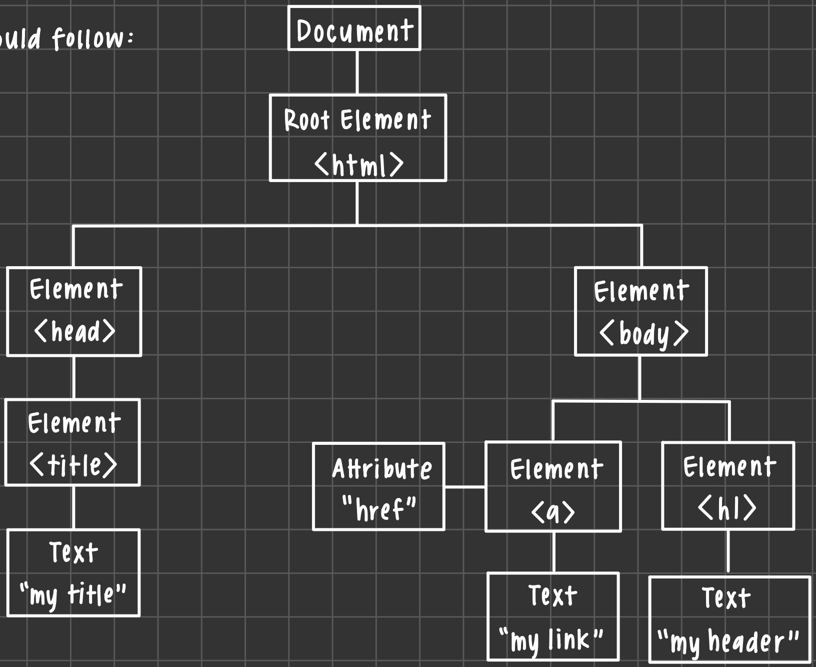
The Document Object Model (DOM)

DOM

clean code = code that works on as many devices as possible

- HTML5 principle : new features should be based on: HTML
  CSS
  DOM
  JavaScript

- DOM provides common tree-like structure all pages should follow:



- 3 parts of a well-formed document: doctype
  head
  body

- doctype = version of HTML you'll be using

  → (eg. HTML5): <!DOCTYPE html>

- head : hold additional info used by the browser
  only title is displayed

  - meta data : language
    title

  - supporting files : JavaScript
    Styling/css
    Add-ons

- body : displayable content (minus any possible meta-data)
  bulk of your page

- validate code!!!! → (validator.w3.org)

HTML5 Elements - Tags, Syntax, Semantics, & Templates

HTML5 Tags & Syntax

- tags have a beginning & end: start tag & ending tag
  self-closing tag

- display attribute

  - block: newline is inserted before & after (eg. takes up whole width)
    can take width & height

  - inline: only uses as much space as it needed to contain the element
    can't take width & height

- common tags

  → (eg. <h1> - <h6>): headings (block)
  have syntax & semantics

  → (eg. <p>): paragraphs (block)
  should only contain inline elements

  → (eg. <div>): divs (block)
  generic section that's larger than a paragraph

  → (eg. <ol>): ordered lists
  marked w/ numbers

  → (eg. <ul>): unordered lists
  marked w/ symbol

  → (eg.
): line breaks

- attributes = provide additional info about an element
  always specified in the start tag
  come in name/value pairs

  - images:

<img src = "logo.jpg"	image filename
alt = "company logo"	info for screenreaders, broken links
title = "AAA1 LLC"	displays over hover
class = "thumbnail/>	extra formatting (height, width, position, etc)

  → (eg. class): applies special properties to groups of elements

  → (eg. id): specifies a unique id to one element on the page

  → (eg. accesskey): a shortcut key to activate an element

→ (eg. tabindex): the order elements will come into focus using the tab key
lower tab·index # = higher priority

