Hypertext Markup Language (HTML)

Structure is a way to build a webpage to make it easier for the reader to understand what he is reading. For this HTML, Hyper Text Markup Language, code describes the structure of Web pages and it is the standard markup language for creating Web pages. HTML consists of a series of elements that tell the browser how to display the content in a Web pages.

The HTML code

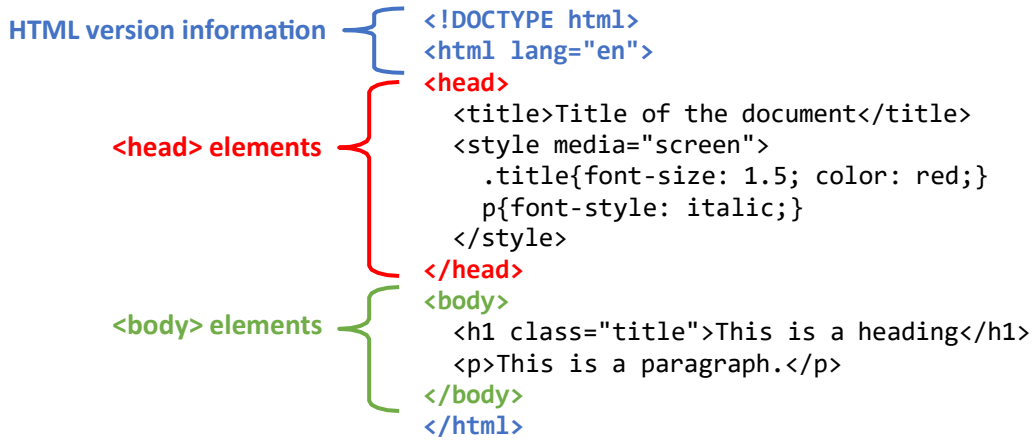
The HTML code is made up of characters that live inside angled brackets which are called HTML **elements**. Elements are usually made up of two **tags:** an opening tag and a closing tag.

Each HTML element tells the browser something about the information that sits between its opening and closing tags.

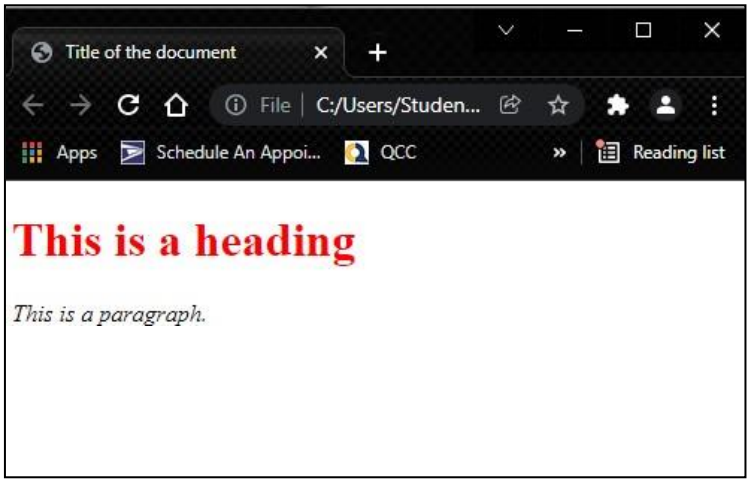
Basic HTML elements

An HTML document has three required parts:

- a line containing HTML version information. When the webpage is loaded, the declaration line is the first line that the browser reads. It tells the browser that the following file is an HTML file.
- a head element. The head element contains metadata, which is data about the HTML document, that is not displayed on the browser when the webpage is loaded, except for the title.
- a body element. The body element contains the document's actual content. Elements in the body build the visual part of a webpage.



Browser display



The basics HTML elements are:

1. The opening **<html>** tag element indicates that anything between it and a closing **</html>** tag is HTML code
2. A **<head>** element contains information about the page such as title
3. The contents of the **<title>** element are either shown in the top of the browser, above where you usually type in the URL of the page you want to visit, or on the tab for that page (if your browser uses tabs to allow you to view multiple pages at the same time).
4. The **<body>** tag indicates that anything between it and the closing tag </body> should be inside the main browser window.
5. A paragraph of text appears between these ** ** and ** ** tags.
6. A heading element is uses to display titles or subtitles on a webpage **<h1>** and **</h1>.** The largest heading is h1, which is twice the normal size, and the smallest heading is h6, which is 0.75% of the normal size.

