



CS474

MULTIMEDIA TECHNOLOGY

Pr. G. Tziritas, Spring 2018

Html – Css Tutorial

G. Simantiris (TA)

OVERVIEW

- Introduction – Definitions
- HTML & History of HTML
- HTML elements, tags, attributes, structure
- HTML basics & examples
- CSS
- Common CSS terms
- Adding CSS to HTML
- CSS basics & examples
- Useful mentions
- References



HTML & CSS

- *HTML*, HyperText Markup Language
gives content structure and meaning by defining that content as, for example, headings, paragraphs, or images.
- *CSS*, Cascading Style Sheets
a presentation language created to style the appearance of content-using, for example, fonts or colors.
- The two languages, HTML and CSS, are independent of one another and should remain that way. CSS should not be written inside of an HTML document and vice versa.
- As a rule, HTML will always represent **content**, and CSS will always represent the **appearance** of that content.



WHAT IS HTML?

HTML is the standard markup language for creating Web pages.

- HTML describes the structure of Web pages using markup.
- HTML **elements** are the building blocks of HTML pages.
- HTML elements are represented by **tags**.
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on.
- Browsers do not display the HTML tags, but use them to render the content of the page.



HTML HISTORY

- 1989: Tim Berners-Lee invented www
- 1991: Tim Berners-Lee invented HTML
- 1993: Dave Raggett drafted HTML+
- 1995: HTML Working Group defined HTML 2.0
- 1997: W3C* Recommendation: HTML 3.2
- 1999: W3C Recommendation: HTML 4.01
- 2000: W3C Recommendation: XHTML 1.0
- 2008: WHATWG** HTML5 First Public Draft
- 2012: WHATWG HTML5 Living Standard
- 2014: W3C Recommendation: HTML5
- 2016: W3C Candidate Recommendation: HTML 5.1

* World Wide Web
Consortium

** Web Hypertext Application
Technology Working Group



UNDERSTANDING COMMON HTML TERMS

- **Elements:**

designators that define the structure and content of objects within a page. Some of the more frequently used elements include multiple levels of headings (identified as **h1** through **h6**) and paragraphs (identified as **p**); the list goes on to include many more.

- **Tags:**

The use of less-than (<) and greater-than (>) angle brackets surrounding an element creates what is known as a tag, e.g. <p>. Tags most commonly occur in pairs of opening and closing tags, e.g. <h1>, </h1>. The content that falls between the opening and closing tags is the content of that element.



UNDERSTANDING COMMON HTML TERMS

○ Attributes:

properties used to provide additional information about an element.

Most common attributes: **id**, which identifies an element; **class**, which classifies an element; **src**, which specifies a source for embeddable content; and **href**, which provides a hyperlink reference to a linked resource.

Attributes are defined within the opening tag, after an element's name, generally including a name and a value. The format for these attributes consists of the attribute name followed by an equals sign and then a quoted attribute value. E.g., an `<a>` element (anchor) including an href attribute would look like:

`Multimedia Technology`



UNDERSTANDING COMMON HTML TERMS

Element:

`Multimedia Technology`

Tag (opening, closing):

`Multimedia Technology`

Attribute (name, value):

`Multimedia Technology`

Content:

`Multimedia Technology`



HTML DOCUMENT STRUCTURE

- HTML documents are plain text documents saved with an **.html** file extension
- You first need a text editor that you are comfortable using → Notepad++
- All HTML documents have a required structure that includes the following declaration and elements:
 - **<!DOCTYPE html>**,
 - **<html>**,
 - **<head>** and
 - **<body>**.



HTML DOCUMENT STRUCTURE

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>This is a web page.</p>
  </body>
</html>
```



HTML DOCUMENT STRUCTURE

<!DOCTYPE html> **Inform web browsers which version of HTML is being used (for the latest version simply *html* is enough.**

```
<html lang="en">  
  <head>  
    <meta charset="utf-8">  
    <title>Hello World</title>  
  </head>  
  <body>  
    <h1>Hello World</h1>  
    <p>This is a web page.</p>  
  </body>  
</html>
```



HTML DOCUMENT STRUCTURE

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>This is a document</p>
  </body>
</html>
```

Beginning and end of the html document.