- special entities: & lt; → <
  > → >
  & copy; → ©
  & cent; → ¢
  & amp; → &

- ultimate HTML5 Cheatsheet : wpkube.com/html5-cheat-sheet/

Semantic Tags

- semantic tags help guide users to info in your page → make page most accessible as possible
- <header> = a group of introductory or navigational aids
  won't format position to top of the page
  block tag

- <nav> = a section of the page that links to other pages or to parts w/in the page
  often found in the header tag

- <footer> = a section that contains info such as : copyright
  data
  related documents
  links to social media
  typically at the bottom, but not required

- <figure>: more semantics > <img>
  can include : caption (<figcaption>)
  multiple multi·media resources

- other new tags

  - structural elements: article
    aside
    main
    menuitem
    summary
    section

  - media elements: audio
    embed
    source
    track
    video

  - form elements: datalist
    keygen
    output

  - graphic elements: canvas
    svg

  - input types: color
    date
    email
    list

Images

Images

- every image · must be downloaded
  requires a HTTP request

- linked image size displayed on browser = size of file it's been saved as → not optimal

  - format: <img src = "img path/img file name" → <img src = "imgs/Cypress.jpg"] default image size
    alt = "alt text"> alt = "My house">

- to change image size : change file]- quick solutions
  use attributes —— most common practice

- change file : use an editor (eg. Preview)
  only works on local files
  can't change size of an image you're linking to on the web

- use attributes : via width & height attributes in <img> tag

  → (eg. using width in pixels) : hardcoded = set image size
  browser will configure height to be proportionate to width

  - format: <img src="source" → <img src = "imgs/Cypress.jpg"
    width="#px" alt="alt text"> width = "500px" alt="My house">

  → (eg. using width & height in pixels): set, but skewed image size

  - format: <img src = "source" → <img src = "imgs/Cypress.jpg"
    width="#px" height="#px" width="500px" height="100 px"
    alt="alt text"> alt="My house">

  → (eg. using percentages): relative image size
  resizes img size by #% of container size (ie. grows & shrinks w/ browser window)

  - format: <img src="source" → <img src = "imgs/Cypress.jpg"
    width = "#%" alt="alt text"> width = "50%" alt="My house">

- favicon = image/logo/icon in title bar in the tab
  must go in <head> section

  - format: <link rel="icon" type="image/png" href="img path/img name">

- alternative text = text alternative to non-text content
  read by screen readers
  displayed in place of images
  provides semantic meaning for search engines

→ (eg. alt="") : empty alt text > no alt attribute
for decorative images w/ non-informative purpose

→ (eg. alt="link") : replace alt text w/ link to separate page w/ full explanation
for images that require elaborate alt text (eg. infographics)

- creating good alt text: be accurate
be succinct/concise
don't be redundant
don't include "picture/graphic of"

Hyperlinks

Hyperlinks

- <a> : anchor link
  needs: hyper-reference & content

- hyper-reference (href) = reference to location of new content
- content = the clickable part
  text or image

- types of links: absolute
  relative
  internal / embedded
  graphical

- absolute reference: fully formed url (protocol + host + document)
  use when linking to somebody else's page

  - format: <a href="url"> content</a>

- relative reference: file name
  - should NEVER include anything specific to your computer/local to your machine

  use when developing your own site

  - format: <a href="file name"> content</a>

  ⟶ (eg. href= "page2.html"): link to a local file in the same folder

  ⟶ (eg. href="docs/page2.html): link to a local file in a different folder called docs

- internal reference: id attribute name
  use to jump to specific location in the same file

  - format: <a href=#id name> content</a>

- using images as links

  - format:
```
<a href="href">
      <img src="src"
           alt="alt text" />
   </a>
```

- target attribute = describes where links will be loaded
  w/in the anchor tag

  ⟶ (eg. _self): opens in the same frame
  default action

→ (eg. _blank): opens in a new tab or window

→ (eg. _top): opens in the full body of the window

→ (eg. _parent): opens in the parent frame

Multimedia

Multimedia

- adding video & audio materials
- HTML5 was designed to avoid use of extra software (eg. plugins) to play multimedia