   Usually **<h1>** is used for main headings or titles, **<h2>** is used for subheadings, and if there are further sections under the subheadings, then the **<h3>** element is used, and so on.
7. There are many occasions when we need to use lists. HTML provides us with three different types: Ordered lists, unordered lists, definition or description lists.
   - The ordered list is created with the **
** element. Ordered lists are lists where each item in the list is numbered. Each item in the list is placed between an opening **
** tag and a closing **
** tag.
   - The unordered list is created with the **
** element. Unordered lists are lists that begin with a bullet point rather than characters that indicate order. Each item in the list is placed between an opening **
** tag and a closing **
** tag.
   - The definition list is created with the **<dl>** element. Definition lists are made up of a set of terms along with definitions for each of those terms. Inside the **<dl>** element you will usually see pairs of **<dt>** and **<dd>** elements. **<dt>** is used to contain the term being defined (the definition term). **<dd>** is used to contain the

definition.

Images in HTML

There are many reasons why you might want to add an image to a web page: you might want to include a logo, photograph, illustration, diagram, or chart.

Images should…

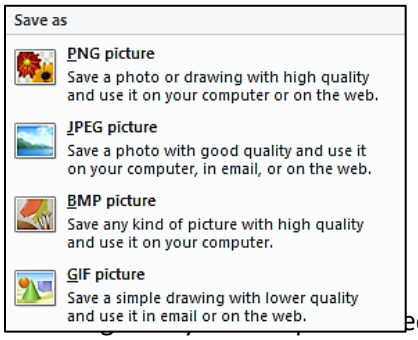
- Be relevant
- Convey information
- Convey the right mood
- Be instantly recognizable
- Fit the color palette

Stock photos

If you do not have photographs to use on your website, there are companies who sell stock images or icons:

- Free Stock Photos, Royalty Free Stock Images · Pexels https://www.pexels.com/
- Find your inspiration. | Flickr https://www.flickr.com/
- Free and premium vector icons, illustrations and 3D illustrations https://www.iconfinder.com/

Image formats



PNG Portable Network Graphics

JPG or JPEG Joint Photographic Experts Group BMP BitMaP

GIF Graphics Interchange format

een are made up of lots of tiny squares known as pixels. The resolution of the screen is the number of pixels represented on it, and on most computers you can increase and decrease this number.

JPEG offers good quality when the image has many different colors.

GIF or PNG images are low quality images which are good for images with few colors or large areas of the same color (flat color). Example of them are logos, illustrations, and diagrams.

SVG - scalable vector graphics

The other image format is SVG - scalable vector graphics.

The `<img>` SVG element includes images inside SVG documents. It can display raster image files or other SVG files. A ***raster image*** is an image file defined as a grid of pixels. They are also referred to as *bitmaps*.

Common raster image formats on the Web are JPEG, PNG, GIF, and ICO.

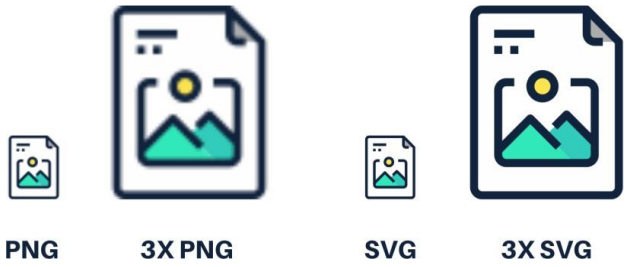
The only image formats SVG software must support are JPEG, PNG, and other SVG files. Animated GIF behavior is undefined.

Using SVGs is an easy choice once you consider the advantages they offer. For a client, you get superb quality on any device. For us as developers, there are even more reasons to use SVG.

Let's discuss some of the benefits of SVG now.

1. *Text-based format*
   SVG elements contain text, which greatly improves the accessibility of a website. But the main advantage is that this text is indexed by search engines. Therefore, a user can find an SVG file via Google.

