

Introduction to HTML and CSS



SoftUni Team
Technical Trainers



SoftUni

Software University

<https://softuni.bg>

sli.do

#html-css

1. What is **HTML**?

- HTML Syntax - **Tags** & **Attributes**
- Common HTML **Tags**
- Common HTML **Attributes**

2. What is **CSS**?

- CSS Syntax - **Selectors** & **Rules**
- Adding **CSS** to our **HTML** documents
- Basic **CSS Selectors**





Introduction to HTML

What is HTML?

- HTML is a **markup language**
- HTML is the basis for **creating web pages** and other information that can be displayed in a **web browser**
- Language for expressing **semantic structure** in textual documents



What is HTML?

- HTML is a **language** for **describing web pages**
- HTML documents contain **HTML tags** and **plain text**
- A markup language is a set of **markup tags**
- The tags describe **document content**

HTML Page Structure

```
<!DOCTYPE html>  ← Tells version of HTML
<html>           ← HTML Root Element

<head>           ← Used to contain page HTML metadata
  <title>Page Title</title> ← Title of HTML page
</head>

<body>           ← Hold content of HTML
  <h2>Heading Content</h2> ← HTML heading tag
  <p>Paragraph Content</p> ← HTML paragraph tag
</body>

</html>
```

What is HTML?

- **HTML** is consumed by **web browsers**
- The purpose of a **web browser** is to read HTML documents and compose them into visible or audible web pages.
- **The browser** does not display the HTML tags, it uses the tags to interpret the content of the page.



- Tags are **keywords** surrounded by **angle brackets**

```
'<' + 'html' + '>' ► <html>  
'<' + 'head' + '>' ► <head>  
'<' + 'body' + '>' ► <body>  
'<' + 'p' + '>' ► <p>
```



- HTML tags normally come in **pairs**

```
'<' + 'p' + '>' and '</' + 'p' + '>'
```

- The first tag in a pair is the **start** tag, the second tag is the **end** tag
- The **end** tag is **written like** the **start** tag, with a **forward slash** before the **tag** name
- Start** and **end** tags are also called **opening tags** and **closing tags**

- Examples

Opening tag

'<' + 'html' + '>'

'<' + 'head' + '>'

'<' + 'body' + '>'

'<' + 'p' + '>'

Closing tag

'</' + 'html' + '>'

'</' + 'head' + '>'

'</' + 'body' + '>'

'</' + 'p' + '>'

- Examples

- Describes the **start** and **end** of the **web page/document**

```
<html></html>
```

- Describes the **start** and **end** of the **metadata section** of the **document**

```
<head></head>
```

- Examples

- Describes the **start** and **end** of the web **page/document**

```
<body></body>
```

- Describes the **start** and **end** of the **title section** of the **document**

```
<title></title>
```

Your first **HTML** document

```
<html>
  <head>
    <title>Simple HTML document example</title>
  </head>
  <body>
    <h1>Simple HTML document example</h1>
  </body>
</html>
```

HTML Document Structure

```
<html>  
  
  <head>  
  
    <title>Page title</title>  
  
  </head>  
  
  <body>  
  
    <h1>This is a heading</h1>  
  
    <p>This is a paragraph.</p>  
  
    <p>This is another paragraph.</p>  
  
  </body>  
  
</html>
```

- Time passes, people think of new ways to entertain themselves
 - So **new** versions of HTML have been created
- How do we differentiate between the **different versions**?

HTML Version History

- 1992 The first version of HTML
- 1995 HTML 2.0
- 1996 HTML 3.0 & 3.2
- 1997 HTML 4.0
- 1999 HTML 4.01
- 2008 HTML 5

THE <!DOCTYPE> DECLARATION

- There are many different documents on the web, and a browser can only display an HTML page **100% correctly** if it knows the **HTML type** and **version** used

```
<!DOCTYPE html>
```


Your first **HTML** document

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple HTML document example</title>
  </head>
  <body>
    <h1>Simple HTML document example</h1>
  </body>
</html>
```

■ Headings

```
<h1>This is a level 1 heading</h1>  
<h2>This is a level 2 heading</h2>  
<h3>This is a level 3 heading</h3>  
<h4>This is a level 4 heading</h4>  
<h5>This is a level 5 heading</h5>  
<h6>This is a level 6 heading</h6>
```

■ Paragraphs

```
<p>This is a paragraph</p>
```

- Text formatting

```
<strong>All of this text will appear bold</strong>  
<em>All of this text will appear italic</em>
```

- Links

