Three Layers of Web Design

Structure, Style, Behavior

Structure	Presentation	Behavior
HTML	**CSS**	**Javascript**

What is HTML?

Hypertext Markup Language, a standardized system for tagging text files to achieve font, colour, graphic, and hyperlink effects on World Wide Web pages.

What is HTML?

HTML documents consist of a tree of elements and text. Each element is denoted in the source by a start tag, such as "<body>", and an end tag, such as "</body>".

History of HTML

- 1989 - Tim Berners-Lee invented **www**
- 1991 - Tim Berners-Lee invented **HTML**
- 1993 - Dave Raggett drafted **HTML+**
- 1993 - The first web browser, **Mosaic**, was introduced.
- 1994 - **Netscape browser**, based on Mosaic, was introduced.
- 1994 – **W3C** was formed

History of HTML

- 1995 - **HTML 2.0** was first published by W3C.
- 1997 - **HTML 3.0** was first published by W3C.
- 1999 - **HTML 4.01** was first published by W3C.
- 2000 – **XHTML 1.0** was published by W3C.
- 2008 – WHATWG published **HTML5** First Public Draft
- 2014 - **HTML5** was published by W3C.
- 2016 - **HTML5.1** was published by W3C.

Standards Organizations – WHATWG

Web Hypertext Application Technology Working Group (WHATWG)

The WHATWG was formed in response to the slow development of World Wide Web Consortium (W3C) Web standards and W3C's decision to abandon HTML in favor of XML-based technologies.

WHATWG wanted to develop HTML as a "**Living Standard**". A living standard is always updated and improved. New features can be added, but old functionality cannot be removed.

What is HTML5?

HTML5 is the latest version of Hypertext Markup Language, the code that describes web pages. It's actually three kinds of code: HTML, which provides the structure; Cascading Style Sheets (CSS), which take care of presentation; and JavaScript, which makes things happen.

Browser Support for HTML5

basic HTML document

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <title>Sample page</title>
 </head>
 <body>
  <h1>Sample page</h1>
  <p>This is a <a href="demo.html">simple</a> sample.</p>
  <!-- this is a comment -->
 </body>
</html>
```

<!DOCTYPE html>

- <!DOCTYPE html> goes at the top of every HTML5 page.
- The HTML5 word <!DOCTYPE html> means "this page is written in HTML5" as opposed to, say HTML 4.01.
- it's better than what they had before. Here's an example of one common type of XHTML:

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

Example Explained

The <html> element is the root element of an HTML page

The <head> element contains meta information about the document

The <title> element specifies a title for the document

The <body> element contains the visible page content

The <h1> element defines a large heading

The <p> element defines a paragraph

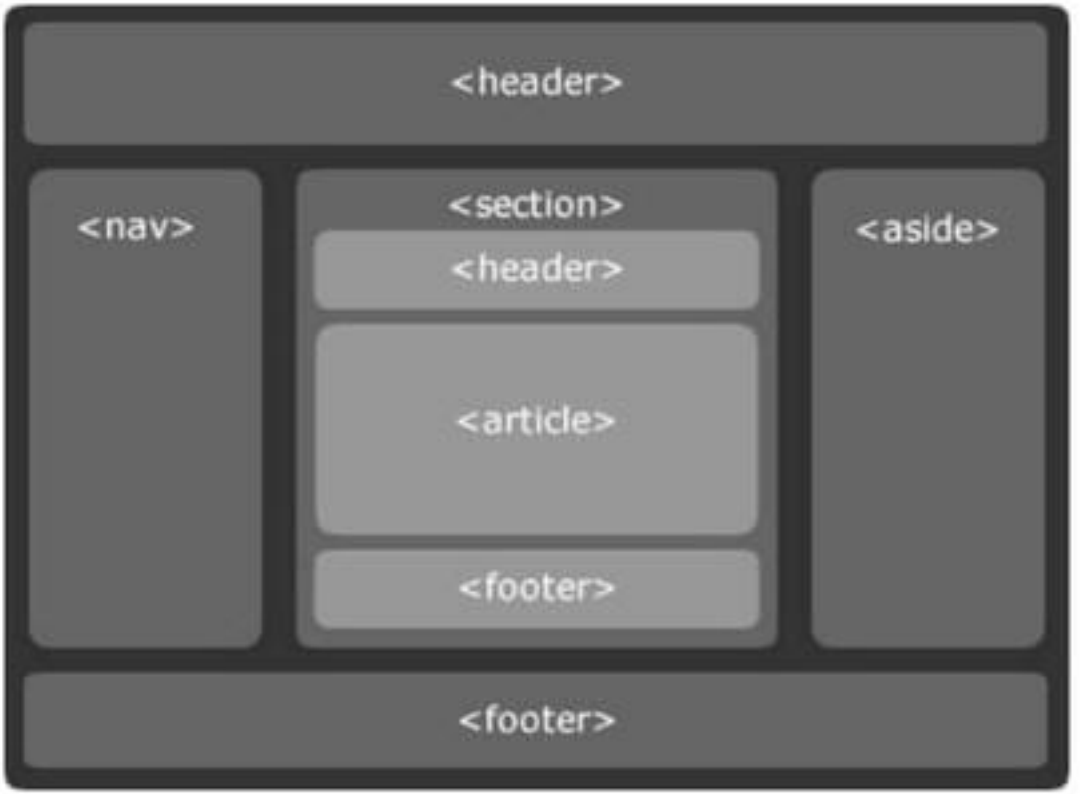
The New Elements of HTML5

For page layout and key features, there are now specific elements such as:

- <header> and <footer>
- <nav> for all the kinds of menus
- <aside> for sidebars or nearby related content
- <article> where content goes such as a blog post
- <section> similar to but more content-oriented
- <audio> and <video> tags to have native browsers manage playback of each. No more plugins and security updates for this
- <canvas> specifically for letting you draw graphics on using a separate scripting language
- <embed> to place external content or applications into the page

The New Elements of HTML5

Here's a nice visual outline of these by **Smashing Magazine**:



Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. **CSS3** is a latest standard of CSS earlier versions(CSS2).

Style Sheet

A stylesheet is a set of rules defining how an HTML element will be presented in the browser.

These rules are targeted to specific elements in HTML document

The Cascade

The **cascade** part of CSS is a set of rules for resolving conflicts with multiple CSS rules applied to the same elements.

For example, if there are two rules defining the color or your h1 elements, the rule that comes last in the cascade order will **trump** the other.

Inline Style

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<header style="color:blue;padding-left:20px;">This is a heading.</header>
```

