HTML Elements

Parts 1 & 2

Shadi Lahham - Programmazione web - Frontend - HTML e CSS

Intro to HTML

Remember the HTML Boilerplate

```html
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Website Title</title>
  <meta name="description" content="My new wonderful website">
  <meta name="author" content="Mister X">
  <link rel="stylesheet" href="css/styles.css?v=1.0">
  <!--[if lt IE 9]>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.js"></script>
  <![endif]-->
</head>

<body>
  <div>My Website</div>
  <!-- end of the body -->
  <script src="js/scripts.js"></script>
</body>
</html>
```

History of HTML

- Invented by Tim Berners-Lee, first web page published August 6, 1991
- Created "hypertext" to share scientific papers
- Intended as a standard way to structuring documents
- Standardized by W3C (World Wide Web Consortium)

History of HTML

- HyperText Markup Language
- Early 90s
- HTML4 in 1997
- XHTML in 2000
- HTML5 in 2014

Terminology

Web design:
The process of planning, structuring and creating a website

Web development:
The process of programming dynamic web applications

Front end:
The outwardly visible elements of a website or application

Back end:
The inner workings and functionality of a website or application.

Anatomy of an HTML Element

- Element
    - Building blocks of web pages - an individual component of HTML
    - Examples: paragraph, heading, table, list, div, link, image, etc.
- Tag
    - Opening tag marks the beginning of an element & closing tag marks the end
    - Tags contain characters that indicate the tag's purpose

```
<tagname>content</tagname>

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

Anatomy of an HTML Element

- Container Element
  - An element that can contain other elements or content
  - A paragraph (<p>) contains text
- Stand Alone Element
  - An element that cannot contain anything else

`<br />`

`<img />`

Anatomy of an HTML Element

- Attribute

  ○ Provides additional information about the HTML element
  ○ Placed inside an opening tag, before the right angle bracket.
  ○ Examples: class, id, language, style, source

- Value

  ○ Value is the value assigned to a given attribute.
  ○ Values must be contained inside quotation marks.

```
<div id="copyright">Copyright message</div>

<img src="my_picture.jpg" />

<a href="http://bbc.co.uk">The BBC</a>
```

The Doctype

The first thing on an HTML page is the doctype, which tells the browser
which version of the markup language the page is using.

For XHTML 1.0 Strict:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

For HTML4 Transitional:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
```

For HTML5:
```
<!doctype html>
```

HTML Tag

- After <doctype>, the page content must be contained between <html> tags.

```
<!DOCTYPE html>

<html>

</html>
```

Head and Body Tags

Head:
The head contains the title of the page & meta information about the page.
Meta information is not visible to the user, but has many purposes, like
providing information to search engines.

Body:
The body contains the actual content of the page. Everything that is
contained in the body is visible to the user.

Review - page structure

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Title of the page </title>
  </head>
  <body>
    The page content here.
  </body>
</html>
```

Nesting

All elements "nest" inside one another

Nesting is what happens when you put other containing tags inside other containing tags.

For example, you would put the <p> inside of the <body> tags. The <p> is now nested inside the <body>

Nesting: Example

Elements are 'nested' inside the `<body>` tag.

```
<body>
  <p>A paragraph inside the body tag</p>
</body>
```

Paragraphs 'nested' inside list items.

```
<ul>
  <li>
    <p>A paragraph inside a list item</p>
  </li>
</ul>
```

Comments

Comments in your code will not be seen by the browser, but only when viewing the code in the editor.

<!-- this is a comment -->

Use comments to organize your code into sections so that it can be easily understood.

<!-- Beginning of header -->
<**div** id="header">Header Content </**div**>
<!-- End of header -->

Comments

You can also 'comment out' large chunks of code to hide it from the browser, but remove commented code in your final delivery.

```
<!-- TODO: uncomment or remove disabled code -->

<!--
<ol>
<li>List Item</li>
<li>Another List Item</li>
</ol>
-->
```

Element: Paragraph

Paragraphs allow you to format your content in a readable fashion.