The *lang* attribute can be used to declare the language of a Web page (or a portion of it).

Meant to assist search engines and browsers.

W3C recommendation: declare the primary language for each Web page. See *ISO Language Codes*.



HTML DOCUMENT STRUCTURE

<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Hello World </head> <body> <h1>Hello World <p>This is a simple </body> </html></pre>	<p>Identifies the top of the document, including any metadata (accompanying information about the page). The content inside is not displayed on the web page itself. Instead, it may include the document title (which is displayed on the title bar in the browser window), links to any external files, or any other beneficial metadata.</p>
---	--




HTML DOCUMENT STRUCTURE

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>This is a v
  </body>
</html>
```

Character encoding of the page.
For HTML5, the default character encoding is UTF-8.

References:
[HTML Character Sets](#),
[HTML Encoding](#),
[HTML <meta> charset Attribute](#)



HTML DOCUMENT STRUCTURE

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>This is a web page.</p>
  </body>
</html>
```

**Title of the document
displayed on the title bar in
the browser window.**



HTML DOCUMENT STRUCTURE

`<!DOCTYPE html>` **All of the visible content
within the web page.**
`<html lang="en">`

`<head>`

`<meta charset="utf-8">`

`<title>Hello World</title>`

`</head>`

`<body>`

`<h1>Hello World</h1>`

`<p>This is a web page.</p>`

`</body>`

`</html>`



HTML DOCUMENT STRUCTURE

```
<!DOCTYPE html>
```

Heading 1.

```
<html lang="en">
```

```
<head>
```

Simple paragraph.

```
<meta charset="utf-8">
```

```
<title>Hello World</title>
```

```
</head>
```

```
<body>
```

```
<h1>Hello World</h1>
```

```
<p>This is a web page.</p>
```

```
</body>
```

```
</html>
```



TIPS

- When an element is placed inside of another element, also known as nested, it is a good idea to **indent** that element to keep the document structure well organized and legible.

- **Self-Closing/empty Elements**

In the previous example, the *meta* element had only one tag and didn't include a closing tag. Not all elements consist of opening and closing tags. Some elements simply receive their content or behavior from attributes within a single tag.

Other common selfclosing elements: *br*, *hr*, *img* etc. Sometimes we have to close empty elements like this: `
` which is the same as `
`.



LET'S PRACTISE

- Open a text editor.
- Write (or copy-paste) the previous html code.
- Save it as HelloWorld.html.
- Open the file with a web browser.

Try the following:

- omit some tags and attributes,
- insert multiple spaces, newlines and white space characters in general,
- write the whole code in one line,
- ...



HEADINGS

- HTML headings are defined with the **<h1>** (most important heading) to **<h6>** (least important heading) tags.
- **Headings Are Important:** Search engines use the headings to index the structure and content of your web pages. Users skim your pages by its headings. It is important to use headings to show the document structure.

PREFORMATTED TEXT

- The HTML **pre** element defines preformatted text. The text inside a is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks.




HORIZONTAL RULES

- The **<hr>** tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.
- The **<hr>** element is used to separate content (or define a change) in an HTML page.

