A. HTML
    a. https://html.spec.whatwg.org/
    b. Part of SGML (standard generalized markup language)
    c. Follows DTD format
        i. DTD = Document Type Definition
        ii. set of markup declarations that define a document type,
            1. Ensures that document renders in standards mode
            2. Defines what's allowed in your document and what's not
    d. Not context-free = cannot be quantified with a series of recursive rules for describing.
        i. Allows omitting some start and ending tags (browser added)
        ii. Soft syntax
        iii. Popular, easy to write
    e. HTML = publishing language for web
        i. Tim Berners-Lee invents the Web
        ii. CERN, the European Laboratory for Particle Physics in Geneva, Switzerland
        iii. Way for researchers to organize/pool data
        iv. Hypertext: link files/text from one to another (cross-referencing)
    f. HTML5
        i. Most recent version (2008)
        ii. Added several semantic tags
        iii. Video + audio
        iv. Vector graphics (svg, canvas)
        v. Web workers (js running in background)
    g. Purpose: structural meaning to web content
B. Viewing HTML in browser
    a. Demo in Chrome
        i. Use browser comfortable and familiar to you
        ii. Things may look different
    b. Open file in browser
        i. Menu command: File > Open File
        ii. Keyboard command: Cmd/Ctrl O
        iii. Navigate to file location
C. Semantic HTML
    a. https://internetingishard.com/html-and-css/semantic-html/
    b. Use of HTML markup to express meaning of the information instead of defining presentation
    c. Separation of concerns
        i. HTML: markup, content structure
        ii. CSS: presentation, appearance
    d. Example:
        i. <H1> vs big & bold
        ii. vs italic
    e. Important to make structure semantic
        i. Maintainability: helps you as a developer keep your site organized
        ii. Accessibility
            1. Every HTML document has an "outline," which is how search engines and screen readers view the hierarchy of the content on the page

2. Outline helps adapt the way they present information to the users according to the structure of the document
3. The more semantic the markup, the easier it is for search engines, screen readers, and other machines to identify the different parts of your website.

iii. Picture a series of boxes tucked away in an attic
1. None of the boxes are labeled
2. How do we know how to organize whatever is inside the boxes when we visit the attic?
3. Semantic HTML = giving the boxes relevant labels to give structure/meaning to whoever has to view the content later
   a. Browser
   b. Web crawler/robot
   c. Code maintainers

f. Elements reference: https://developer.mozilla.org/en-US/docs/Web/HTML/Element

D. What does HTML look like?

a. Tag = some keyword between < > brackets
   i. <tag>
   ii.
   iii.
   iv. <img>

b. Open tag + (USUALLY) closing tag
   i. <tag></tag>
   ii.
   iii. <img /> ← does not take a closing tag, the /> is the closure
      1. Closing tags enclose content encompassed by the tag
      2. Foo
      3. Bar
      4. <img /> ← images don't have content to enclose
   iv. Some tags don't need to be closed
      1. "Self closing tags"
      2. Closing tags are optional because it's implied that a new tag would not be able to be started without closing it
      3. `html, head, body, p, dt, dd, li, option, thead, th, tbody, tr, td, tfoot, colgroup`
      4. Tags that never take an explicit close: `img, input, br, hr, meta`
      5. If unsure, use the HTML validator: https://validator.w3.org/
   v. HTML can be nested
      1. Inline elements are nested within block elements
         a. Eg This is strong while this is not
      2. Block elements can be nested within block elements
         a. Eg <section> This is a paragraph in a section </section>
      3. Inline elements can be nested within inline elements, in some cases
         a. Eg This is both strong and emphasized
      4. Nesting must be closed from inside out, like parentheses
         a. Cannot cross tags while closing nesting.
         b. Eg This is both strong and emphasized is incorrect!

