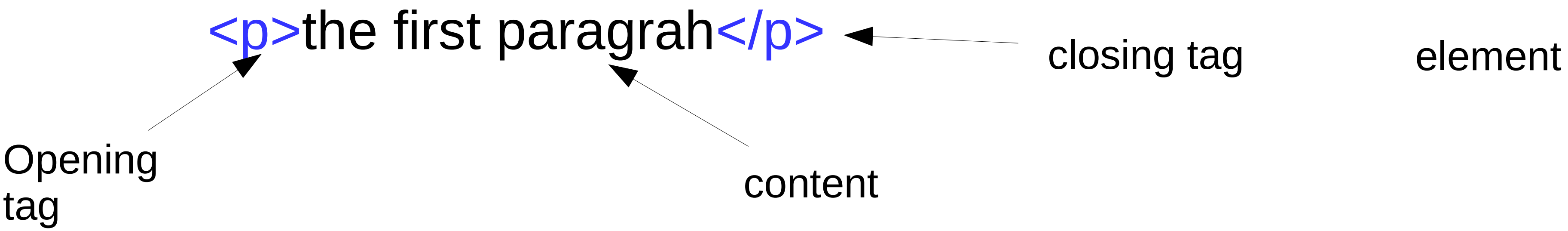
HTML page structure

- HTML markup language used to define web page elements such as heading, paragraph, list, image, link etc.
- CSS used to style HTML (for example alter your text size and fonts used, add borders and drop shadows, layout your page with multiple columns, add animations and other visual effects)
- JavaScript to add dynamic functionality to web pages

HTML elements

- HTML consists of a series of elements, which you use to enclose, wrap, or mark up different parts of the content to make it appear or act a certain way. The enclosing tags can make a bit of content into a hyperlink to link to another page on the web, italicize words, and so on. For example, take the following line of content:



<p>the first paragrah</p>

closing tag

element

Opening tag

content

- case-insensitive

Nesting elements

- You can put elements inside other elements too — this is called nesting.

```
<p>My <strong>first</strong> paragraph</p>
```

- Elements need to be nested correctly inside other elements

Block and inline elements

- Block-level elements form a visible block on a page, they will appear on a new line from whatever content went before it, and any content that goes after it will also appear on a new line e.g. paragraphs, lists, footers.
- Inline elements are those that are contained within block-level elements and surround only small parts of the document’s content, not entire paragraphs and groupings of content e.g. links, em, strong
- Exercise: https://codepen.io/pen/

Empty elements

- An empty element is an element that cannot have any child nodes (nested elements or text nodes)
- <img src=”http://….”>
- <meta charset="utf-8">
- <link `rel=`"stylesheet" href="my_css.css">
- <br>

Attributes

```
<p class=”summary”>My <strong>first</strong> paragraph</p>
```

- A space between attribute and the element name (or the previous attribute, if the element already has one or more attributes.)
- The attribute name, followed by an equals sign.
- An attribute value, with opening and closing quote marks wrapped around it.

Boolean Attributes

Boolean attributes can only have one value, which is generally the same as the attribute name eg.

```
<input type="text" disabled="disabled">


<input type="text" disabled>


<input type="text">
```

- Omitting quotes around attribute values
- Single or double quotes?

HTML Document

to give text structure and meaning

```
<!DOCTYPE html>
<html>
    <head>
       <meta charset="utf-8">
       <title>My first page</title>
     </head>
     <body>
       <p>This is my first page</p>
    </body>
</html>
```

HTML Document Structure

<!DOCTYPE html>: The doctype

Compliance with XHTML rules

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

<html></html>

wraps whole HTML page, known as root element

<head></head>

a container for all information that is not content and is not displayed, for example keywords, page description, CSS to style the content, etc

Meta tags

Metadata is data that describes data and in HTML metadata is added to the document in the <meta> element. For example: character encoding, author and description, Open Graph Data.