```
<a href="https://softuni.bg">Click this to go to SoftUni.bg</a>
```

- Images

```

```

■ Ordered List

```
<ol>  
  <li>List item</li>  
  <li>List item</li>  
</ol>
```

■ Unordered List

```
<ul>  
  <li>List item</li>  
  <li>List item</li>  
</ul>
```

■ Definition List

```
<dl>  
  <dt>Definition title</dt>  
  <dd>Definition description</dd>  
  <dt>Definition title</dt>  
  <dd>Definition description</dd>  
</dl>
```

■ Table

```
<table>
  <thead>
    <tr>
      <th>Table header 1</th>
      <th>Table header 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Table cell 1</td>
      <td>Table cell 1</td>
    </tr>
  </tbody>
</table>
```

- Reference Documentation <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

- **Attributes** provide **additional** information about HTML elements
- Tags elements can have **attributes**



Attribute Value

↓

`<p class="my_paragraph" >`

↑

Attribute Name

The diagram shows an HTML tag `<p class="my_paragraph" >` inside a white box. An arrow points from the text 'Attribute Value' above to the value `"my_paragraph"` in the tag. Another arrow points from the text 'Attribute Name' below to the attribute name `class` in the tag.

- Attributes provide **additional** information about an **element**
- Attributes are **always** specified in the **start tag**
- Attributes come in **name/value pairs** like - **name="value"**



The diagram shows an HTML start tag `<p class="my_paragraph" >` inside a white box. An arrow labeled "Attribute Name" points to the word `class`, and another arrow labeled "Attribute Value" points to the string `"my_paragraph"`.

```
<p class="my_paragraph" >
```


- Examples

- **href** – gives the tag the location information for the link

```
<a href="http://initlab.org">init Lab</a>
```

- **src** – tells the tag where to look for the image file

```

```

- Some tips
 - Always **Quote Attribute Values**. Attribute values should always be **enclosed** in quotes
 - **Double style quotes** are the **most common**, but **single style quotes** are also allowed
 - Be careful when **combining single** and **double** quotes, make sure you use **only one type**
 - Reference Documentation - <https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes>

- HEAD Tag
 - The **<head>** element is a container for all the head elements. Elements inside **<head>** can include scripts, instruct the browser where to find style sheets, provide meta information, and more.

```
<html>  
  <head>  
    ...  
  </head>  
  <body>  
  </body>  
</html>
```

- The following tags can be added to the head section:
 - <title>, <style>, <meta>, <link>, <script>, <noscript>
- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head>

```
<html>  
  <head>  
    ...  
  </head>  
  <body>  
  </body>  
</html>
```

- Title

```
<head>  
  <title>HTML Document title</title>  
</head>
```

- Link

- The **<link>** tag defines the relationship between a document and an external resource.

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

■ Meta

```
<head>  
  <!-- Define keywords for search engines: -->  
  <meta name="keywords" content="HTML, CSS, XML, XHTML,  
JavaScript">  
  
  <!--Define a description of your web page:-->  
  <meta name="description" content="Courses on HTML and CSS">  
  
  <!--Define the author of a page:-->  
  <meta name="author" content="Koko">  
</head>
```

Indentation & Code formatting

- We will be using tabs that are 4 spaces long
- **Indentation is extremely important**
 - It helps with navigating the code
 - It helps to find mistakes faster
 - It makes debugging issues faster
- Bad indentation is shameful – when sharing



Introduction to CSS

What is CSS?

- CSS stands for **Cascading Style Sheets**
- Styles define the **visual presentation** of HTML elements
- CSS solved a problem
- HTML was **never** intended to **contain tags** for formatting a document
- With CSS the separation between **semantic content** and **visual presentation** can be achieved again



What is CSS?

- CSS can **save** you a lot of work!
- External visual style guide shared across all pages of your site.
- Change the style guide - change all page's visual presentation.

1. Every **CSS document** is a **collection** of **CSS rules**
2. **CSS rule** has **two main** parts, **Select** and **one or more** declarations
3. Each declaration consists of a **property** and a **value**



- **CSS Rule** has two main parts:

1. Selector
2. One or more declarations

```
[selector] {  
    [declaration]  
    [declaration]  
}
```

- The **CSS Selector** is an identifier of the HTML element or the group of HTML elements you want to style.

```
body {  
    ...  
}
```

- **CSS Declarations** end with a semicolon, and declaration groups are surrounded by curly brackets.