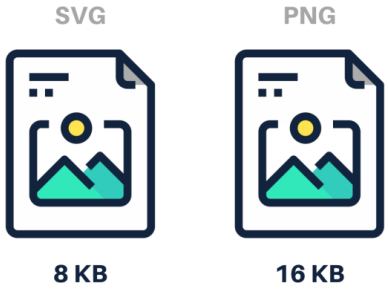
2. *Scalability*
   The quality of SVG images does not depend on the resolution. Unlike images of other formats or icon fonts, SVGs look perfectly sharp on any device with any screen size. Scalability also means that if you use the same image throughout the website but in different sizes, you use a single SVG. You do not have to create multiple copies of it as in the case of PNG. Instead, you embed the same image and define the size of it directly in SVG code.



3. *High performance*
   If you prioritize performance, you should use SVG. With SVG, there is no need for an HTTP request to load in an image file. The page loads faster as it has no files to download. Faster loading time translates into better webpage performance and higher search engine ranking. In turn, it improves user experience.

4. *Small file size*
   The size of simple SVG files is defined by the colors, layers, gradients, effects, and masks that it contains. The size of a PNG or any other raster graphics file is defined by the number of pixels that it consists of.

The larger a PNG image is, the heavier it gets in size. This is not the case for SVG icons, though. Also, SVGs can be optimized, and I will tell how later in this article.



5. Numerous editing and animating opportunities

Unlike raster images, vector images can be edited both in special vector drawing programs and directly in a text editor. You can also edit colors or sizes of SVG icons directly via CSS. As for animating SVGs, it can be done with the help of SMIL, Web Animations API, WebGL, or CSS animation. Scroll down to learn more about CSS animation of SVG images.

6. Integration with HTML, XHTML, and CSS

SVG was designed "to integrate with and extend other prominent open Web platform technologies, such as X/HTML, CSS, and Javascript", according to W3C. So, unlike different image formats, this format can be easily integrated with other documents and technologies.

7. W3C Document Object Model support

There is growing community support for SVG. The World Wide Web Consortium (W3C) has always claimed that the Internet cannot do without vector images. This organization basically created the SVG format, and they actively support it nowadays.

What Are the Inconveniences of SVG?

The large number of small parts makes the use of the SVG format irrational. The more parts an image consists of, the heavier it grows in size.

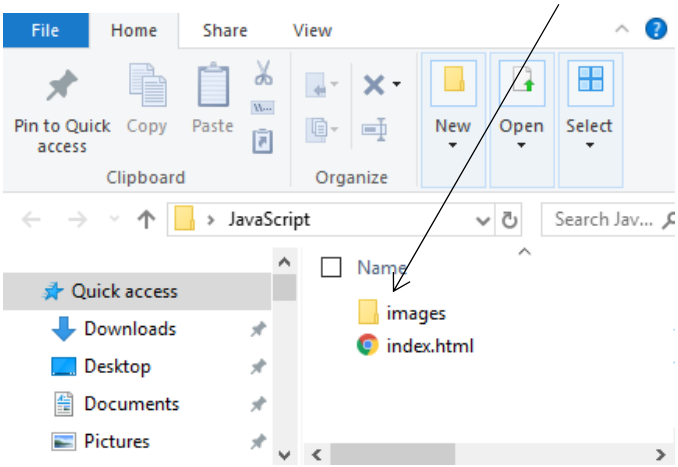
For example, here are two SVG maps of the United States. The second one is slightly more detailed than the first one. But the higher level of detail cost almost a fivefold increase in file size – 33 KB compared to 147 KB. If this map was not monochromatic, the increase would be much greater.

If the picture is linear and contains a few colors – SVG is a solution. However, if the details matter and there are a lot of them, PNG or JPEG may be more suitable. Also note that SVG cannot be used for

photographs. If you use a photograph on your website, SVG is not the best option. You definitely should go with a raster image format.

Store Images

As a website grows, keeping images in a separate folder helps you understand how the site is organized. Usually they are stored in sub-folder called “images”.



On a big site you might like to add subfolders inside the images folder. For example, images such as logos and buttons might sit in a folder called interface, product photographs might sit in a page called products, and images related to news might live in a folder called news.