LINKS

- HTML links are defined with the **<a>** tag. The link's destination is specified in the **href attribute**. E.g.:

```
<a href="http://www.csd.uoc.gr/~hy474/ ">Multimedia Technology</a>  
<a href="./intro.pdf ">introduction</a>
```



IMAGES

- HTML images are defined with the **** tag. The source file (src), alternative text (alt), and size are provided as attributes. E.g.:

```

```

```

```

```

```

```

```



MORE ELEMENTS / TAGS (FORMATTING)

- **br**: inserts a single line break
- **b**: bold text
- **em**: emphasized text
- **i**: italic text
- **small**: smaller text
- **strong**: important text
- **sub**: subscripted text
- **sup**: superscripted text
- **ins**: inserted text
- **del**: deleted text
- **mark**: marked/highlighted text



MORE ATTRIBUTES

- **style:** used to specify the styling of an element, like color, font, size etc.
- **title:** added to the **p** element. The value of the title attribute will be displayed as a tooltip when you mouse over the paragraph.
- **id:** specifies a unique id for an element.
- **class:** specifies one or more class names for an element (used by CSS and JavaScript).

`<p style="color:red">I am a paragraph</p>`

`<p title="I'm a tooltip">This is a paragraph.</p>`

`<p id="par3">This is the 3rd paragraph.</p>`



HTML STYLES

`<tagname style="property:value;">`

property and *value* are CSS. For a quick, one-time styling you can use it inline an HTML element.

Better define styles in a separate CSS file!

- **background-color:** background color
- **color:** text color
- **font-family:** font to be used
- **font-size:** text size
- **text-align:** horizontal text alignment

```
<p style="background-color:powderblue;
color:red; font-family:verdana; font-size:300%;
text-align:center;">This is a paragraph.</p>
```



QUOTATION & CITATION ELEMENTS

- **abbr:** abbreviation or acronym
- **address:** contact information
- **blockquote:** section that is quoted from another source
- **cite:** the title of a work
- **q:** short inline quotation

Try them.... ☺



COMMENTS IN HTML

```
<!-- Write your comments here -->
```

Notice that there is an exclamation point (!) in the opening tag, but not in the closing tag. Comments are not displayed by the browser, but they can help document your HTML source code. With comments you can place notifications and reminders in your HTML

```
<!-- This is a comment -->
```

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

```
<!-- Remember to add more information here -->
```

```
<!-- Do not display this at the moment
```

```

```

```
-->
```



HTML COLORS

- Specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.
- Used for background, text, border.

```
<h1 style="background-color:DodgerBlue;  
border:4px solid Tomato;">A nice  
heading</h1>
```

```
<p style="background-color:black;  
color:Orange; font-size:200%">A paragraph  
again.</p>
```

- Notice that the value **Tomato** can be written equally as: **rgb(255, 99, 71)** or **#ff6347** or **hsl(9, 100%, 64%)**
- Useful tool: [HTML Color Picker](#)



WHAT IS CSS?

- Stands for **Cascading Style Sheets**.
- Defines the visual style and appearance of our web pages.
- Describes how HTML elements are to be displayed on screen, paper, or in other media.
- **Saves a lot of work**. It can control the layout of multiple web pages all at once.
- Uses selectors, properties and values to achieve its purpose.



HISTORY OF CSS

- HTML was NEVER intended to contain tags for formatting a web page! It was created to **describe the content** of a web page.
- When tags like , and color attributes were added to the HTML 3.2, it started a *nightmare* for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.
- CSS removed the style formatting from the HTML page!
- The style definitions are normally saved in external .css files → you can change the look of an entire website by changing just one file!



SELECTORS

- Designate exactly which element or elements within our HTML to target and apply styles (such as color, size, and position) to.
- May include a combination of different qualifiers to select unique elements, all depending on how specific we wish to be (e.g. every paragraph on a page, or only one specific paragraph on a page).
- Generally target an **attribute value**, such as an id or class value, or target the **type of element**, such as h1 or p.
- Within CSS, selectors are followed with curly brackets, { }, which encompass the styles to be applied to the selected element, e.g. **p** { ... } targets all paragraph elements.



PROPERTIES

- Determines the styles that will be applied to a selected element.
- Property names fall after a selector, within the curly brackets, {}, and immediately preceding a colon :
- There are numerous properties we can use, such as background, color, font-size, height, width, and new properties are often added. E.g.:

```
p {  
    color: ....;  
    font-size: ...;  
}
```

defines the color and font-size properties to be applied to all p elements.