E. <!DOCTYPE html>
    a. Not actually an element or HTML tag itself
    b. Every HTML5 document (ie, all new web documents) should begin w/ DOCTYPE declaration to be compliant with HTML standards
        i. First element in the document
        ii. no closing tag.
    c. Informs the website visitor's browser that the document being rendered
        i. is an HTML document
        ii. how the document should be interpreted, by indicating what version or standard of HTML is being used
            1. Prevents the browser from switching into “quirks mode” when rendering
            2. ensures that the browser tries to follow specifications, rather than using a rendering mode that is incompatible with some specifications
    d. 3 layout engine modes
        i. Full standards mode: layout behavior is behavior described by HTML5 + CSS specs
            1. Simplest doctype <!doctype html>
            2. All existing browsers will interpret as full standards mode, attempt to render against that spec
        ii. Almost standard mode: layout behavior is close to specs with some deviations
        iii. Quirks mode: layout behavior is nonstandard behavior from Navigator 4 and Internet Explorer 5
            1. Support websites that were built before the widespread adoption of web standards (corp. Intranets, eg.)
F. HTML tag
    a. the root (top-level element) of an HTML document
    b. Also called “the root element”
    c. All elements must be descendants of this element, except doctype
    d. Lang
        i. Define the language of an element
            1. Uneditable elements = language written in
            2. Editable elements = language user should use
        ii. Could tag/define every single element of a page as different language
        iii. In general, defined on HTML tag
    e. Dir
        i. Directionality of language
            1. Ltr = left to right, eg. English, Spanish
            2. Rtl = right to left, eg. Hebrew, Arabic
            3. Auto = let browser decide
        ii. Can be overridden by css
            1. Recommended use HTML attributes if CSS not supported for some reason
G. Head tag
    a. provides general information about the document
        i. Title
        ii. Metadata
        iii. links to scripts and style sheets
    b. Meta tag
        i. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta

ii. represents metadata that cannot be represented by other elements
iii. Charset
1. declares the page's character encoding
2. Always specify encoding; needed to process non-ASCII characters entered by the user in forms, in URLs generated by scripts, and so forth.
3. Always use utf-8
a. Unicode-based encoding
b. Supports many languages
c. Wide browser support
d. Wide usage
i. https://w3techs.com/technologies/details/en-utf8/all/all
ii. UTF-8 is used by 93.5% of all the websites whose character encoding we know
4. Right after <head>
5. Equivalent declarations
a. `<meta charset="utf-8">`
b. `<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>`
iv. Name
1. defines the name of a piece of document-level metadata
2. viewport

c. Title
i. Identify page in in browser tab
ii. Example

H. Content sectioning
a. Outliner: https://gsnedders.html5.org/outliner/
b. HTML5 brings precision to how documents are broken into sections using sectioning blocks and headers
i. Allows document outlines to be predictable and used by the browser to improve the user experience
c. Sections
i. https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using_HTML_sections_and_outlines
ii. All content lying inside <body> is part of a section, even if the section is the body itself
iii. sections in HTML5 can be nested
iv. Explicit sections = enclosing content in opening/closing tags like section, article, aside, etc.
1. Example
v. Implicit sections = dividing content with h1-h6 headers
1. Each header causes browser to close previous section and start new
2. Example
vi. To make your markup human-understandable, good practice to use explicit tags for opening and closing sections
vii. Exception: reusable components that may be assembled instead of top to bottom outline
1. H1 for top level
2. Best judgement for next level headers
3. Outline will be generated

d. <Body>
    i. Content section of webpage
    ii. 1 per HTML document
    iii. All visible content in the viewport will be located inside the body
e. <Main>
    i. dominant content of the <body> of a document, portion of a document or application
    ii. Usually defined as being separate from document header/footer
f. <Section>
    i. Generic sectioning block
    ii. explicitly delineate block of website
    iii. A section must have a header to be valid
g. <Article>
    i. self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable
        1. forum post
        2. magazine or newspaper article
        3. blog entry
        4. Twitter post
h. Sectioning blocks that don't get added to the document outline
    i. <Nav>
        1. Indicates a block of navigation links, either within the current document or to other documents
            a. Menus
            b. tables of contents
            c. Indexes
            d. Breadcrumbs
        2. Can be own block or within context of header, footer, etc.
        3. Do not need to mark individual links as nav
            a. The <a> itself is an indicator that it is navigation
    ii. <Aside>
        1. portion of a document whose content is only indirectly related to the document's main content
            a. Sidebars
            b. call-out boxes
        2. Aside does not imply "to the side"! Can be located anywhere within content.
        3. Could be used to markup ad space/promoted content/affiliate info
i. Content dividers
    i. Not sectioning blocks
        1. do not produce new sections in outline
    ii. <Div>
        1. block of content
        2. hook for css
    iii. <P>
        1. paragraph of text
j. <Blockquote>
    i. Long quotation
    ii. Usually rendered indented visually
    iii. Cite attr: provide URL reference to where the quote comes from