<meta charset="utf-8">

sets the character set your document should use to UTF-8, which includes most characters from the vast majority of human written languages, ISO-8859-1 – latin alphabet

<meta name="author" content="Agata">

<meta name="description" content="The first page containing the basic HTML tags.">

- eg. Sussex Wildlife Trust

Other meta tags : The Open Graph Data, deprecated: keywords

More HTML elements

<title></title>

Adding custom icons

```
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
https://www.favicon.cc/
```

Applying CSS and JavaScript to HTML

<link> element and the <script> element

```
<link rel="stylesheet" href="my_css.css">
<script src="my_js.js"></script>
```

Setting the primary language of the document

<html lang="en-GB">, <html lang="en-US"> <html lang="de-DE">

https://www.w3schools.com/tags/ref_language_codes.asp

<body></body>

The Basics of HTML

- Headings and Paragraphs
  - <p>, <h1> ...<h6> structural hierarchy
- Why is this important
  - readers scan headings first
  - search engines index headings for keywords
  - screen readers often provide outline of the page by reading headings
  - to style content with CSS or make elements accessible to JavaScript you need to markup the content
- Consider these :

```
<h1>This is a top level heading</h1>

<span style="font-size: 32px; margin: 21px 0;">Is this a top level heading?
</span>
```

Basics of HTML

- Emphasis and importance
  - in written language we tend to stress words by putting them in italics : <em> (recognised by screen readers)
  - <i> or <span> with styling
- Strong importance
  - <strong>
- Italic, bold, underline… <i><b><u>

  not to be used for presentation, but ok if convey meaning for example: italic for foreign words, taxonomic designation, technical terms, bold for key words, product names, lead sentence and underline for proper name, misspelling etc.

Exercise

1. Copy the HTML page example listed earlier into a text editor and save the new file as index.html. Or copy it from:

`https://github.com/agatacruickshank/learning_html`

2. Just below the opening tag of the <body> element, add a main title for the document. This should be wrapped inside an <h1> opening tag and </h1> closing tag.

3. Edit the paragraph content to include some text about something you are interested in.

4. Make any important words stand out in bold by wrapping them inside a <strong> opening tag and </strong> closing tag

5. Add an image to your document, below the paragraph. Tip:

   <img src=”http://….”>

Comments

To make a comment the comment text needs to be wrapped in the special markers <!-- and →

```
<p>I'm not inside a comment</p>
<!-- <p>I am!</p> -->
```

Lists

- Unordered list
  - <ul> element wraps around the list
  - <li> (list item) element around each item
- Ordered list
  - <ol> element wraps around the list
  - <li> (list item) element around each item
- Description list, with terms and descriptions

  <dl><dt><dd>
- Nesting lists

Exercise

- Create a shopping list
- Create a detailed directions of how to get from the campus to somewhere else.
- Create a list of chapters in a book with sub-chapters.

Hyperlinks

- Allow linking documents to any other document (or other resource)
- also linking to specific parts of documents
- A basic link is created by wrapping the text with <a> element, and giving it an href attribute
  - <a href="https://digitalcareerinstitute.org/en/">DCI home page</a>
  - title attribute - it will come up as a tooltip when the link is hovered over
  - target attribute - specifies where to display the linked URL
  - _self, _blank, _parent, _top

Hyperlinks

- Block level links

```
<a href="https://www.mozilla.org/en-US/">

<img src="mozilla-image.png" alt="mozilla logo that
links to the mozilla homepage">

</a>
```

URLs and paths

- A URL, or Uniform Resource Locator is simply a string of text that defines where something is located on the Web
- URLs use paths to find files. Paths specify where in the filesystem the file you are interested in is located.
- The root of this directory structure is called creating-hyperlinks. When working locally with a web site, you will have one directory that the whole site goes inside. Inside the root, we have an index.html file and a contacts.html. In a real website, index.html would be our home page or landing page (a web page that serves as the entry point for a website or a particular section of a website.).

Hyperlinks

- Same directory
- Moving down into subdirectories
- Moving back up into parent directories
- document’s section
  - <a href="contacts.html#section">mailing address</a>.</p>
  - <h2 id="section">Mailing address</h2>
  - <p>The <a href="#section">company mailing address</a> can be found at the bottom of this page.</p>

Absolute and relative links

- absolute URL
  - http://www.example.com/projects/index.html Or http://www.example.com/projects/
- relative URL (to the document in which link originates)
  - http://www.example.com/projects/pdfs/example.pdf Can be replaced by pdfs/example.pdf in any document in projects directory

Hyperlinks

Clear link wording is a good practice (avoid “click here”)

- Accessibility: screen readers jump from link to link, search engines use links text, readers often skim the page
- Don't repeat the URL as part of the link text
- Don't say "link" or "links to" in the link text
- Keep your link label as short as possible
- Minimize instances where multiple copies of the same text are linked to different places

Hyperlinks

- Use relative links wherever possible

  it is a lot easier to read the code

  it is more efficient to use relative URLs especially when changing site structure or moving to a different server/domain

- Linking to non-HTML resources

  make it clear – pdf, video, audio

- Use the download attribute when linking to a download

```
<a href="https://download.mozilla.org/?product=firefox-
39.0-SSL&os=win&lang=en-US" download="firefox-39-
installer.exe">
    Download Firefox 39 for Windows
</a>
```

