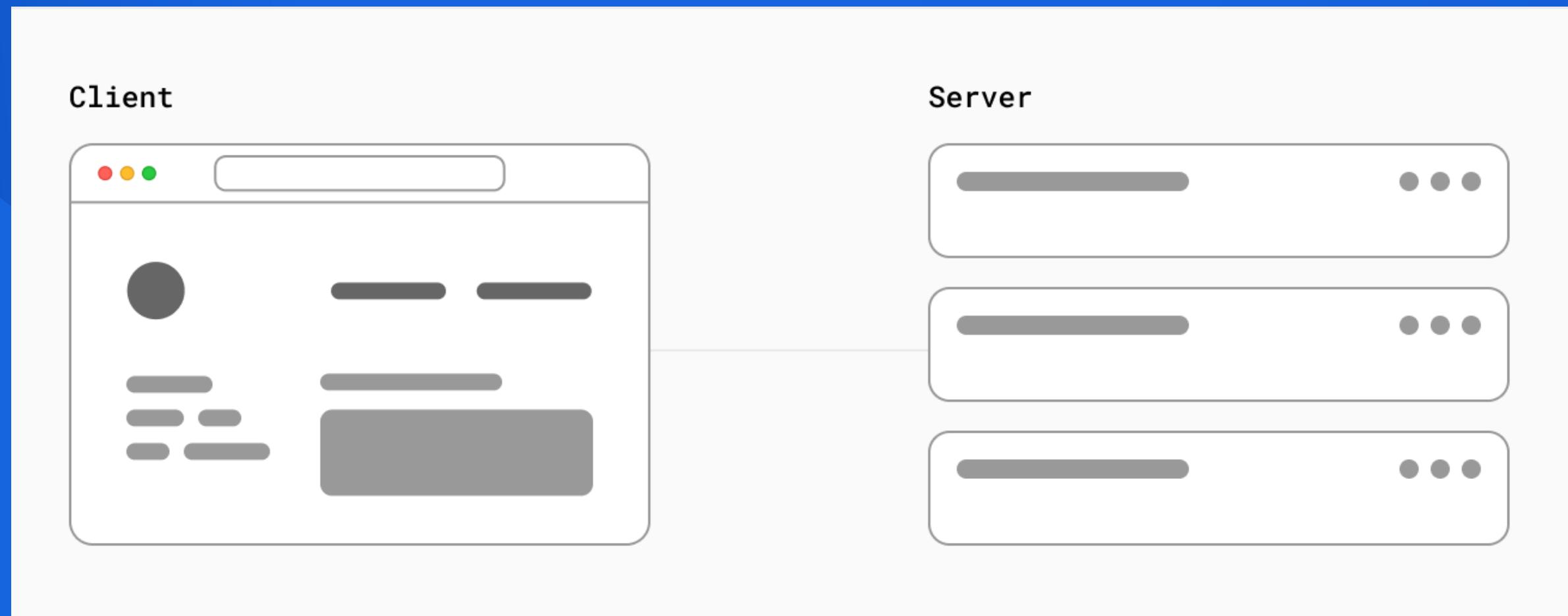


Client-Side

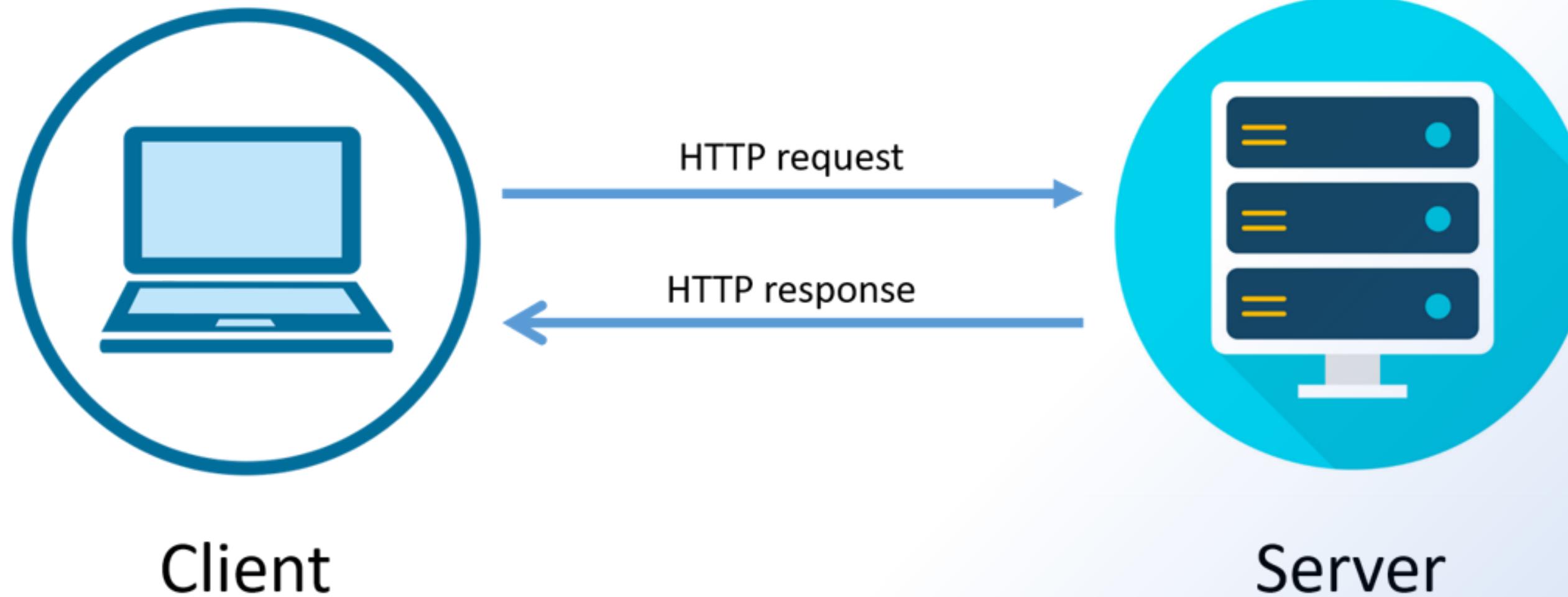
The client refers to the browser on a user's device that sends a request to a server for your application code. It then turns the response it receives from the server into an interface the user can interact with.

Server-Side

Server refers to the computer in a data centre that stores your application code, Receives requests from a client, does some computation, and sends back an appropriate response.