```
{  
    font:16px/1.5 Verdana, sans-serif;  
    color: #333;  
}
```

↓ Selector

```
.my-css-rule {  
  background: red;  
  color: beige;  
  font-size: 1.2rem;  
}
```

Declaration →

Property Value

The diagram illustrates the components of a CSS rule. A blue box highlights the selector '.my-css-rule'. A light blue box highlights the property 'color' in the declaration 'color: beige;'. A light purple box highlights the value 'beige;'. A light green box highlights the property 'font-size' in the declaration 'font-size: 1.2rem;'. Arrows point from the labels 'Selector', 'Declaration', 'Property', and 'Value' to their respective parts in the code.

- In CSS, **selectors** are used to target the **HTML elements** on our web pages that we want to style
- There are a wide variety of CSS selectors available, allowing for fine-grained **precision** when selecting elements to style
 - A CSS selector is the first part of a CSS Rule
- https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors

- **Type selectors**

- The CSS type selector matches elements by **node name**. In other words, it selects all elements of the given type within a document
- https://developer.mozilla.org/en-US/docs/Web/CSS/Type_selectors

```
[tag-name] {  
    [declaration]  
    [declaration]  
}
```

```
h1 {  
    color: red;  
    font-weight: bold;  
}
```

■ Class selectors

- The CSS class selector matches elements based on the contents of their class attribute
- https://developer.mozilla.org/en-US/docs/Web/CSS/Class_selectors

```
[tag-name] {  
  [declaration]  
  [declaration]  
}
```

```
[class-name] {  
  color: red;  
  font-weight: bold;  
}
```

■ ID selectors

- The CSS ID selector matches an element based on the value of the element's id attribute. For the element to be selected, its id attribute must match exactly the value given in the selector
- https://developer.mozilla.org/en-US/docs/Web/CSS/ID_selectors

```
#[tag-name] {  
    [declaration]  
    [declaration]  
}
```

```
#[id-name] {  
    color: red;  
    font-weight: bold;  
}
```

- **Attribute selectors**

- The CSS attribute selector matches elements based on the element having a given attribute explicitly set, with options for defining an attribute value or substring value match

- https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors

```
[attribute="value"] {  
    [declaration]  
    [declaration]  
}
```

```
[href="https://softuni.bg"] {  
    color: red;  
    font-weight: bold;  
}
```

- **Universal selector**

- The CSS universal selector (*) matches elements of any type
- https://developer.mozilla.org/en-US/docs/Web/CSS/Universal_selectors

```
* {  
  [declaration]  
  [declaration]  
}
```

```
* {  
  color: red;  
  font-weight: bold;  
}
```

■ Descendant combinator

- The descendant combinator — typically represented by a single space (" ") character — combines two selectors such that elements matched by the second selector are selected if they have an ancestor (parent, parent's parent, parent's parent's parent, etc.) element matching the first selector
- https://developer.mozilla.org/en-US/docs/Web/CSS/Descendant_combinator

```
header p {  
    color: red;  
    font-weight: bold;  
}
```

■ Selector list

- The CSS selector list (,) selects all the matching nodes. A selector list is a comma-separated list of selectors
- https://developer.mozilla.org/en-US/docs/Web/CSS/Selector_list

```
header, p, div {  
    color: red;  
    font-weight: bold;  
}
```

■ Child combinator

- The child combinator (>) is placed between two CSS selectors. It matches only those elements matched by the second selector that are the direct children of elements matched by the first
- [https://developer.mozilla.org/en-US/docs/Web/CSS/Child combinator](https://developer.mozilla.org/en-US/docs/Web/CSS/Child_combinator)

```
header > p {  
    color: red;  
    font-weight: bold;  
}
```


Adding **CSS** to Our **HTML** Documents

- There are **three ways** of inserting a style sheet:
 - **External** style sheet
 - **Internal** style sheet
 - **Inline** style



- External Style Sheet

- An external style sheet is ideal when the style is applied to **many pages**. With an external style sheet, you can change the **look** of an **entire Web site** by changing **one** file. Each page must link to the style sheet using the tag
- The tag goes inside the **head** section:

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

- **Internal Style Sheet**

- An internal style sheet should be used when a single document has a **unique** style. You define internal styles in the **head** section of an **HTML** page, by using the **style** tag

