

HTML and CSS

Tags and Selectors



SoftUni Team

Technical Trainers



SoftUni



Software University

<https://softuni.bg>

sli.do

#QA-FrontEnd

Table of Contents

1. Introduction to HTML
2. HTML Elements
3. HTML Attributes
4. HTML Metadata
5. Introduction to CSS
6. CSS Selectors
7. CSS Properties
8. Typography in CSS
9. CSS Units





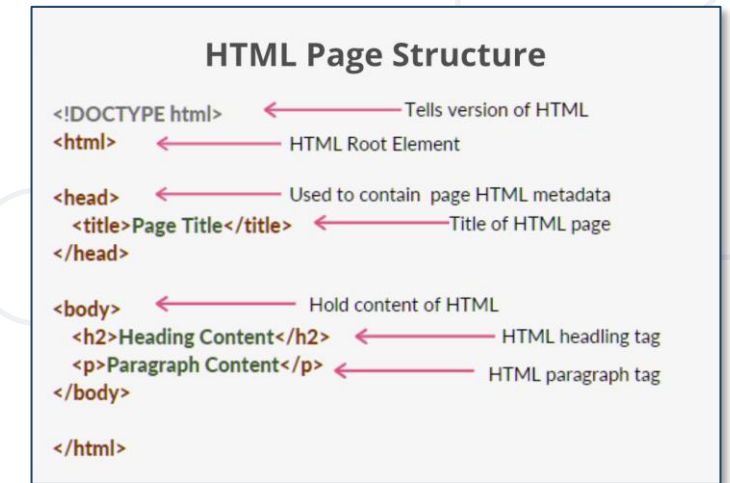
Introduction to HTML

What is HTML?

- HTML == **H**yper **T**ext **M**arkup **L**anguage
- HTML is a **markup** language
 - A markup language is a set of markup **tags**
 - Tags describe **document content**
- Language for expressing **semantic structure** in **textual documents**
 - HTML documents contain HTML tags and plain text
- HTML is the **basis** for creating and describing web pages and other information that can be displayed in a web browser

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



HTML Tag Pairs

- Tags are keywords surrounded by **angle brackets**
- 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
 - Some HTML elements will display correctly, even if you forget the end tag
 - Never rely on this! **Unexpected errors** may result if you forget it
- Start and end tags are also called **opening** tags and **closing** tags

- Root element that describes the **whole** HTML document

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

- Describe the start and end of metadata section of the document

```
<meta> </meta>
```

- Defines the document's body

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

- Describe the start and end of the title of the document

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


HTML Document Structure – Example

```
<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>
```

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>
```



HTML Elements

- 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

```

```

■ 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>
```

■ 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>
```

- Elements documentation

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



src

HTML Attributes

- Tags elements can have **attributes**
 - Attributes provide **additional information** about HTML elements
- Attributes are always **specified** in the **start** tag
- Attributes come in name/value pairs, e.g. **name="value"**

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

Attribute **value**

Attribute **name**

Most Common Tags

- Define the hyperlink
 - Specify the URL of the page the link goes to

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

- Define the path to the location of the image

```

```

- Specify alternate text for an image
 - Used in case the image cannot be displayed

```

```

■ Id

- Specifies an **unique** identifier for a single HTML element
- No more than 1 HTML element
- Syntax

```
<p id="myPara">My Text</p>
```

■ Class

- Specifies a **group** identifier for multiple HTML elements
- More than 1 HTML element
- Syntax

```
<p class="myParas">Text</p>
```



- 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>



<meta>

HTML Metadata

- **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, etc.

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

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

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


- Describe the page title
 - Displayed in the browser's title bar

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

- Define the relationship between a document and external resources

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

- Meta data