Adding images to a webpage

To add an image into the page you need to use an **`<img>`** element. This is an empty element (which means there is no closing tag). It must carry the attribute src.

<img> attributes

`src` tells the browser where it can find the image file. This will usually be a relative URL pointing to an image on your own site.

`alt` provides a text description of the image which describes the image if you cannot see it.

`title` provides additional information about the image. Most browsers will display the content of this attribute in a tooltip when the user hovers over the image.

HTML5: figure and figure caption

HTML5 has introduced a new **`<figure>`** element to contain images and their caption so that the two are associated.

You can have more than one image inside the **`<figure>`** element as long as they all share the same caption.

The **`<figcaption>`** element has been added to HTML5 in order to allow web page authors to add a caption to an image.

HTML <picture> tag

The <picture> tag gives web developers more flexibility in specifying image resources. The most common use of the <picture> element will be for art direction in responsive designs. Instead of having one image that is scaled up or down based on the viewport width, multiple images can be designed to more nicely fill the browser viewport.

The <picture> element contains two tags: one or more <source> tags and one <img> tag.

The browser will look for the first <source> element where the media query matches the current viewport width, and then it will display the proper image (specified in the srcset attribute). The <img> element is required as the last child of the <picture> element, as a fallback option if none of the source tags matches.

Tip: The <picture> element works "similar" to <video> and <audio>. You set up different sources, and the first source that fits the preferences is the one being used.

Example) if we want to show four New York City photos when we change the browser window to:

- 1100px and up, shows photo of Empire State Building.
- Minimum width of 800px and less than 1100px, shows photo of Statue of Liberty.
- Minimum width of 500px and less than 800px, shows photo of Times Square.
- Any other pixels width, shows photo of Brooklyn Bridge.

The code will look as:

```
<p>Resize the browser window to load different photos of New York City </p>
<picture>
   <source media="(min-width:1100px)" srcset="EmpireStateBuilding.jpg">
   <source media="(min-width:800px)" srcset="StatueLiberty.jpg">
   <source media="(min-width:500px)" srcset="TimesSquare.jpg">
   <img src="BrooklynBridge.jpg" alt="Images of New York City" style="width:60%;">
   <figcaption>Images from <a href="https://www.pexels.com"> pexels.com
   </a></figcaption>
</picture>
```

Links

Links are the defining feature of the web because they allow you to move from one web page to another — enabling the very idea of browsing or surfing.

Writing Links

Links are created using the **<a>** element. Users can click on anything between the opening **<a>** tag and the closing **** tag.

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue

- A visited link is underlined and purple
- An active link is underlined and red

Some of **<a>** attributes are **href** and target:

- **href:** Specifies the URL of the page the link goes to.
- **target:** Specifies where to open the linked document. Some of the value of attribute target are: _blank, _parent, _self, _top

Linking to other sites

Links are created using the **<a>** element which has an attribute called **href**. The value of the **href** attribute is the page that you want people to go to when they click on the link. If you want a link to open in a new window, you can use the **target** attribute on the opening **<a>** tag. The value of this attribute should be **"blank".**

For example, create an external link that when a user clicks on the links `Visit QCC website`, it opens a QCC website in a new internet browser window.

```
<a href="http://www.qcc.cuny.edu" target="_blank">Visit QCC website</a>
```

Link to an Email Address

Use mailto: inside the href attribute to create a link that opens the user's email program (to let them send a new email):

```
<p><a href="mailto:hwu@qcc.cuny.edu">Send email to professor Wu</a></p>
```

HTML Links - Use an Image as a Link

To use an image as a link, just put the <img> tag inside the <a> tag:

```
<a href ="https://www.google.com" target="_blank">
    <img src="google_logo.svg" alt="Icon of Google Map" width="10%">
</a>
```

HTML Image Tags and Links

HTML images also use <map> and <area> to control the links within an image

Tag	Description
<map>	Defines an image map
<area>	Defines a clickable area inside an image map

HTML <map> tag

The <map> tag is used to define an image map. An image map is an image with clickable areas.

The required name attribute of the <map> element is associated with the <img>'s usemap attribute and creates a relationship between the image and the map.