```
<head>  
  <style>  
    body {  
      ...  
    }  
  </style>  
</head>
```

■ Inline Styles

- An inline style loses many of the advantages of style sheets by mixing content with presentation
- **Do not use** this method unless you have **no other choice**
- To use inline styles, you use the **style** attribute in the relevant tag. The **style** attribute can contain any **CSS property**

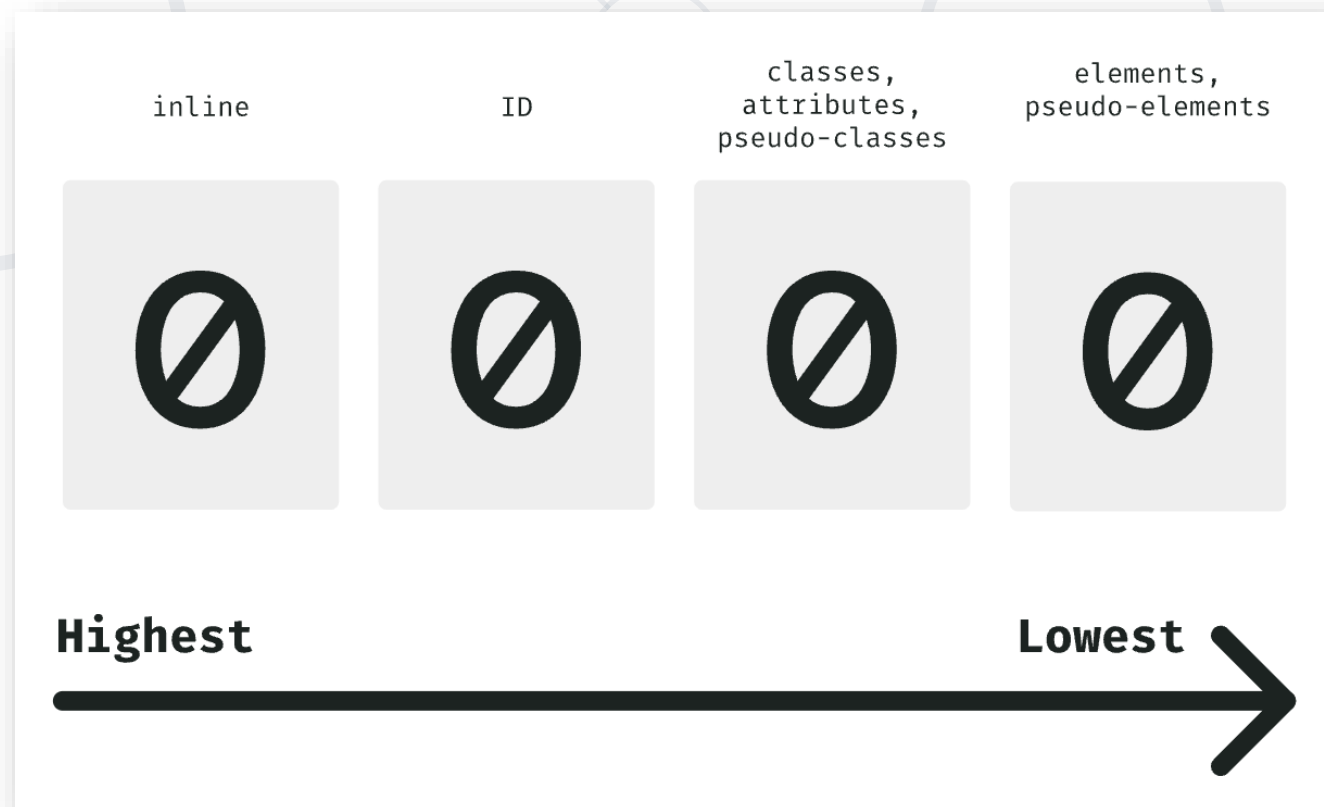
```
<div style="color: red;">
```