Inline Stylesheet

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align: center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
<h1> Without Style</h1>
</body>
</html>
```

This is a heading

This is a paragraph.

Internal Style Sheet

An internal style sheet may be used if one single page has a unique style.

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

```
<head>
<style>
h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
```

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}
h1 {
  color: maroon;
  margin-left: 40px;
}
p {
  color: green;
  margin-left: 20px;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<h1>This is another heading</h1>
```

External Style Sheet

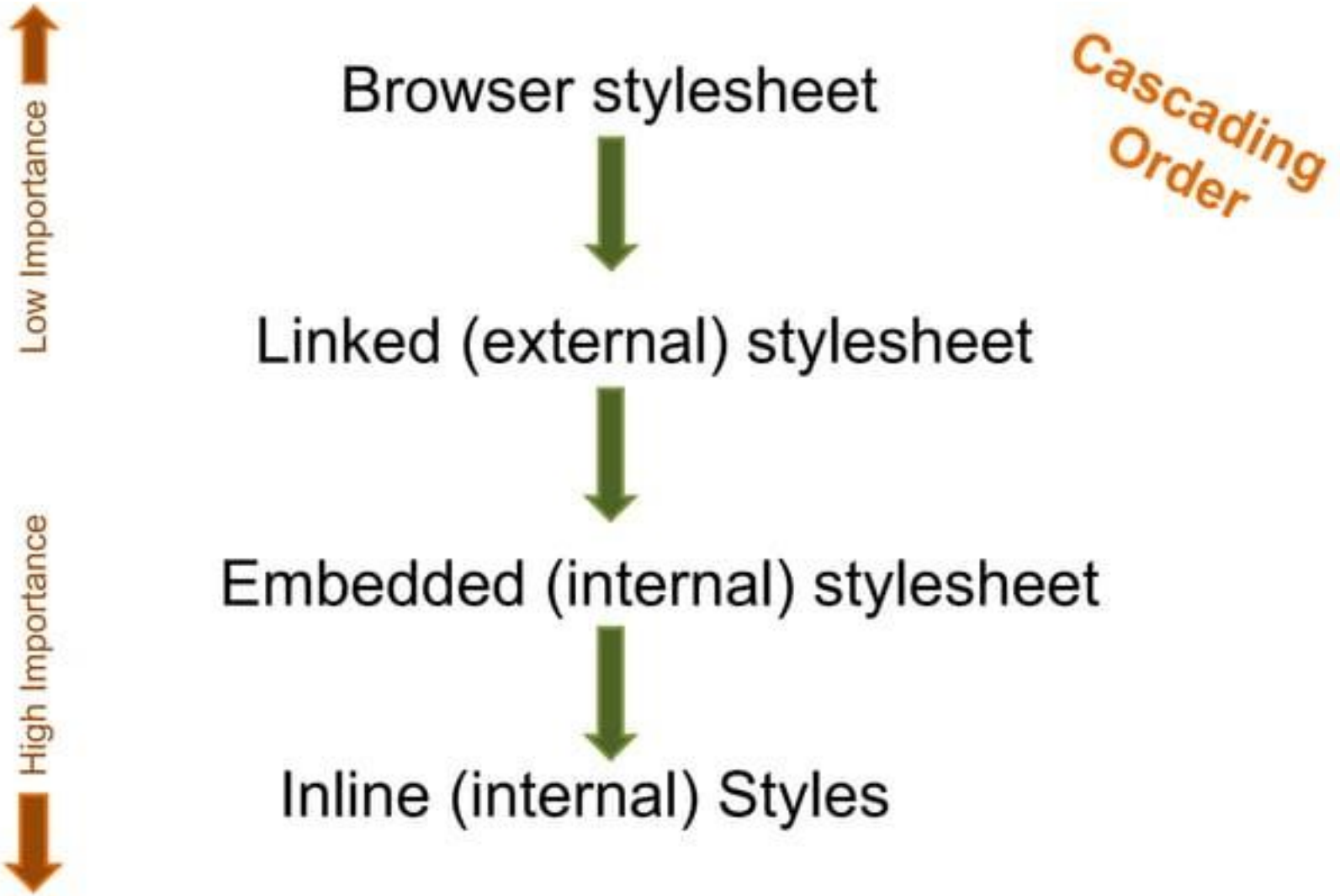
With an external style sheet, you can change the look of an entire website by changing just one file!

Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:

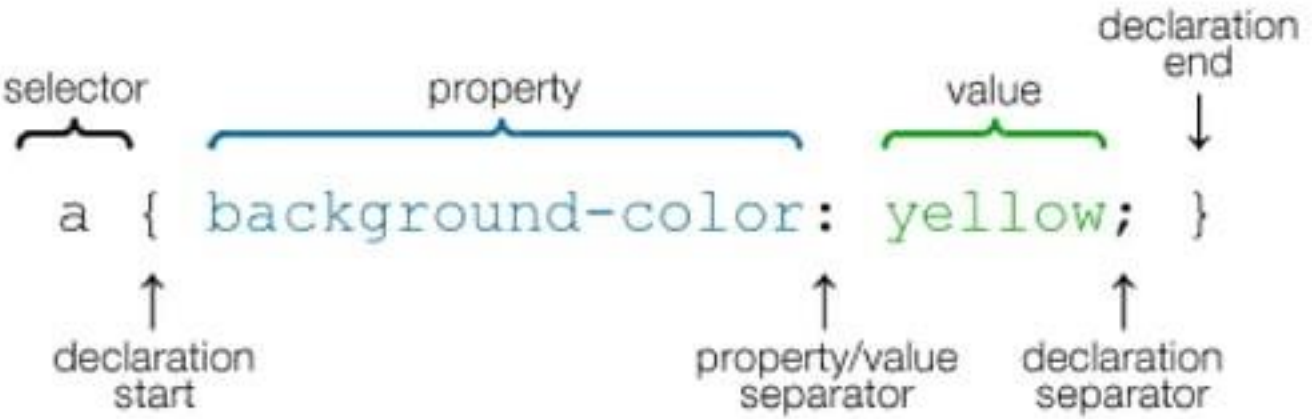
External Style Sheet

The external style sheet file must be saved with a .css extension.

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```


Low Importance
High Importance
Browser stylesheet
Linked (external) stylesheet
Embedded (internal) stylesheet
Inline (internal) Styles
Cascading Order

CSS Rules


selector
property
value
declaration end
a { background-color: yellow; }
declaration start
property/value separator
declaration separator

CSS Selectors

- Selectors are used to target which HTML to style. Properties and values are used to set the style rules.
- There are three kinds of selectors:

CSS Selectors - Tag

- Tag An HTML tag such as h1 or p.