I. <A> = creates hyperlink to other reference
    a. Surrounds content to be clicked on
        i. Could be simple text, image, etc.
    b. Examples
        i. Absolute link = Google
        ii. Relative link = Foo
        iii. Mailto: = April's email
    c. Target attribute
        i. _self: load url into current browsing context. (default)
        ii. _blank: load url into new browsing context. (tab or window)
        iii. Other targets
            1. _parent
            2. _top
            3. Pretty much only seen with iframes/frames, hardly ever used anymore

J. Text markup
    a. Block vs. inline elements
        i. Block
            1. block-level elements may contain inline elements or other block-level elements
            2. block elements create "larger" structures than inline elements
        ii. Inline
            1. inline elements may contain only data and other inline elements
            2. can't put block elements inside inline elements
            3. inline elements do not force a new line to begin in the document flow
    b.
        i. Generic inline text container
        ii. Like div, used for css hooks
    c.
        i. Content with strong importance, seriousness, or urgency
        ii. Typically rendered as bold
        iii. Why not use <b>?
            1. Possible to provide emphasis without making something bold
            2. Color, border, etc.
            3. Bold !== strong and vice versa
    d.
        i. Content to be emphasized
        ii. Text that may be italicized in text
        iii. Typically rendered as italic
        iv. Why not use <i>?
            1. Same as strong/bold
            2. Italic !== emphasis
    e. <sub> or <sup>
        i. Subscript
            1. Footnote numbers
            2. Chemical symbols: C_8 H_{10} N_4 O_2
        ii. Superscript
            1. Exponents: a ^ 2 (a^2)
            2. Ordinal numbers: 4^{th}
    f. <abbr title="Northeastern University">NEU</abbr>

i. Abbreviation
ii. Attribute title
1. Instructs browser to give definition inline

g.
i. Line break
ii. Equivalent to carriage return
iii. Break up lines of text that are still related by block

h. <address>
i. contact information associated with the webpage itself
ii. can be used in a variety of contexts
1. providing a business's contact information in the page header
2. indicating the author of an article by including an `<address>` element within the `<article>`

i. <q>
i. Short inline quotation
ii. Browser will render in quotation marks
1. Example: quotations appropriate for language! En vs. fr

j. <s>
i. Strikethrough
ii. Text that is no longer relevant or accurate

k. <time datetime="2018-11-22">November 22, 2018</time>
i. presenting dates and times in a machine readable format
ii. Datetime attribute = needs to be machine readable
1. Valid datetimes: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/time
iii. Date/time enclosed by tags can be human readable

K. Lists
a. 3 types
i. Unordered: lists that do not have inherent order
1.
2. List item: markup list items within ordered and unordered tags
a.
ii. Ordered: lists that represent ordered information
1.
2. Example: series of steps
iii. Definition list: list of related term & definition pairs
1. <dl></dl>
2. <dt>term</dt>
3. <dd>definition</dd>
4. Example: implement a glossary or to display metadata (a list of key-value pairs)

L. H1-H6
a. Outline concept
i. H1 highest level content
b. 1 h1 per page, ideally the most important piece of information
c. Header for block of content
d. Increasing header numbering as traversing outline
i. Start with h1, next most important is h2, etc.
e. Try not to skip headers

i. Start with h1, next level h2, etc.
ii. Use in conjunction with <header> or <hgroup> tags

M. More content blocks
   a. Header
      i. introductory content
      ii. navigational aids
      iii. may contain logo, search form, author name, and so on
      iv. Multiple blocks can have headers
      v. Group headers in <hgroup>
   b. Footer
      i. footer for its nearest sectioning content.
      ii. typically contains information about the author, copyright data, publishing data
   c. <Figure>
      i. self-contained content
      ii. Often with <figcaption>
      iii. typically referenced as a single unit
      iv. image, illustration, diagram, code snippet, etc., that is referenced in the main flow of a document
      v. Can be moved elsewhere without affecting main flow
      vi. <figcaption>
         1. caption or legend for the rest of the contents its parent <figure>