```
...
```

```
</div>
```

CSS Selectors Inheritance & Specificity

- CSS relies heavily on specificity and style overwriting
- It's in the name!
- **Cascading** Style Sheets



- **Cascade Order**

- In increasing order of priority.

1. External **<link>**

2. In the **<head>**

3. Inline style attribute

4. Using **!important**

CSS Selector Specificity



- **background**

- **background-color:** #fff;
- **background-image:** url(../image.png);
- **background-repeat:** repeat | repeat-x | repeat-y | no-repeat;
- **background-position:** top left;

■ text

- **color:** #000;
- **text-align:** left | right | center | justify;
- **text-decoration:** underline | overline | line-through | none;
- **text-transform:** uppercase | lowercase | capitalize;
- **text-indent:** 50px;

■ font

- **font-family:** Verdana, Arial, sans-serif
- **font-style:** italic | normal;
- **font-size:** 16px;
- **font-weight:** bold | normal;

- **border**

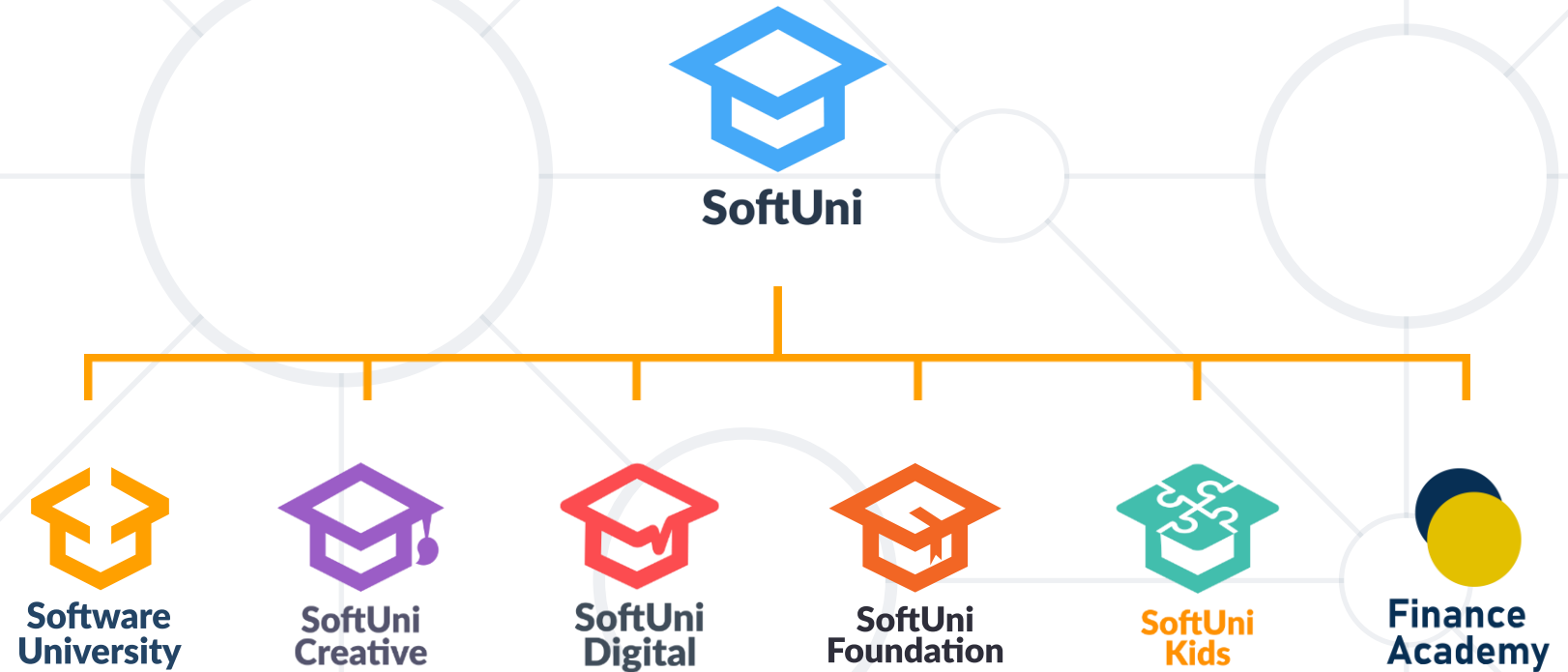
- **border-style:** solid | dotted | dashed | double;
- **border-color:** #C00;
- **border-width:** 2px;

- https://en.wikipedia.org/wiki/Markup_language
- [https://developer.mozilla.org/en-US/docs/Learn/Getting started with the web/How the Web works](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works)
- <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [https://developer.mozilla.org/en-US/docs/Learn/Getting started with the web/HTML basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics)
- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [https://developer.mozilla.org/en-US/docs/Learn/Getting started with the web/CSS basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics)

- What is **HTML**?
 - HTML Syntax - **Tags & Attributes**
 - Common HTML **Tags**
 - Common HTML **Attributes**
- What is **CSS**?
 - CSS Syntax - **Selectors & Rules**
 - Adding **CSS** to our **HTML** documents
 - Basic **CSS Selectors**



Questions?



SoftUni Diamond Partners



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg/>
- © Software University – <https://softuni.bg>



- Software University – High-Quality Education, Profession and Job for Software Developers

- softuni.bg, about.softuni.bg

- Software University Foundation

- softuni.foundation

- Software University @ Facebook

- facebook.com/SoftwareUniversity

- Software University Forums

- forum.softuni.bg



Software University