VALUES

- Determines the behavior of a property.
- Values can be identified as the text between the colon : and semicolon ;
- Example: selecting all p elements and setting the value of the color property to be orange and the value of the font-size property to be 16 pixels.

```
p {  
    color:orange;  
    font-size:16px;  
}
```



ADDING CSS TO HTML

CSS can be added to HTML elements in 3 ways:

- **Inline:** by using the style attribute in HTML elements.
- **Internal:** by using a style element in the head section.
- **External:** by using an external CSS file.

The most common way to add CSS, is to keep the styles in **separate CSS files**.



INLINE CSS

- Is used to apply a unique style to a **single HTML element**.
- Uses the **style** attribute of an HTML element.

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```



INTERNAL CSS

- Is used to define a style for a **single HTML page**.
- is defined in the <head> section of an HTML page, within a <style> element.



INTERNAL CSS

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {background-color:powderblue;}
      h1   {color:blue;}
      p    {color:red;}
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```



EXTERNAL CSS

- Is used to define the style for **many HTML pages**.
- **You can change the look of an entire web site, by changing one file!**
- To use an external style sheet, add a link to it in the <head> section of the HTML page:

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



LET'S PRACTISE

File HelloWorldwCss.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet" href= "HWstyles.css">
    <meta charset="utf-8">
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>This is a web page.</p>
  </body>
</html>
```



LET'S PRACTISE

File HWstyles.css

```
body {  
    background-color: powderblue;  
}  
h1 {  
    color: blue;  
}  
p {  
    color: red;  
}
```



LET'S PRACTISE

- Open a text editor.
- Write (or copy-paste) the previous html and css code.
- Save it as HelloWorldwCss.html and HWstyles.css.
- Open the html file with a web browser.
- Experiment with different attributes, use the same styling on a different page, change the looks of the entire page etc.
- Keep your code indented!
- If multiple files are used for styling, keep them under one directory, e.g.: stylesheets/main.css, stylesheets/fonts.css



WORKING WITH SELECTORS

- Selectors indicate which HTML elements are being styled.
- It is important to fully understand how to use selectors and how they can be leveraged.
- The first step is to become familiar with the different types of selectors.
- Most common selectors: *type*, *class*, and *id*.



TYPE SELECTORS

- Target elements by their element type. For example, should we wish to target all division elements, `<div>`, we would use a type selector of `div`.
- (
- The `<div>` tag defines a division or a section in an HTML document.
- The `<div>` tag is used to group block-elements to format them with CSS.

)

CSS

```
div { ... }
```

HTML

```
<div> ... </div>
```



CLASS SELECTORS

- Allow us to select an element based on the element's class attribute value.
- They are a little more specific than type selectors, as they select a particular group of elements rather than all elements of one type.
- Allow us to apply the same styles to different elements at once by using the same class attribute value across multiple elements.
- Within CSS, classes are denoted by a leading period . followed by the class attribute value.

CSS

```
.cname { ... }
```

```
p.cname { ... }
```

HTML

```
<div class="cname">...</div>
```

```
<p class="cname">...</p>
```



ID SELECTORS

- Are even more precise than class selectors.
- They target only one unique element at a time.
- Use an element's id attribute value as a selector.
- Regardless of which type of element they appear on, id attribute values **can only be used once per page** → they should be reserved for significant elements.
- Within CSS, id selectors are denoted by a leading hash sign # followed by the id attribute value.

CSS

```
#idname { ... }
```

HTML

```
<div id="idname">...</div>
```



ADDITIONAL SELECTORS STUFF

- Grouping Selectors if you have elements with the same style definitions (minimizes the code):

```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```

- CSS comments:

```
p {  
    color: red;  
    /* This is a single-line comment */  
    text-align: center;  
}
```

```
/* This is  
a multi-line  
comment */
```



MULTIPLE STYLE SHEETS

- If some properties have been defined for the same selector (element) in different style sheets, the value from the **last read** style sheet will be used.

```
<head>  
<link rel="stylesheet" href="mystyle.css">  
<style>  
h1 {  
    color:orange;  
}  
</style>  
</head>
```



```
h1 {  
    color:navy;  
}
```

- What will be the h1 text color?