```
<p>Paragraph 1</p>
<p>Paragraph 2</p>
<p>Paragraph 3</p>

-- OR --
<p>Paragraph 1</p> <p>Paragraph 2</p>
<p>Paragraph 3</p>

-- OR --
<p>Paragraph 1</p>

<p>Paragraph 2</p>
   <p>Paragraph 3</p>
```

White space is only for humans.

Lorem ipsum dolor sit amet consectetuer accumsan Sed vitae mauris risus. Magna eget a Phasellus a nibh nisl risus netus mauris nec. Mus nibh non scelerisque Curabitur et Lorem Nunc lacus Vivamus nibh. Pharetra Nam sagittis.

Lorem ipsum dolor sit amet consectetuer accumsan Sed vitae mauris risus. Magna eget a Phasellus a nibh nisl risus netus mauris nec. Mus nibh non scelerisque Curabitur et Lorem Nunc lacus Vivamus nibh. Pharetra Nam sagittis.

Lorem ipsum dolor sit amet consectetuer accumsan Sed vitae mauris risus. Magna eget a Phasellus a nibh nisl risus netus mauris nec. Mus nibh non scelerisque Curabitur et Lorem Nunc lacus Vivamus nibh. Pharetra Nam sagittis.

Heading number indicates hierarchy, not size

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

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Formatted Text

em and **strong** are meant to indicate meaning through style. If you want to have italicized for appearance and not to communicate meaning, you should use CSS.

```html
<p>
  Here is a paragraph with <em>emphasized</em>
text and <strong>important</strong> text.
</p>
```

Here is a paragraph with *emphasized* text and **important** text.

Formatted Text

b and *i* produce bold and italic text, but are not semantic.

According to the HTML 5 specification, the these tags should be used as a LAST resort when no other tag is more appropriate.

```
<p>
  Here is a paragraph with <i>italic</i> text
and <b>bold</b> text.
</p>
```

Here is a paragraph with *italic* text and **bold** text.

Element: Line Break

```
<p>
  Imagine there's no Heaven<br/>
  It's easy if you try<br/>
  No hell below us<br/>
  Above us only sky
</p>
```

Note: It's not good convention to put line breaks inside paragraphs.

Imagine there's no Heaven
It's easy if you try
No hell below us
Above us only sky

Use the <hr> tag to define a thematic change
in the content

```
<h1>HTML</h1>
<p>HTML is a language for describing web
pages.....</p>

<hr>

<h1>CSS</h1>
<p>CSS defines how to display HTML
elements.....</p>
```

HTML

HTML is a language for describing web pages.....

CSS

CSS defines how to display HTML elements.....

Element Details: Attributes

- Attributes describe additional characteristics of an HTML element
- An attribute has 2 parts: name & value.
- Values should be contained inside quotation marks.

Syntax:

```
<tagname name="value">content</tagname>
```

Element: Anchor

Linking to anchors is very similar to normal links:
- Normal links always point to the top of a page
- Anchors point to a place within a page

To link to an anchor:
1. Create the anchor
2. Create a link pointing to the anchor

1.Anchor
- tag: `<a></a>`
- Name attribute: "chapter4"

2. Link to the anchor
- tag: `<a></a>`
- href attribute starts with **#**: "#chapter4"
- title attribute: "The 4th Chapter"

Element: Anchor - syntax

Anchor
```
<a name="chapter4"></a>
```

Link to the anchor
```
<a href="#chapter4" title="The 4th Chapter">Go to chapter 4</a>
```

Link an anchor in an external page
```
<a href="https://developer.mozilla.org/en-US/docs/Web/CSS/Reference#Selectors">MDN CSS selectors</a>
```

Link to the top of the page
```
<a href="#top">Back to top of page</a>
```

Note:
href="#top" or the empty fragment (**href="#"**) can be used to link to the top of the current page

Element: Link

Links have three components

- tag: <a>
- href attribute: "http://www.bbc.co.uk"
- title attribute: "The BBC"

The <a> tag surrounds text or images to turn them into links

Syntax:

```
<a href="http://www.bbc.co.uk" title="BCC UK website">The BBC UK website</a>
```

Link Attributes

Links can have attributes that tell the link to do different actions like
- open in a new tab
- launch your e-mail program

`<a href="home.html" target="_blank">Link Text</a>`
Link opens in a new window/tab with **target="_blank"**

`<a href="mailto:shadi.lahham@its-ictpiemonte.it">E-mail me!</a>`
Adding **mailto:** directly before the email address means the link will open in the default email program.

Link Targets

_blank
Opens the linked document in a new window or tab

_self
Opens the linked document in the same frame as it was clicked (this is default)

_parent
Opens the linked document in the parent frame

_top
Opens the linked document in the full body of the window

framename
Opens the linked document in a named frame

You will generally use _blank or the default _self (can be omitted)

Relative vs. Absolute Paths for Links & Images

- Relative: paths change depending on the page the link is on.
  - Links within the same directory need no path information. "filename.jpg"
  - Subdirectories are listed without preceding slashes. "img/filename.jpg"