```
<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>
```



Introduction to CSS

What is CSS?

- CSS == **C**ascading **S**tyle **S**heets
- Used for **styling** an HTML **document**
 - Defines the **visual presentation** of HTML **elements**
 - Describes color, fonts, spacing, etc.
- External visual style guide shared across all pages of a web page
 - Write CSS **once** and **reuse** the **same stylesheet** in **multiple** HTML pages
 - Change the style guide - change all pages' visual presentation

- Every CSS document is a collection of CSS rules
- CSS **rule** consists of a **selector** and one or more **declarations**
 - Selectors select the HTML element(s) for styling

```
selector {  
    declaration  
    declaration  
}
```

- Each **declaration** consists of a **property** and a **value**

```
div {  
    text-align: justify;  
}
```

property

value

Adding CSS to HTML Documents

- **External** style sheet

```
<link rel="stylesheet" href="mystyle.css">
```

- **Internal** style sheet

```
<style>  
  body {  
    background-color: linen;  
  }  
</style>
```

- **Inline** style

```
<p style="color:red;">This is a paragraph.</p>
```



.id

CSS Selectors

- **Type** selectors
 - Matches elements by **HTML tag** name
 - Selects all elements of the given type within a document

```
tag-name {  
  declaration  
  declaration  
}
```

- Example

```
h1 {  
  color: blue;  
}
```


- **Id** selectors
 - Matches elements based on the contents of their **id** attribute

```
#selector {  
  declaration  
  declaration  
}
```

- Example

```
#paragraph-article {  
  color: blue;  
}
```

- **Class** selectors
 - Matches elements based on the contents of their **class** attribute

```
.selector {  
  declaration  
  declaration  
}
```

- Example

```
.center {  
  color: blue;  
}
```

- **Universal** selectors
 - Matches elements of **any** type

```
* {  
  declaration  
  declaration  
}
```

- Example

```
* {  
  color: blue;  
}
```

- **Grouping** selectors
 - Groups different selectors, pointing to one and the same style
 - Different combinations, adjusted to the developer's needs

```
selector, #selector, .selector {  
    declaration  
}
```

- Example

```
h1, p, #article-title, .article-paragraphs {  
    text-align: center;  
}
```

- **Descendant** combinator selector
 - Represented by a **single space** character (' ')
 - Matches all elements, which are **descendants** of the specified element

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

```
/* ALL of the <p> elements inside the <header> element will be  
selected */
```

- **Child** combinator selector
 - Represented by the **>** character
 - Matches all elements, which are the **children** of the specified element

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

/ ALL of the <p> elements that are children of the <header> element will be selected */*



#fff

CSS Properties

- Background properties
 - background-color
 - background-image
 - background-repeat
 - repeat
 - repeat-x
 - repeat-y
 - no-repeat
 - background-position

- Text properties
 - **color**
 - **text-align**
 - left
 - right
 - center
 - justify
 - **text-indent**

- Text properties
 - **text-decoration**
 - underline
 - overline
 - line-through
 - none
 - **text-transform**
 - uppercase
 - lowercase
 - capitalize

- Font properties
 - **font-family**
 - **font-style**
 - **font-size**
 - **font-weight**
- Border Properties
 - **border-style**
 - **border-color**
 - **border-width**

The background features a light gray geometric pattern consisting of a network of circles and lines. A large, solid dark blue circle is centered in the upper half of the image. Below it, the text "#font-family" is written in white, bold, sans-serif font. Further down, the text "Typography in CSS" is written in a dark blue, bold, sans-serif font. The background pattern includes several smaller circles of varying sizes, some of which are connected by thin gray lines, creating a web-like structure. The overall aesthetic is clean and modern.

#font-family

Typography in CSS

What is Typography?

- Typography == the art and technique of **arranging** type to make written language **legible**, **readable**, and **appealing** when displayed
 - Selecting typefaces
 - Defining point sizes
 - Adjusting spaces between letters
 - Etc.
- Typography is the **visual** component of the written word
- Style or appearance of text

- **Font**

- A computer font (or font) is implemented as a digital data file containing a set of graphically related glyphs, characters, or symbols

- **Font Family**

- In typography, a **font family** (also known as **typeface**) is a set of one or more fonts, each composed of glyphs that share common design features

- **Serif**
 - Small stroke at the edges of each letter
- **Sans-serif**
 - Clean lines
- **Monospace**
 - All letters have same fixed width
- *Cursive*
 - Imitate human writing
- *Fantasy*
 - Decorative fonts

- **Web fonts**
 - Refer to and automatically download remote fonts
 - **@font-face** construct