CASCADING ORDER

- What style will be used when there is more than one style specified for an HTML element?
- Generally all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:
 - 1. Inline style (inside an HTML element)**
 - 2. External and internal style sheets (in the head section)**
 - 3. Browser default**
- So, an inline style (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or a browser default value.



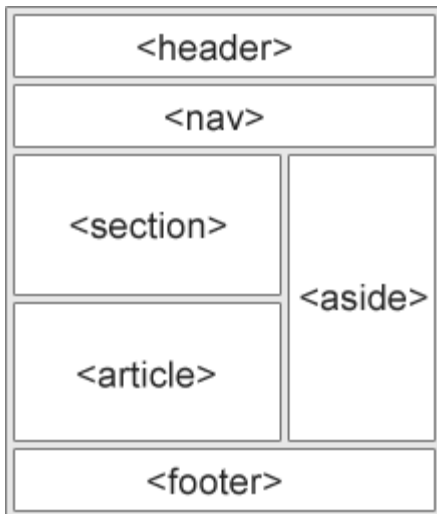
USEFUL MENTIONS

- **Image Maps:** image with clickable areas (see [example](#)).
- **Tables:** defined with the `<table>` tag. Each table row is defined with the `<tr>` tag. A table header is defined with the `<th>` tag (by default, table headings are bold and centered). A table data/cell is defined with the `<td>` tag (see [example](#)).
- **Lists:** An unordered list starts with the `` tag. Each list item starts with the `` tag. The list items will be marked with bullets (small black circles) by default. An ordered list starts with the `` tag. Each list item starts with the `` tag. The list items will be marked with numbers by default. ([Examples](#))



USEFUL MENTIONS ...

- **Iframes:** used to display a web page within a web page and is defined with the `<iframe>` tag (see [example](#)).
- **JavaScript:** makes HTML pages more dynamic and interactive.
- **Layout:** HTML5 offers new semantic elements that define the different parts of a web page:



- Layout techniques:
 1. HTML tables
 2. CSS float property
 3. CSS framework
 4. CSS flexbox

[Find pros and cons here!](#)

USEFUL MENTIONS ...

- **Forms**: The HTML `<form>` element defines a form that is used to collect user input. Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.
- **Responsive Web Design**: makes your web page look good on all devices (desktops, tablets, and phones). Responsive Web Design is about using HTML and CSS to *resize, hide, shrink, enlarge, or move* the content to make it **look good on any screen!**



USEFUL MENTIONS ...

- **Web Templates:** get already developed webpage or website templates (either for a cost or for free), customize them to your needs and add your own content.
- **CSS resets:** Every web browser has its own default styles for different elements. How browser1 renders headings, paragraphs, lists, and so forth may be different from how browser2 does. To ensure **cross-browser compatibility**, CSS resets have become widely used.

CSS resets take every common HTML element with a predefined style and provide one unified style for all browsers. These resets generally involve removing any sizing, margins, paddings, or additional styles and toning these values down. Because CSS cascades from top to bottom—more on that soon—our reset needs to be at the very top of our style sheet. Doing so ensures that those styles are read first and that all of the different web browsers are working from a common baseline.

Resets available to use: [Eric Meyer's reset](#), [Normalize.css](#).



REFERENCES

- ❖ [w3schools.com](#)
- ❖ [HTML5](#)
- ❖ [HTML5 Tutorial](#)
- ❖ [CSS Tutorial](#)
- ❖ [HTML Element Reference](#)
- ❖ [CSS Reference](#)
- ❖ [HTML5 Tutorial \(2\)](#)
- ❖ [CSS Tutorial \(2\)](#)