```
<a href="cat-photo.jpg">A file in same folder</a>
<a href="projects/index.html">A file from the projects folder</a>
```

- Absolute: paths refer to a specific location of a file, including the domain.
  - Typically used when pointing to a link that is not within your own domain.
  - Example: https://www.bbc.com/news/world

```
<a href="https://www.bbc.com/news/world">BBC world</a>
```

Empty Elements

Container Element

An element that can contain other elements or content

`<a href="http://example.com">A link element contains text.</a>`

Empty Element

An element that cannot contain anything else

`<br/>`
`<img/>`

Element: Image

Images have three parts:

- **`<img />` tag**
- **src attribute: the location and name of the image file**
- **alt attribute: a brief description of the image content**

`<img src="img/circle-logo.svg" alt="brand logo"/>`

`<img src="img/blinky-lights.png" alt="blinky lights"/>`

`<img src="img/simon-the-cat.gif" alt="simon my cat"/>`

Note: alt should always be populated for accessibility and SEO purposes.

Element: figure

```
<figure>
  <img src="pic_trulli.jpg" alt="Trulli">
  <figcaption>Fig.1 - Trulli, Puglia, Italy.</figcaption>
</figure>

Note: <figure> is supported by modern browsers.
```
<figure> browser support

Element: Unordered and ordered lists
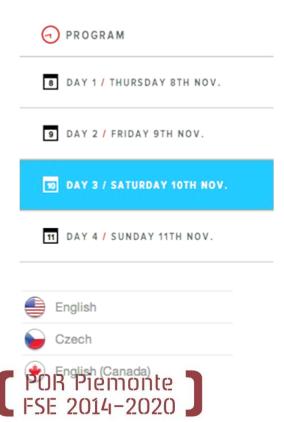
```
<ul>
  <li>List Item</li>
  <li>Another List Item</li>
</ul>
<ol>
  <li>List Item</li>
  <li>Another List Item</li>
</ol>
```

- List Item
- Another List Item

1. List Item
2. Another List Item

Lists: Examples

Tables

Tables are a way to represent complex information in a grid format.

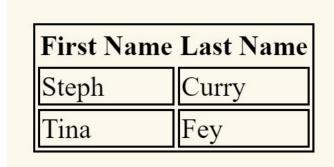Tables are made up of header row(s) with the column names, and body rows with the data

Element: Table

```html
<table>
 <thead>
  <tr>
    <th>First Name</th>
    <th>Last Name</th>
  </tr>
 </thead>
 <tbody>
  <tr>
    <td>Steph</td>
    <td>Curry</td>
  </tr>
  <tr>
    <td>Tina</td>
    <td>Fey</td>
  </tr>
 </tbody>
</table>
```

First Name	Last Name
Steph	Curry
Tina	Fey

Character Codes

There are character codes for many different characters in many different languages

[HTML Character Escape Codes and Entity References](#)

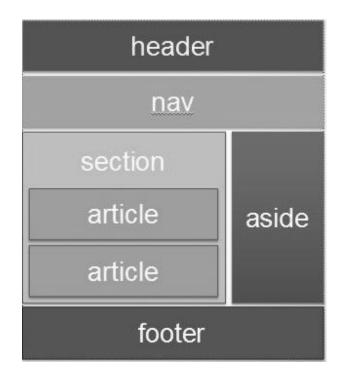| Copyright	©	©
Less than	<	<
Greater than	>	>
Ampersand	&	&

HTML5 elements

HTML5 offers new elements for better document structure and semantics

Some of the most commonly used new tags include:

<**header**></**header**>
<**nav**></**nav**>
<**main**></**main**>
<**section**></**section**>
<**article**></**article**>
<**aside**></**aside**>
<**footer**></**footer**>

HTML5 Semantic Elements
HTML5 - New elements - complete list

Deprecated elements

- A deprecated element or attribute is one that has been outdated by newer constructs
- Deprecated elements are defined references in appropriate locations, but are clearly marked as deprecated
- Deprecated elements may become obsolete in future versions of HTML
- Deprecated elements should not be used

MDN - Obsolete and deprecated elements
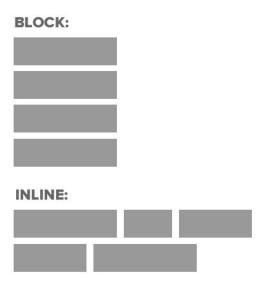
Deprecated HTML Tags

Inline vs Block

Block elements.

Appear on the next line, like paragraphs

Inline elements.

Appear on the same line that they are written on

BLOCK:

INLINE:

Inline vs Block

- CSS divides HTML into two types: inline and block.
- After block elements, browsers render a new line.
- Example of inline elements:
  - img, a, br, em, strong
- Example of block elements:
  - p, h1, ul, li, almost everything else

Element: div

- Block level element. Each new div is rendered on a new line.
- A division, or section of content within an HTML page.
- Used to group elements to format them with CSS.
- Apply IDs and Classes to divs to control their styles with CSS.

Element: div