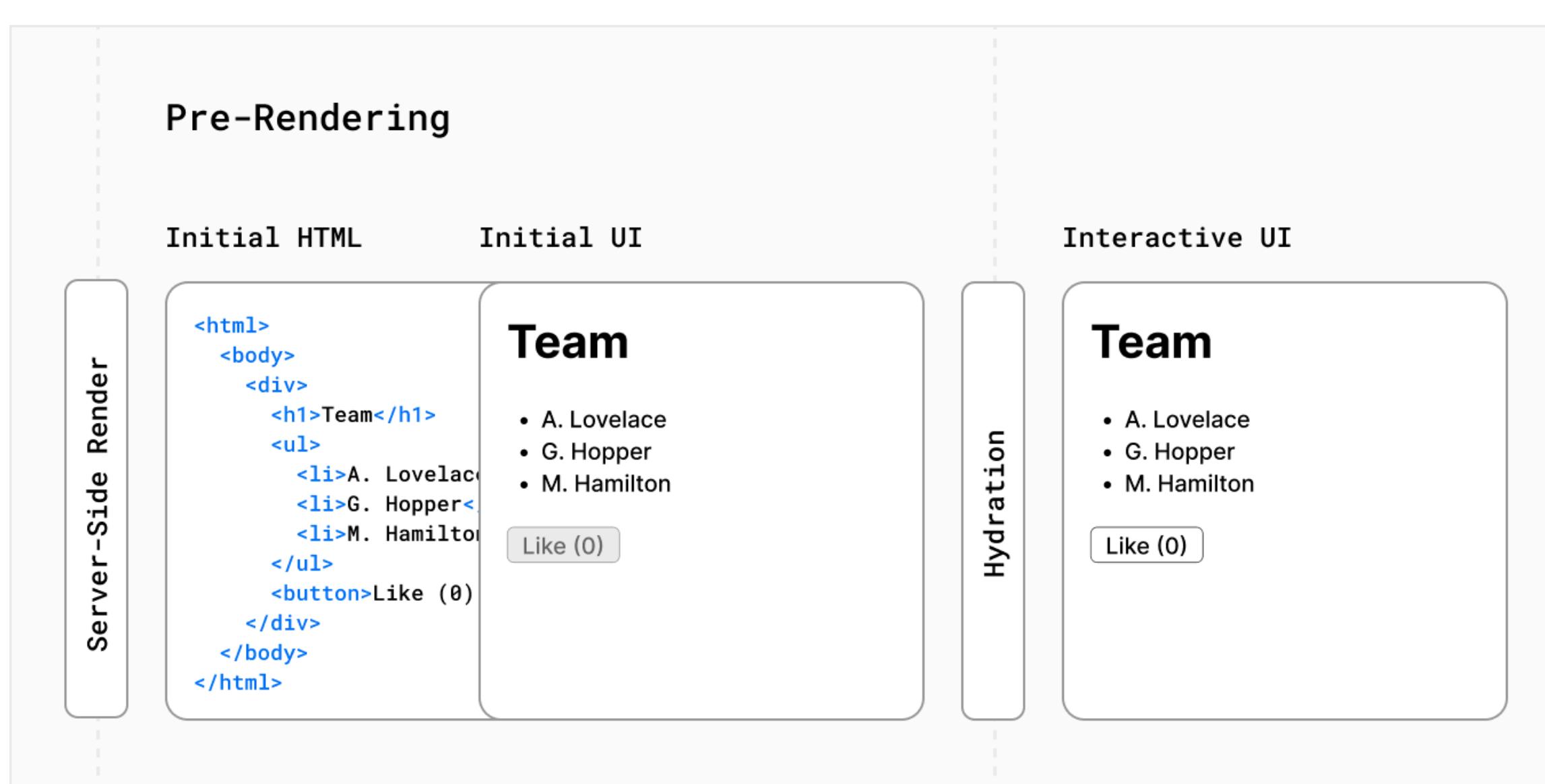


Request-Response Model



Server-Side Rendering

- With server-side rendering, the HTML of the page is generated on a server for each request.
- The generated HTML, JSON data, and JavaScript instructions to make the page interactive are then sent to the client.



BENEFITS OF SSR

Better SEO

Server-side rendering allows search engine bots to crawl and index the content of your website easily, leading to better search engine rankings.

Faster initial loading

Server-side rendering provides faster initial loading of web pages by sending HTML content to the browser before the JavaScript is executed. This helps reduce page load time and improves user experience.

Improved accessibility

Server-side rendering can improve website accessibility by ensuring that content is available to all users, including those with slow or unreliable internet connections, or those using assistive technologies.

Improved security:

