Webpage

HTML: Hypertext Markup Language
not case-sensitive，recommend use lowercase.

        - Semantic Container Element
    - Element-permitted-content

```
// html structure
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <!--other elements goes here-->
</body>
</html>
```

Semantics

refers to meaning of a piece of code.

Every HTML element has their own roles

- a element: hyperlink
- p element: paragraph
- h element: heading level

```
<!--By default, h1 is bold, and has bigger font size-->
<h1> Heading </h1>
<!--By default, p has margin etc-->
<p> Paragraph </p>
```

These styles are ruled by CSS (user agent/ browser style sheet) and can be overriden.

Importance: Choosing an element is highly dependent on the context and semantic meaning and has nothing to do with their default styles.

Why semantics?

1. Search Engine Optimisation (SEO): the process of improving your website to increase its visibility in search engine.

- semantic HTML provides explicit meaning to the content structure, making it easier for search engine to understand and index.
- SE: Google, Bing, ...
- SE will retrieve HTML source code from the Internet and store them in database every internal time very frequently.

2. Accessibility: practice of making your websites usable by as many people as possible.

- browser reading mode - only show content for better experience
- audio mode

Elements

> other name: tags

Element = Opening Tag + Closing Tag + Content + Attributes

Attribute = Name + Value

Attribute Types:

- Global: common to all elements
- Local: only available to certain elements

```
// body element
<body></body>

// p element
<p id="text">Hello</p>

// a element
<a href="https://www.google.com">google</a>
```

Empty tags

> a.k.a void tags, singletons tags

Some elements do not have closing tag.

No content

No children

```
// two ways of writing
<meta charset="UTF-8"> // HTML5 allows this.
<meta charset="UTF-8"/> // old way

<img src="" alt="">
```

Nesting

Elements inside elements.

Ancestor
e.g., html is the ancestor of meta element.

Descendant
e.g., p elememnt is the descendant of body element

Parent
e.g., body is the parent of div.

Sibling
e.g.1, meta element and title element are sibling.
e.g.2, title element are p element are not sibling.

Child
li element is the child of ul/ ol element

```
// Nesting
<div>
  <p></p>
</div>

// Invalid!
<div>
  <p>
</div>
  </p>
```

Standard HTML Structure

Web Page: Page, HTML document

```
<!DOCTYPE html>
```

Document Type Declaration, tell browsers, the current Doc use HTML5 standard.

No writing this results in the browser rendering in quirks mode

HTML element

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

Root element, one page can only has one and it serves as ancestor/ parent to other elements.

Writing root element is not necessary in HTML5 but recommended.

lang attribute: language, global attribute, determine the language of the text used on any webpage. e.g., en (English) / cmn-hans (Chinese)

Head element

```
<head>

</head>
```

Document head, contains machine-readable information (metadata) about the document, not displaying on webpage.

Meta element

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

charset attribute: character encoding in which the document is encoded.

In PC, low voltage (0~2V) 0, high voltage (2~5 V) 1, represent 2 in binary system (10)

3 = 11
4 = 100

In PC, store only numbers - img, video can't.

text? map text with number

e.g., a -- 97, A -- 65

this dictionary called Character encoding system. e.g., UTF-8, GB2312, GBK, etc to parse characters of a webpage.

e.g., Simplified Chinese and Traditional Chinese Encoding

> 袁 -- GB2312 -- 100000 -- GBK -- ?

100000 might be different words in GBK, resulting in scrambled characters

UTF-8: one of the version/ variant of unicode, recognise every characters

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
// for mobile device
```

```
<meta http-equiv="X-UA-Compatible" content="ie=edge">
// force Internet Explorer (IE) to use the latest version of the rendering engine
```

```
<title>Document</title>
// defines the document's title that is shown in a browser's title bar or a page's tab.
// text-only
```

Body element

document body, content to be displayed on the webpage.

Text element

h (h1 ~ h6) : represents six level of section heading.

Note: use h1 for logo - use css to hide - for SEO.
p : represents paragraph

Span element

span has no semantic meaning and is only used for styling.

```
<p> This is <span style="color: red;"></p>
```

Note:

Some elements only occupy one line, some won't.

Previously, they are referred to as block-level element and inline element.

HTML5 doesn't use them anymore.

Block and inline both refers to display layout, which is governed by CSS.

Pre element

Preformatted Text Element (Pre) represents text displayed exactly as written in the HTML file.

no white space collapsed.

due to css property "white-space: pre;"

```
<!--Generally use code to wrap code
we can set css property of code so it behaves as pre.
-->
<code>
  <pre>
    aaa  aaa
      bb
    ccc  ccc
  </pre>
</code>
```

Anchor element

html is special because of a element hyperlink

js can also achieve hyperlink

```
<a href="address">any text</a>
<a href="https://www.google.com/">Google</a>
```

href (hypertext reference) attribute

- absolute URL - points to another web site.
- relative URL - points to a file within a web site.
- anchors (#) - different positions on the current page (no refresh if it is on the same page).
- functional links *
  - run JavaScript
  - send email - require user pc has email software installed.
  - phone call - require mobile/ pc requires dialing software.
  - note: both redirect address and anchor are same.
  - change the address bar
  - both refresh the page (anchor also refresh page if it involves different pages)

```
// how clicking on anchor changes the url
http://127.0.0.1:5500/04-a.html#chapter4

// run JavaScript
<a href="javascript: alert('Hello')">Hello</a>
```

id attribute

- global attribute - means common to all html elements.
- must be unique in the whole document.

target attribute

- where to display the linked URL
- _self: current browsing context
- _blank: new tab/ new window
- etc.

title attribute

- global attribute
- represent advisory information related to the element it belongs to (tip).

Emmett