```
<div>
  <p>Content<p>
  <p>Content<p>
</div>
<div>
  <h1>Main Heading<h1>
</div>
<div>
  <p>Some more content<p>
</div>
```

Element: div

- The div tag is used everywhere to group elements together into sections
- You can wrap groups of elements in a div to style them differently

Element: div

```html
<div style="color: purple;">
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit</p>
  <p>Sed do eiusmod tempor incididunt ut labore et dolore.</p>
</div>
<p>Magna aliqua. Ut enim ad minim veniam.</p>
<p>Quis nostrud exercitation ullamco.</p>
```

Element: span

- Inline element
- Each new span is rendered next to each other & only wraps when it reaches the edge of the containing element
- Can be used to apply styles to text inline so as not to break the flow of content

Element: span

span is used to apply a specific style inline

```html
<p>Paragraph with <span style="color:teal;">teal</span> text.</p>
```

Your turn

1. Basic text

- Build your first HTML page using a correct skeleton structure
- Add some content to your site
- Add one of each level of heading with 1-2 short paragraphs of text below each heading.
- Use <strong> and <em> within a few paragraphs
- Use <span> to highlight some text in the paragraphs

2. Favorite

- Make a page about your favorite food, animal, or activity.
  - For text: Either write your own, or copy and paste from a Wikipedia article, or generate text from an Ipsum like: Veggie Ipsum, Hipster Ipsum, Cupcake Ipsum, LoremIpsum
  - Add paragraph tags to break it into at least 3 paragraphs.
  - Add formatting tags to emphasize "important" words.
  - Add headings to describe your amazing paragraphs.

- Bonus: add fun characters from copypastecharacter

3. Links and images

- Create a page with at least the following elements.
  - At least one image and line break
  - A link that opens in the same window
  - A link that opens in a new window
  - A link to an email address
  - Turn one of the images you added into a link
  - Add other elements that you learned from previous exercises to complete the page as much as possible with meaningful content

4. Lists and tables

- Create a page with at least the following elements.
  - One ordered list and one unordered list
  - A list of links
  - A list of images
  - One or more tables
  - Putting other elements inside your table, like images and lists

5. Complete HTML5 page

How would the following markup look in HTML5?

```html
<div id="header">
    <h1>My Page Title</h1>
</div>
<div id="content">
    <p>The main content</p>
</div>
<div id="sidebar">
    <p>Some stuff in a sidebar</p>
</div>
<div id="footer">
    <p>Copyright me</p>
</div>
```

Create a complete and valid HTML5 page with required elements.
Add real content and other HTML tags that you have learned in this unit.

References

When to use target="_blank"

Why External Links Should Open in New Tabs

The Anchor element

HTML Ordered, Unordered, and Definition Lists

HTML - Lists

References

Image formats for the web

What's the Difference Between PNG, JPEG, GIF, and TIFF? | Widen

PNG vs. SVG: What are the differences? | Adobe

Solved - Best Image Format for the Web? PNG, JPG, GIF, and SVG.

What are SVG Images, and why are they so great?

5 Advantages to Using SVG Files