Exercise

- Create a home page with navigation menu linking to home, projects, pictures, contacts

E-mail links

```
<a href="mailto:agata@dci.org">Send email to Agata</a>
```

the email address is even optional, if you leave it out (a href="mailto:") a new outgoing email window will be opened by the user's mail client

- Other information can be provided.

```
<a href="mailto:agata@dci.org?
cc=name2@dci.org&bcc=name3@dci.org&subject=The%20subject
%20of%20the%20email&body=The%20body%20of%20the
%20email">Send mail with cc, bcc, subject and body</a>
```

The values of each field must be URL-encoded, that is with non-printing characters (invisible characters like tabs, carriage returns, and page breaks) and spaces percent-escaped. Use a question mark (?) to separate the main URL from the field values, and ampersands (&) to separate each field in the mailto: URL

Quotations

- Blockquotes for block level content quoted
  - <blockquote cite=”URL”>…..</blockquote>
- Inline quotations
  - <q cite=”URL”>…..</q>
- Citations
  - Cite attribute is ignored by browsers and screenreaders. Use <cite>element next to (or inside) the quote element, it is also advisable to make it into a link to the source of the quotation
  - <cite></cite>

```
<blockquote cite="https://">
  <p>……..</p><cite>citation origin</cite>
</blockquote>
```

Exercise

- Find text on any website and quote it on your page giving the source of quotation -
- Find a short statement and quote it on your page
- You can use http://www.brainyquote.com/ for passages to quote

Abbreviations

```
<abbr title="Hypertext Markup
Language">HTML</abbr>
```

Exercise

Put the following paragraph in your page:

DCI operates a training program for refugees and those who have the right to stay, who are interested in a technical qualification for the German job market.

Contact details

<address> element is meant for marking up the contact details of the person who wrote the HTML document

```
<address>
  <p>Page written by <a
href="../authors/author-name/">name of
the author</a>.</p>
</address>
```

Superscript and subscript

Items like dates, chemical formulae, and mathematical equations require superscript and subscript.

<sup> and <sub> elements

Exercise

My birthday is on the 1st of May 2000.

Chemical formulae for sugar is $C_6H_{12}O_6$.

3^2 is 9, and 4^2 is 16.

Representing computer code

<code>: For marking up generic pieces of computer code.

<pre>: For retaining whitespace (generally code blocks) — if you use indentation or excess whitespace inside your text, browsers will ignore it and you will not see it on your rendered page. If you wrap the text in <pre></pre> tags however, your whitespace will be rendered identically to how you see it in your text editor.

<var>: For specifically marking up variable names.

<kbd>: For marking up keyboard (and other types of) input entered into the computer.

<samp>: For marking up the output of a computer program.

Times and dates

Standard simple date

```
<time datetime="2019-01-01">1
January 2019</time>
```

Entities references

Whitespace in HTML

```
<p>Dogs are silly.</p>

<p>Dogs        are

      silly.</p>
```

In HTML, the characters <, >,",' and & are special characters. To represent them in the web page use equivalent character references. Each character reference is started with an ampersand (&), and ended by a semicolon (;)

Entities references

Literal character	Character reference equivalent
<	&lt;
>	&gt;
"	&quot;
'	&apos;
&	&amp;

Entities

```
<p>In HTML, you define a paragraph using the <p>
element.</p>
```

```
<p>In HTML, you define a paragraph using the &lt;p&gt;
element.</p>
```

Basic sections of a document

header

navigation bar

main content

sidebar

footer

HTML page structure

- header: <header>.
- navigation bar: <nav>.
- main content: <main>, with various content subsections represented by <article>, <section>, and <div> elements.
- sidebar: <aside>; often placed inside <main>.
- footer: <footer>

Colorblind people represent around 8% of the world population

Assistive technologies like screenreaders can recognise those elements

Non-semantic wrappers

Sometimes an ideal semantic element cannot be found.

<div>is a block level non-semantic element, which you should only use if you can't think of a better semantic block element to use, or don't want to add any specific meaning.

<span> is an inline non-semantic element, which you should only use if you can't think of a better semantic text element to wrap your content, or don't want to add any specific meaning.

Validation

- The W3C has built both HTML and CSS validators that will scan code for mistakes
- http://validator.w3.org/
- http://jigsaw.w3.org/css-validator/

Sources

- https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals
- https://developer.mozilla.org/en-US/docs/Learn/Common_questions/
- https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started