- <video> : uses src attribute or embedded <source>
  - format: <video src="source" optional attributes > text displayed if browser can't support tag </video>
  - common attributes: width, height
    autoplay
    loop — boolean attributes, no value assigned
    controls

- <audio> : uses a src attribute to link to audio file
  typically .mp3 or .wav
  - format: <audio src="source" optional attributes> text if audio isn't supported </audio>
  - common attributes: autoplay, controls, loop
    buffered
    muted
    volume

- setting clips : set multimedia to play/end at a specific timestamp by adding onto the src attribute
  - format: src = "source#t=start,end"
    - ⟶ (eg. ...#t=5,35"): plays from 5 s - 35 s
    - ⟶ (eg. ...t=,39"): plays from start-39s (first 39s)
    - ⟶ (eg. t=42"): play from 42s-end
    - ⟶ (eg. t= 15:38"): play from 15m 38s-end

- provide links to plugins for accessibility issues

Tables

Tables

- to display data, not for layout
- sketch/design table beforehand → # of rows & columns?
  will rows or columns span multiple cells?

- tags

  → (eg.<table>): container tag

  → (eg.<tr>. . </tr>): rows

  → (eg.<td>...</td>): columns

  → (eg.<th>...</th>): table heading

  - format:

```
<table>
    <tr><th>heading1</th><th>heading 2</th></tr>
    <tr><td> r1c1 </td><td> r1 c2 </td></tr>
    <tr><td> r2c1 </td><td> r2c2 </td></tr>
    <tr><td> r3c1</td><td> r3c2 </td></tr>
<table>
```

heading 1	heading 2
r1c1	r1c2
r2c1	r2c2
r3c1	r3c2

  → (eg. <caption>): link text to table (w/ semantic meaning)

- attributes

  → (eg. rowspan & colspan): to combine multiple rows and/or columns (aka spanning multiple cells)

  → (eg. border): adds border to table & lines in between cells
  *avoid styling in HTML, but this is commonly used

  - format:

```
<table border="1">
    <tr>
        <th> Parent's Name:  </th><th> Child Name(s): </th>
    </tr>
    <tr><td> Colleen </td> </tr>
    <tr>
        <td rowspan="2"> Catherine </td><td> Michael</td>
    </tr>
    <tr><td> Maggie</td><td> Sheila </td> </tr>
    <tr> <td> Jeff</td></tr>
    <tr>
        <td> Edward </td><td> Cooper </td>
    </tr>
</table>
```

Parent's...	Child...
Colleen	Catherine
	Michael
Maggie	Sheila
Jeff	Edward
	Cooper

Useful Tags

Useful Tags

- generic : <p>

  <div>

- semantic : <header>

  <footer>

  <nav>

  <figure>

- block tags : <article>

  <aside> containers

  <section>

  <main>

  <hr> = horizontal rule (line)

  <address> . allows screen readers to easily find your address

  <blockquote> : has cite attribute

  <details> w/ <summary> : drop down open/closed text (▸closed ▾open)

  - not supported in Firefox

- inline tags : <span> : og inline tag for plain text <a>

  <cite> : makes it easier for ppl & bots to search code for specific information

  <abbr> : defining your abbreviations

  <time> : specifying the use of time

  <code> : identifying code fragments, displayed in different font usually

  <sub> & <sup> = subscript & superscript

- tags that need "more" : should be incorporated w/ Javascript, etc

  <button>

  <meter>

  <progress>

  <iframe> : nest content w/in each other, often used to embed documents

  <bdo> = bidirectional override, defines direction of the text, dir attribute (ltr or rtl)

  <map> w/ <area> : creates clickable element in image

- tag problems? : validate code

  run code in multiple browsers

[Q. the alt text of an image should not describe the appearance of an image]