Server-side rendering can improve website security by reducing the risk of cross-site scripting (XSS) attacks that are common with client-side rendering.

Better performance on low-powered devices:

Server-side rendering can improve the performance of web pages on low-powered devices such as mobile phones and tablets by reducing the processing requirements on the client-side.

OPPOSITE OF SSR

Higher server load

Server-side rendering requires more server resources to generate and serve pages, which can increase server load and affect website performance.

Limited interactivity

Server-side rendering does not allow for complex interactive features that require heavy client-side processing, such as real-time updates or animations.

Limited flexibility

Server-side rendering may limit the ability to customize the user interface or provide personalized experiences, as the HTML is generated on the server.

Higher development costs

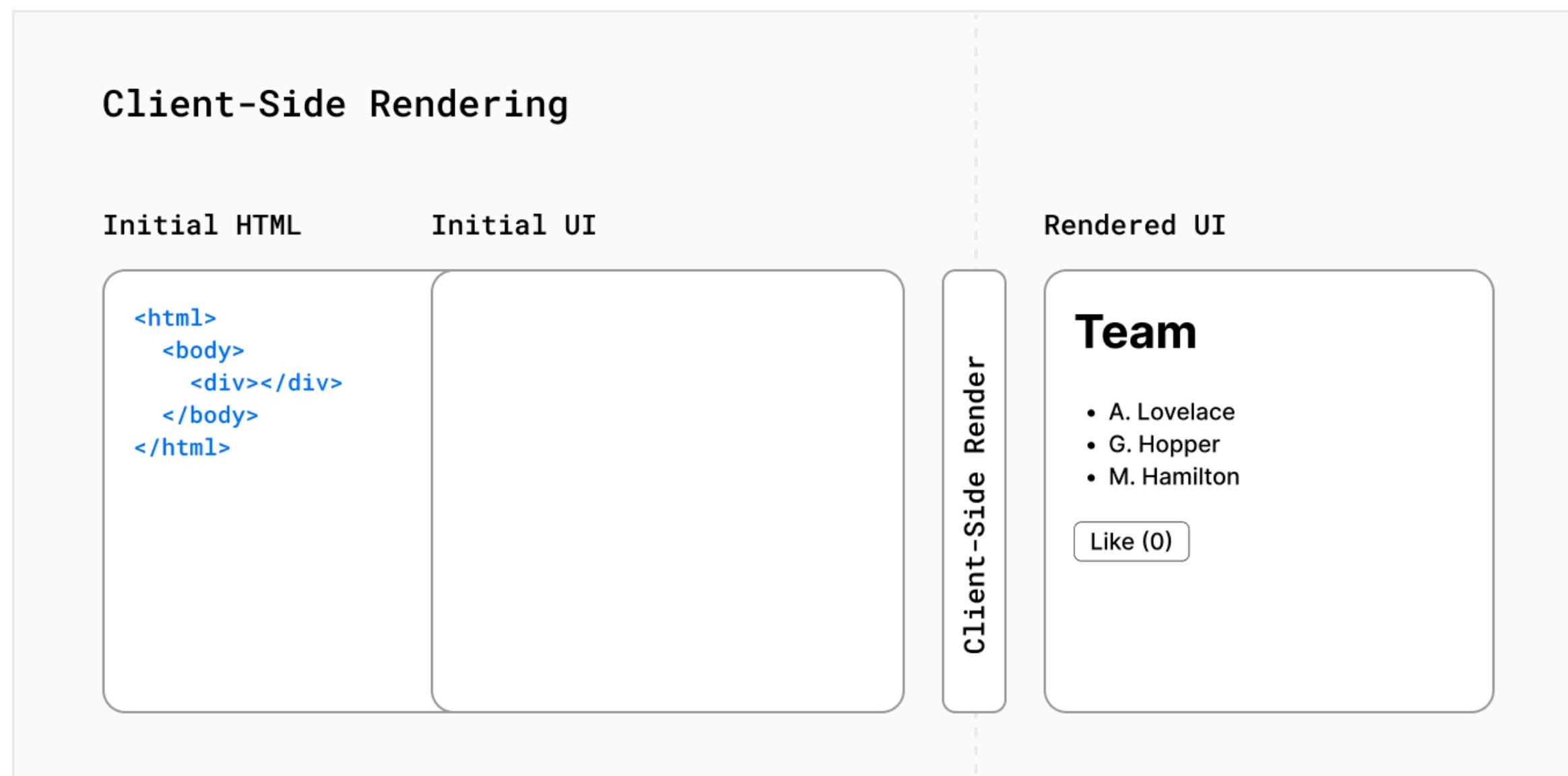
Server-side rendering may require additional development resources and expertise, leading to higher development costs.