```
@font-face {  
  font-family: "Trickster";  
  src:  
    local("Trickster"),  
    url("trickster-COLRv1.otf") format("opentype") tech(color-COLRv1),  
    url("trickster-outline.otf") format("opentype"),  
    url("trickster-outline.woff") format("woff");  
}
```


The background features a light gray geometric pattern. It consists of a network of thin lines connecting various circles of different sizes. Some circles are solid white, while others are hollow with a thin gray outline. The lines are also thin and gray, creating a web-like structure across the entire image.

3px

CSS Units

- **Fixed** units

- Values expressed in an absolute unit appear exactly the defined size
- Not recommended to use on screen
 - Only when the output medium is known

- **Absolute** units

- Values are relative to another size property
- Recommended to use on screen
 - Scale better between different rendering mediums

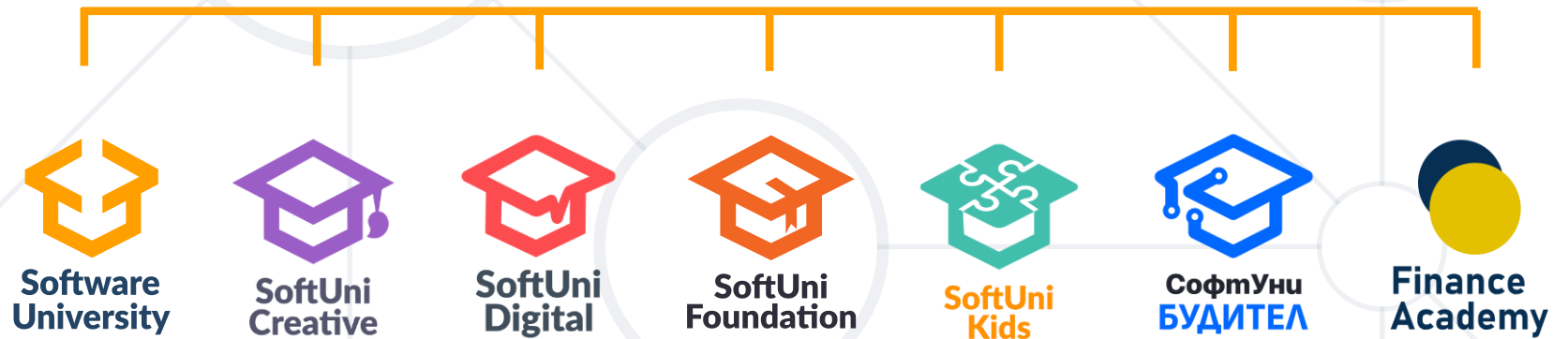
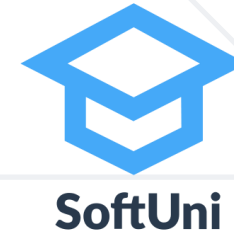


| Fixed Units | Absolute Units |
|-------------------------|--|
| cm (centimeters) | em (relative to the font-size of the element) |
| mm (millimeters) | rem (relative to the font-size of the root) |
| in (inches) | % (relative to the parent element) |
| px (pixels) | vh (relative to 1% of the viewport's height) |
| pt (points) | vw (relative to 1% of the viewport's width) |
| pc (picas) | |

- **HTML** == **H**yper**T**ext **M**arkup **L**anguage
 - Used to create and structure web pages
- **CSS** == **C**ascading **S**tyl**S**heets
 - Used to control the layout and appearance of elements withing web pages
- By **combining** HTML & CSS, developers can create **well-organized** and **good-looking** web pages



Questions?



Diamond Partners



THE CROWN IS YOURS



- 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



- 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>