HTML tag selector:

```
<p>Sample paragraph.</p>
```

HTML tag selector CSS:

```
p {
   color: red;
   font-size: 200%;
}
```

CSS Selectors - Class

- A class attribute of one or more elements. Referenced in CSS with a "." before it.

Class selector HTML:

```
<p class="warning">This is a paragraph with a class="warning".</p>
```

HTML class selector CSS:

```
.warning {
    background-color: #ffc;
}
```

CSS Selectors - Id

- An id attribute of a unique element, should only be used once. Referenced in CSS with a "#" before it.

Id selector HTML:

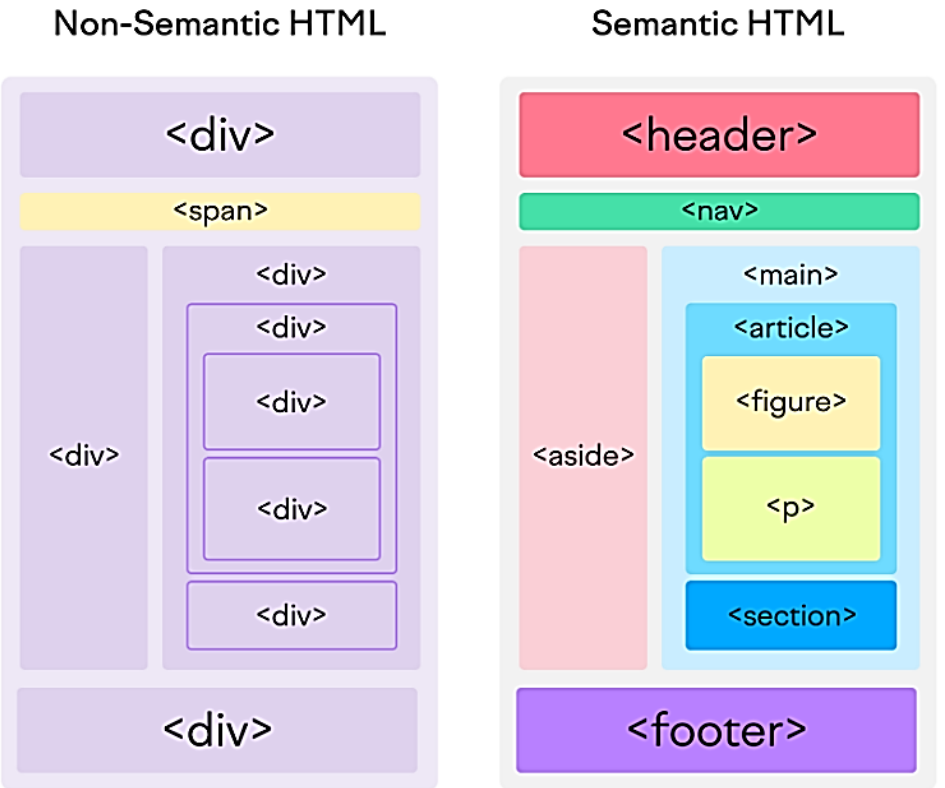
```
<h1 id="site-title">Sample Blog Title</h1>
```

HTML Id selector CSS:

```
#site-title {
   font-size: 3em;
}
```

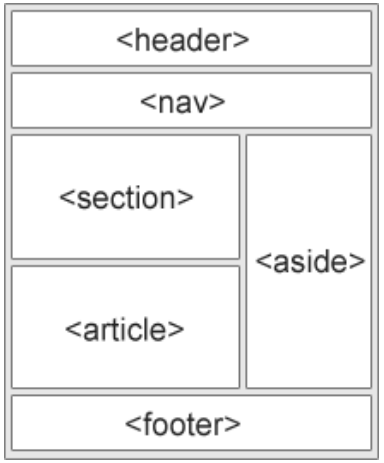
Semantic VS Non Semantic HTML

Semantic HTML, also known as semantic markup, refers to the use of HTML tags that convey the meaning—or semantics—of the content contained within them. By adding semantic HTML tags to your pages, you provide additional information that helps define the roles and relative importance of the different parts of your page. (As opposed to non-semantic HTML, which uses tags that don't directly convey meaning.)