The <map> element contains a number of <area> elements, that defines the clickable areas in the image map.

Specifying shape and size with coords

The coords attribute specifies the shape and size of the clickable area. The meaning of the coordinate values depends on the value of the shape attribute. In all cases, the (x, y) coordinates use the top-left corner as the origin (0, 0).

Coordinates for Rectangles

If the shape attribute is set to rect, the coordinates define the **top-left** and **bottom-right** of the rectangle. There should be four numeric values, separated by commas. The first two values are the (x, y) coordinates of the first corner. The third and fourth numbers are the (x, y) coordinates of the second corner.

Read more: https://html.com/attributes/area-coords/#ixzz88lY1cTwn

Example) if we want to select a rectangle shape to the following image:


```
<figure>
   <img src="fashion.jpg" width="400px" usemap="#fashionmap">
</figure>
```

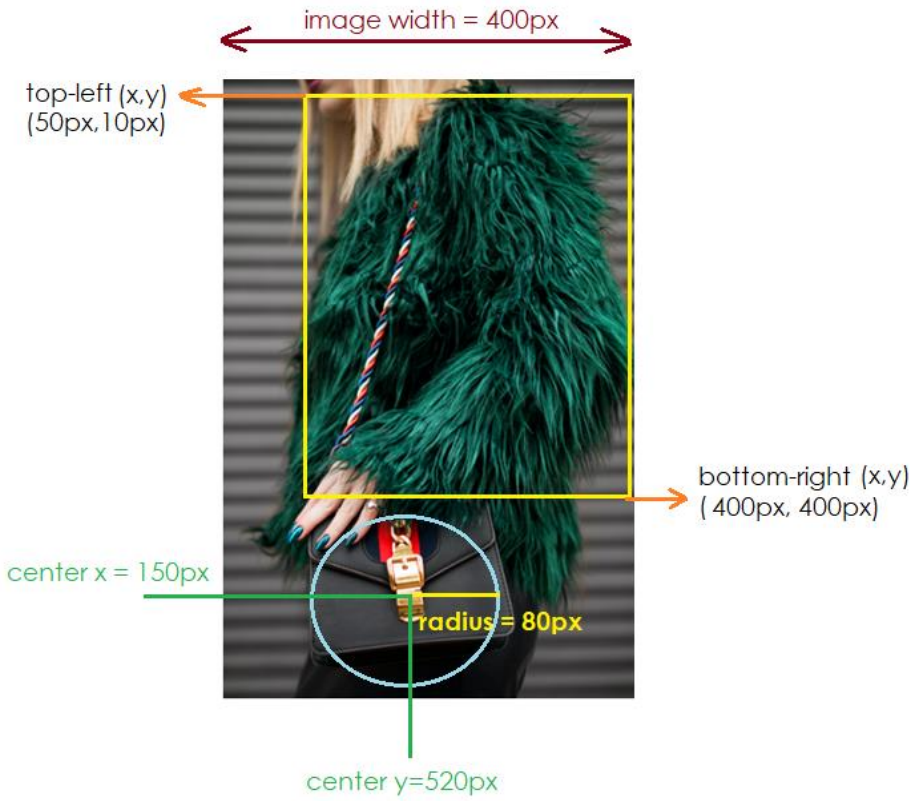
```
<map name="fashionmap">
      <area shape="rect" coords="50px,00px,400px,400px"
      href="https://www.macys.com/shop/womens-clothing " target="_blank">
</map>
```

Coordinates for Circles

If the shape attribute is set to circle, the coordinates define the center of the circle and the length of its radius. There should be three numeric values, the first indicating the (x, y) coordinates of the circle's center, and the third specifying the radius in pixels.