```
((h2[id="chapter$"]>{Chapter $})+p>lorem10000)*6
```

The Anchor element

Path

1. Absolute Path: for off-site/ external resources.

url addresses

```
schema://host:port/path
https://www.google.com/path
http://renren.com/path
file:///C:/path

// schema identifies the protocol to be used to access the resource on the Internet (transport)
// host identifies the host that holds the resource. (street)
// path identifies the specific resource in the host that the web client wants to access. (sect
// some parts like schema and port can be omitted in certain cases.
```

The components of a URL

2. Relative Path / Absolute Path: for site resources.

Start with ./ (represents current directory)

../ (represents parent directory)

/ (represents root directory that serves as the starting point for organizing all the files and folder)

```
./subhtml/a.html
```

```
../subhtml/a.html
```

Img element

img shorts for image, empty element.

src attribute

source

```
<img src="https://duyihome.oss-cn-beijing.aliyuncs.com/images/home/lbt/2.png">
```

alt attribute

When image is not displaying

```
<img src="https://duyihome.oss-cn-beijing.aliyuncs.com/images/home/lbt/2.png" alt='this is a pi
```

example

work with a element

```
<a href="">
  <img src="" alt="">
</a>
```

work with map element

map element: define an image map (a clickable link area).
area element (child element) defines an area inside an image map that has predefined clickable areas.

attributes:

shape:

1. circle
2. rect
3. poly

coords:

1. circle: center x, center y, radius.
2. rect: top_left x, top_left y, bottom_right x, bottom_right y.
3. poly: x1, y1, x2, y2, x3, y3, x4, y4,...xn, yn of each vertex.

use professional screenshot software/ photoshop to get coordinates to prevent deviation.

```
<map name="solarMap">
  <area shape="circle" coords="824,352,86" href="https://en.wikipedia.org/wiki/Jupiter" alt="
  <area shape="rect" coords="793,456,866,470" href="https://en.wikipedia.org/wiki/Jupiter" al
  <area shape="poly" coords="" href="https://en.wikipedia.org/wiki/Pluto" alt="">
</map>
```

work with figure element

figure element: represents self-contained content for an image, illustration, diagram, code snippet, etc.

figcaption element (child element): represents a caption or legend providing figure element an accessible description.

Media element

both video and audio share similar attributes.
boolean attributes:
controls: show control
autoplay: autoplay video
muted: play with muted video (no audio play)
loop: play again when finished

compatibility:

1. older browsers may not support.
2. different browsers support different audio/ video format.

Video element

embeds a media player which supports video playback into the document.

Note:

Some attributes can only have two states 1. not write 2. value set to attribute name. They are called boolean attribute

```
<video controls="controls" src=""></video>
<video controls src=""></video>
<!--For better compatibilities-->
<video>
  <!--If we have different video with different formats-->
  <source src="">
  <source src="">
  <!--If source is also not supported in user browser, use text/ flash to embed video/audio-->
  <p>Sorry your browser cannot support video element, please download the latest browser ...</p
</video>
```

Audio element

embed sound content in documents.

Note: Initially, it is invisible, set controls to the audio element to make it visible.

List element

li (list item): child element
represent an item in a list.

Ordered list (ol)

represents an ordered list of items — typically rendered as a numbered list.

attributes:

1. type:

- default - numbers
- i - Roman numerals
- a - Lower-cased alphabets
- A - Capital alphabets

2. reversed (boolean)

Note: Encouraged using css list-style-type to set the style unless for legal documents.

Unordered list (ul)

represents an unordered list of items, typically rendered as a bulleted list common usage: menu

Description list (dl)

represents a description list of description terms (dt) and description details (dd).

common usage: literature

child:

1. dt - description term
2. dd - description details

```
<dl>
  <dt>HTML</dt>
  <dd>Hypertext Markup Language</dd>
</dl>
```

Container Element (div)

non-semantic. No style.

Use CSS for display and effect.

represents the generic container for flow content.

use to put other elements inside for layout.

Browser couldn't recognise its meaning.

Semantic Container Element

1. **header** : represents introductory content, may contain some heading elements but also a logo, a search form, an author name, and other elements.
2. **footer**: represents a footer for its nearest ancestor sectioning content or sectioning root element, contains information about the author of the section, copyright data or links to related documents.
3. **article**: represents a self-contained composition in a document, page, application, or site. (whole article)
4. **section**: represents a generic standalone section of a document, which doesn't have a more specific semantic element to represent it. (chapter)
5. **aside** : represents a portion of a document whose content is only indirectly related to the document's main content (side bar)

6. **nav** : represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents.

Note:

1. If no sure, use div.
2. Division of areas lays a foundation for styling.

Element-permitted-content

Previous: block level element can include inline elements, while inline element cannot include block level element.

Element inclusion relationship is determined by content Type.

- most elements used in body element are categorized as flow content and flow content can have different types.
    - sectioning content
    - heading content
    - phrasing content
    - etc.
- each element has their permitted content (which elements can be contained inside them).

Conclusion:

1. container element could contain any elements.
2. a element can almost wrap any element.
3. certain element has their fixed children elements (ul > li, ol > li, dl > dt + dd)
4. No nesting between heading and paragraph, no container elements inside them as well.

e.g., can h1 contain p element?
Nope. h1's permitted content is phrasing content (not include p element), same goes to p element (not include h element).

// Why browser is so tolerant of errors? Browser wars