N. Tables
   a. Presentation of tabular data
   b. Data that can be represented as a 2 dimensional display of columns and rows
   c. Tables were used for content layout, should only be used for tabular data now
   d. Caption
      i. caption/title of a table
      ii. Always comes after <table> (first child)
   e. Thead
      i. set of rows defining the head of the columns
   f. Tbody
      i. Set of rows defining the body of the table
   g. Tfoot
      i. set of rows summarizing the columns of the table
      ii. Example: totals in a spreadsheet
   h. Tr
      i. row of cells in a table
      ii. Container for combination of <th> and <td> cells
   i. Th
      i. Header of a group of table cells
   j. Td
      i. cell of a table that contains data
   k. Rowspan/colspan
      i. Allows a single table cell to span the width or height of more than one cell or column
         1. Picture “merge cell” in spreadsheet programs
      ii. Rowspan: Allows a single table cell to span the height of more than one cell or row
      iii. Colspan: Allows a single table cell to span the width of more than one cell or column

iv. might be used for a header cell that titles a group of columns or a side-bar that groups rows of entries.
v. colspan= and rowspan= are attributes <th> and <td>
vi. The value of either attribute must be a positive integer (a whole number)
1. specifies the number of columns or rows that the cell fills

O. Forms
a. document section that contains interactive controls
i. submitting information to a web server
ii. capturing/handling interactive actions in a human-usable way
iii. Action = URI of a program that processes the form information
1. Often not used for React app
a. Variables -> state -> API calls, etc.
2. Would be used to send data to form processing script on a server (ruby, php, perl, etc.)
iv. Method = HTTP verb method of sending data. POST, GET, etc.
v. In modern single-page apps (eg, react), inputs are often used independent of full forms in order to add hooks for interactivity via handlers on elements (buttons and inputs)
b. Fieldset
i. Used to group multiple related fields/controls/labels within a form
ii. Example: first/middle/last name
iii. Browser default is to put border around, can be removed with CSS
c. Legend
i. Caption/title for content of its parent fieldset
d. Label
i. title/caption for interactive form element
ii. Best practice: associate label + input with “for”
1. allows screen readers and other non-visual browsers to make link between label and input
2. allows input to be activated when label is activated, especially on very small inputs (eg, checkbox, radio)
iii. Best practice: 1 label per input
1. can have multiple labels
2. Screen readers can have problems with them
e. Input
i. create interactive controls for web-based forms in order to accept data from the user
ii. Multiple control widgets
1. Listing most common, several more depending on application you need
2. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input
3. Not all supported by all browsers
4. [type=button]: button widget with no default behavior
a. Use for any action where you need to capture javascript action and trigger code
5. [type=text]: default text inputs
6. [type=checkbox]: allows multiple values to be selected for an input response
a. Value: not seen in the browser, corresponds to the value to be given to as the input response
i. If value is omitted, default value is “on”
b. Multiple checkboxes

i. Server will receive values separated by “&”

7. [type=radio]: allows only 1 value to be selected for an input response
8. [type=submit]: button widget with default behavior of submitting the form
9. [type=password]: password inputs
10. [type=number]: number inputs
    a. Mobile browsers will launch number-only keypad
    b. Browser provides automatic validation entered text is a number
    c. set of up and down buttons to step the value up and down
11. [type=tel]: telephone inputs
    a. Mobile browsers will launch telephone keypad
    b. makes adding custom validation and handling of phone numbers more convenient
    c. the input value is not automatically validated to a particular format

iii. Disabled
1. State where user cannot interact with the control
    a. Not clickable
    b. User cannot activate/input value

iv. Readonly
1. User cannot modify the value of the input
2. Different than disabled: user can still click on/interact with control

v. Required
1. Indicates form is invalid if left empty (will not submit)

f. Select
i. control that provides a menu of options
ii. Multiple: allows multiple options to be selected by cmd/ctrl clicking

g. Optgroup
i. Group options in a select
ii. Label displayed is not selectable

h. Option
i. Defines items contained in a select or optgroup
ii. Value: value to be sent to the form
iii. Selected: indicates default selected option
1. If none specified, defaults to first in the options list
2. If multiple specified, multiple can be selected

P. Button
a. Clickable button element
b. Can be used either inside or outside forms
c. Presented as same style as OS button by default with no styling
d. Value: initial value of the button.
e. Button content: enter between tags

Q. Image
a. Embeds image into a document
b. Src attribute = path for images
i. Relative
ii. Absolute
c. Width & height attributes inherent on tag to set layout for stable page layout
i. CSS for width & height
d. Each browser supports different set of image formats

    i. Eg Firefox: JPEG; GIF, including animated GIFs; PNG; APNG; SVG; BMP; BMP ICO; PNG ICO
e. Alt attribute: alternate text displayed
    i. Eg while loading, loading error