Read more: https://html.com/attributes/area-coords/#ixzz88lY6yTQu

For example, using the previous example, we can add a circle shape to the link as:



The <area> code will be as:

```
<area shape="circle" coords="150px,520px,80px"
href="https://www.macys.com/shop/handbags-accessories" target="_blank">
```

The complete code will look as:

```
<figure>
   <img src="fashion.jpg" width="400px" usemap="#fashionmap">
</figure>
```

```
<map name="fashionmap">
   <area shape="rect" coords="50px,00px,400px,400px"
   href="https://www.macys.com/shop/womens-clothing " target="_blank">
   <area shape="circle" coords="150px,520px,80px"
   href="https://www.macys.com/shop/handbags-accessories" target="_blank">
</map>
```

Linking to a specific part of the same page

Before you can link to a specific part of a page, you need to identify the points in the page that the link will go to. You do this using the **id** attribute (which can be used on every HTML element).

The value of the **id** attribute should start with a letter or an underscore (not a number or any other character) and, on a single page, no two id attributes should have the same value.

To link to an element that uses an **id** attribute you use the **<a>** element again, but the value of the **href** attribute starts with the **#** symbol, followed by the value of the **id** attribute of the element you want to link to.

Example) identify the part of your page where the **id bottom** will return when is clicked:

```
<img src="images/ny.jpg" id="bottom" alt="New York City“>
```

When the word ***Go to Image*** is clicked, it will link to an element with id **bottom:**

```
<p><a href="#bottom">Go to Image</a></p>
```

This is possible if the image is at the same webpage. If the link is in the other webpage, the code has to link to the webpage that it has followed by the id name:

```
<p><a href="index.com#bottom">Top</a></p>
```

How to link a webpage within the same site?

- To link to a file in the same folder, just use the file name

```
<p><a href="projects.html">Professional Projects</a></p>
```

- To link a file in a child folder, use the name of the child folder, followed by a forward slash, then the file name

```
<p><a href="images/imagesGallery.html">Project Gallery Images</a></p>
```

- To link to the homepage from a child folder, or to one folder before it, use **../**

  For example, if you are writing a code in webpage imagesGallery.html, which is located in a child folder “images”, and you want to make a link to go back to the index page, which is located in a parent folder, you will write the code as the following:

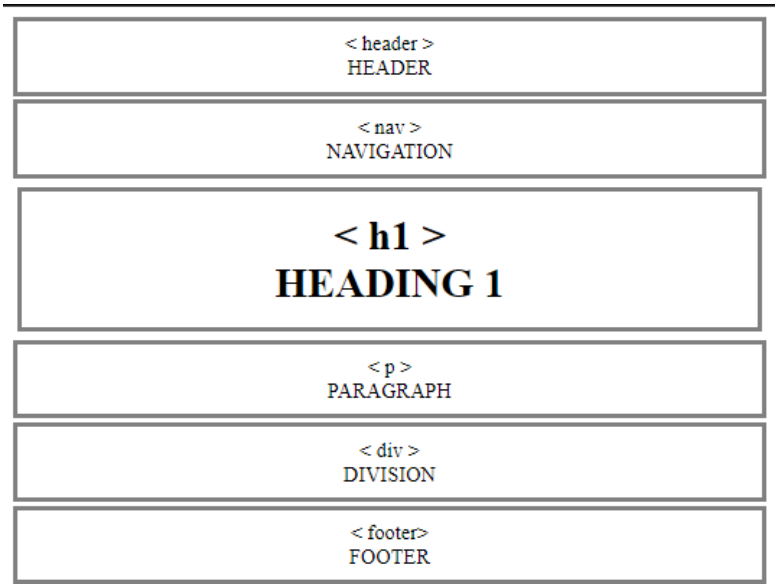
```
<p><a href="../index.html">Go back to homepage</a></p>
```

Block and inline elements

There are two display elements in HTML: block and inline.

Block elements

Block-level elements always start on a new line and takes a full width of the screen. It also has a top and bottom margin.