Complexity

Implementing server-side rendering can be complex and require specific technologies and configurations, which can be difficult for some developers to manage.

Client-Side Rendering

- With client-side rendering, the HTML of the page is generated inside the web browser.
- This approach allows for faster and more dynamic interactions with the web page as the browser can update specific parts of the page without having to reload the entire page from the server



BENEFITS OF CSR

Improved Performance

Client-side rendering reduces server load and bandwidth consumption, initial page load requires a smaller amount of data to be transferred from the server to the client rendering process is performed on the client-side, which reduces the server's workload and frees up resources.

Better User Experience

Client-side rendering provides a more interactive and dynamic user experience. The content can be updated dynamically, and page navigation is faster, as only the necessary data is fetched from the server.

Easier Development

Client-side rendering frameworks, such as React, Angular, and Vue.js, provide developers with tools and libraries that simplify the development process.

Faster Iteration

With client-side rendering, developers can iterate quickly and test changes in real-time without having to wait for server-side rendering. This speeds up the development process and reduces the time it takes to release new features and updates.

OPPOSITE OF CSR

SEO Challenges

While modern search engines are better at indexing client-side rendered pages, there can still be challenges with search engine optimization (SEO). It can be difficult to ensure that the content is fully visible and crawlable by search engines, which can negatively impact search rankings.

Initial Load Time:

The initial load time of a client-side rendered web page can be slower than that of a server-side rendered page because the web browser needs to download the JavaScript code and execute it before displaying the content. This can result in slower page load times, especially on slower devices or slower internet connections.

Browser Compatibility:

Different web browsers may interpret and execute JavaScript differently, which can lead to compatibility issues. This can require additional testing and development work to ensure that the web page functions correctly across all major web browsers.

Security Concerns:

Client-side rendering can increase the risk of security vulnerabilities, such as cross-site scripting (XSS) attacks. This is because the client-side JavaScript code can be modified by malicious actors, which can lead to data theft or other security breaches.

Maintenance Complexity:

Client-side rendering frameworks and libraries can be complex and require a high level of technical expertise to maintain and update. This can make it challenging for smaller development teams to keep up with the latest updates and ensure that the web page remains secure and functional.

THE OPTIMIZED SOLUTION

- Use Server and Client Components.
- We can build a good combination of SSR, CSR using modern JavaScript .
- Use SSR,CSR Where ncesseassary.





MEET TAILWIND

- 01.** Tailwind CSS is basically a Utility first CSS framework for building rapid custom UI.
- 02.** It is a highly customizable,
- 03.** Low-level CSS framework that gives you all of the building blocks that you need.
- 04.** It is a cool way to write inline styling and achieve an awesome interface without writing a single line of your own CSS.





Learn
with Summit

FULL PLAYLIST LINK

<https://lwsbd.link/tp>



OVERVIEW TUTORIAL

<https://lwsbd.link/tailwind>

GITHUB REPOSITORY LINK

<https://github.com/learnwithsumit/tailwind-playlist/branches/stale>