HTML Layout Elements

HTML has several semantic elements that define the different parts of a web page:



- <header> - Defines a header for a document or a section
- <nav> - Defines a set of navigation links
- <section> - Defines a section in a document
- <article> - Defines an independent, self-contained content
- <aside> - Defines content aside from the content (like a sidebar)
- <footer> - Defines a footer for a document or a section

HTML Layout Example

Web Form 2.0

- Form elements and attributes in HTML5 provide a greater degree of semantic mark-up than HTML4 and remove a great deal of the need for tedious scripting and styling that was required in HTML4.
- The forms features in HTML5 provide a better experience for users by making forms more consistent across different web sites and giving immediate feedback to the user about data entry.

Web Form 2.0 – Input Types

- **url** - For entering a URL. It must start with a valid URI scheme, for example http://, ftp:// or mailto:.
- **tel** - For entering phone numbers. It does not enforce a particular syntax for validation, so if you want to ensure a particular format, you can use pattern.
- **email** - For entering email addresses, and hints that the @ should be shown on the keyboard by default.

Web Form 2.0 – Input Types

- **search** - A text input field styled in a way that is consistent with the platform's search field.
- **number** - For numeric input, can be any rational integer or float value.
- **range** - For number input, but unlike the number input type, the value is less important. It is displayed to the user as a slider control.

Web Form 2.0 – Input Types

- **datetime-local** - For entering a date and time value where the time zone provided is the local time zone.
- **date** - For entering a date (only) with no time zone provided.
- **time** - For entering a time (only) with no time zone provided.
- **week** - For entering a week (only) with no time zone provided.

Web Form 2.0 – Input Types

- **month** - For entering a month (only) with no time zone provided.
- **color** - For picking a color.

Web Form 2.0 – Attributes

- placeholder
- autofocus
- autocomplete
- required
- pattern
- novalidate
- formnovalidate
- list
- multiple
- form
- formaction
- formenctype
- formmethod
- formtarget

<audio>

The HTML <audio> element is used to embed sound content in documents. It may contain one or more audio sources, represented using the **src** attribute or the **<source>** element; the browser will choose the most suitable one.

Reference: https://developer.mozilla.org/en/docs/Web/HTML/Element/audio

<audio> example

Basic usage

```
<audio src="sample/sample.mp3" controls autoplay loop>
  This feature is not supported. please upgrade your browser.
</audio>
```

Audio element with source element

```
<audio controls="controls">
  Your browser does not support the <code>audio</code> element.
  <source src="foo.wav" type="audio/wav">
</audio>
```

<audio> - supported formats

						
MP3 (.mp3)	●	●	●	●	●	●
AAC (.m4a)	●	●	●	●	●	●
Ogg (.ogg)	●	●	●	●	●	●
WAV (.wav)	●	●	●	●	●	●

<video>

The HTML <video> element is used to embed video content in a document. It may contain one or more video sources, represented using the **src** attribute or the **<source>** element; the browser will choose the most suitable one.

Reference: https://developer.mozilla.org/en/docs/Web/HTML/Element/video

<video> example

Basic usage

```
<video src="sample/sample.mov" controls autoplay loop>
  This feature is not supported. please upgrade your browser.
</video>
```

Video element with source element

```
<video id="sampleMovie" width="640" height="360" controls>
            <source src="HTML5Sample_H264.mov" />
            <source src="HTML5Sample_Ogg.ogv" />
            <source src="HTML5Sample_WebM.webm" />
</video>
```

<video> - supported formats

						
MP4 (.mp4)	●	●	●	●	●	●
webm (.webm)	●	●	●	●	●	●
Ogv (.ogv)	●	●	●	●	●	●

<canvas> - supported contexts

- 2d - representing a two-dimensional rendering context.
- webgl - representing a three-dimensional rendering context. This context is only available on browsers that implement WebGl version 1 (OpenGL ES 2.0).
- webgl2 - representing a three-dimensional rendering context. This context is only available on browsers that implement WebGl version 2 (OpenGL ES 3.0).
- Bitmaprenderer - provides functionality to replace the content of the canvas with a given ImageBitmap.

<canvas> - How to Clear?

When any element on the canvas needs to change (move, erase, etc), the standard method is to completely erase the canvas and redraw the canvas with the elements in their new positions (or not redraw the elements if they are erased).

That's because canvas does not "remember" where it drew any individual element and therefore cannot individually move or erase any element.

<canvas> - How to Clear?

```
var canvas = document.getElementById('myCanvas');
var context = canvas.getContext('2d');
context.clearRect(0, 0, canvas.width, canvas.height);
```