Some of the block-level elements in HTML are:

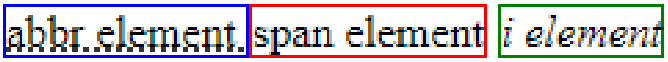
<address>	<figcaption>	<noscript>
<article>	<figure>	
<aside>	<footer>	<p>
<blockquote>	<form>	<pre>
<canvas>	<h1>-<h6>	<section>
<dd>	<header>	<table>
	<hr>	<tfoot>
<dl>		
<dt>	<main>	<video>
<fieldset>	<nav>	

Inline elements

Inline elements do not start on a new line; instead, it only takes up to the space as needed within the element content. They can set width and height values.

```
<abbr title="Abbreviation Template" style = "border:1px solid blue;"> abbr element </abbr>
<span style = "border:1px solid red;"> span element</span>
<i style = "border:1px solid green;">i element</i>
```

HTML code



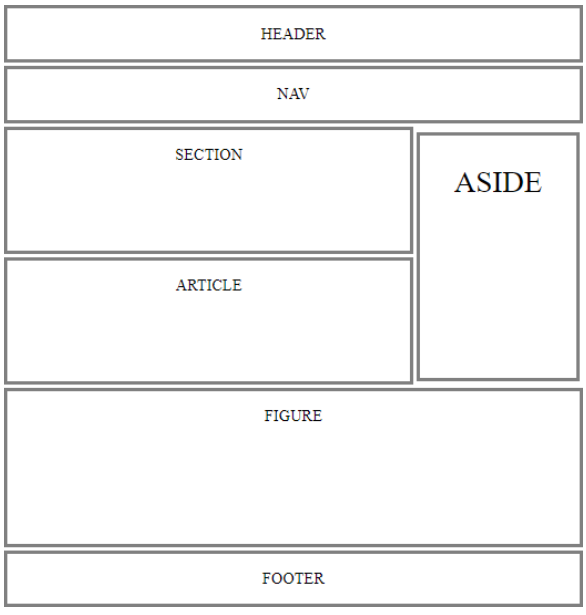
Some of the inline-level elements in HTML are:

<a>		<small>
<abbr>	<i>	
<acronym>	<img>	
<b>	<input>	<sub>
<bdo>	<kbd>	<sup>
<big>	<label>	<textarea>
 	<map>	<time>
<button>	<object>	<tt>
<cite>	<output>	<var>
<code>	<q>	<script>
<dfn>	<samp>	<select>

Semantic elements

Semantic elements are elements with a meaning and clearly describe its meaning to both, the browser and the developer.

There are some semantic elements in HTML that can use to create the layout of a webpage as shown in the image on the below:



Some of the semantic elements are:

Tag	Description
<article>	Defines independent, self-contained content
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<figcaption>	Defines a caption for a **<figure>** element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for a document or section
<header>	Specifies a header for a document or section
<main>	Specifies the main content of a document
<mark>	Defines marked/highlighted text
<nav>	Defines navigation links
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time

<section>

The <section> element defines a section in a document. According to W3C's HTML documentation: "A section is a thematic grouping of content, typically with a heading."

Examples of where a <section> element can be used:

- Chapters
- Introduction
- News items
- Contact information

A web page could normally be split into sections for introduction, content, and contact information.

<article>

The <article> element specifies independent, self-contained content. An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site.

Examples of where the <article> element can be used:

- Forum posts
- Blog posts
- User comments
- Product cards
- Newspaper articles

The <article> element specifies independent, self-contained content. The <section> element defines section in a document. Can we use the definitions to decide how to nest those elements? No, we cannot!

So, you will find HTML pages with <section> elements containing <article> elements, and <article> elements containing <section> elements.

<header>

The <header> element represents a container for introductory content or a set of navigational links. A <header> element typically contains:

- one or more heading elements (<h1> - <h6>)
- logo or icon
- authorship information

Note: You can have several <header> elements in one HTML document. However, <header> cannot be placed within a <footer>, <address> or another <header> element.

<footer>

The <footer> element defines a footer for a document or section.

A <footer> element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

You can have several <footer> elements in one document.

<nav>

The <nav> element defines a set of navigation links.

Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major blocks of navigation links.

Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

<aside>

The <aside> element defines some content aside from the content it is placed in (like a sidebar). The <aside> content should be indirectly related to the surrounding content.

<figure>

The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
The <figcaption> tag defines a caption for a <figure> element. The <figcaption> element can be placed as the first or as the last child of a <figure> element.

Bibliography

Duckett, J. (2016). *HTML and CSS Design and build websites.* Indianapolis: John Wiley and Sons Inc.

HTML, CSS, JavaScript, JQuery, and Bootstrap. (2017, June). Retrieved from w3schools: www.w3schools.com