

SEMANTIC TAGS



INTRODUCTION

Semantic tags are used in html to communicate meaning of the content to both the developer and the browser.

Semantic tags are exactly what they say, simple semantics. They all do the same thing as a div, but are designed to give even more context to your site through tags.

You might have noticed that the amount of divs required to keep yourself organized as your site adds more and more content can get a little crazy, you just see divs everywhere. Semantic tags are designed to give your tags a little more variation and context to what the grouping is actually doing.

SEMANTIC TAG EXAMPLES

There are lots of semantic tags that allow you to express what you are trying to achieve to your browser through code. These tags also help people that are using screen readers as they can understand where important content might be.

Again, these tags are functionally identical to div tags, but are simply more descriptive:

- header
- nav
- footer
- main
- section
- article
- figure

DISCLAIMER

We are literally talking about semantics here. What that means is there will be some grey area when it comes to questions like “When should I use tag x vs tag y” since they kind of do the same thing.

What I can say about semantic tags is that you should use them. They give text based context to your site which means that things like search engine robots, screen readers and other technologies will understand your site better.



ARTICLE

An HTML article is a semantic tag. Again, it has no visual difference in the look of your site when compared to using a div.

The idea of an article is to have a block of HTML that could be placed on any page in your site as a standalone element.

They are used for things like:

- Blog posts
- News stories
- Comments
- Forum Posts

ARTICLES CONT.

Example:

```
<article class="story">
```

```
  <h3>Title of the article</h3>
```

```
  <a href="link"></a>
```

```
    <p>This might be the first sentence of the story</p>
  </article>
```



SECTION

The section is used to split up our pages logically into their respective sections.

Your page might have a header, multiple sections of content, and then a footer. Sections are designed to be high level containers of content, less specific when compared to something like article.

SECTIONS CONT.

Example:

```
<section id="stories">
```

```
  <h1>News Today</h1>
```

```
  <article class="story">
```

```
    <h3>Title of the article</h3>
```

```
    <a href="link"></a>
```

```
      <p>This might be the first sentence of the story</p>
    </article>
```

```
  ... More stories here
```

```
</section>
```




HEADER

The header tag is for making headers on the page. Again because we are dealing with semantics, different developers like using different rules for header tags.

Some developers think the header tag should be used just once per page (at the top) and there are others that believe you should use header tags more often (like the header of a section or article).

I say, pick one and stick to it. If you want to use the header only for the top of your page feel free. If you want to use it more liberally, again feel free. Just make sure the usage makes sense! Don't use a header for the footnote of a section.



NAV

The nav tag can be used in any grouping of links to navigate your site. Note this does not mean a nav tag needs wrap every a tag your site contains, but any large grouping of internal links.

Think about the usual format of a site where you have the links around the top of the site. You might want to wrap those a tags in a nav tag!



FOOTER

The footer tag is very similar to the header tag except the opposite in usage. Again there is a debate when it comes to singular vs multiple uses per page, but my suggestion is the same.

Use one or use multiple, just keep constant in a site. If you use multiple, make sure the usage actually makes sense. Don't throw a footer in the middle of some content.



WHEN TO USE WHAT

With all of these tags that functionally do the same thing, it might be a little tough to figure out where to use what. My suggestion is to simply use semantic tags where they make logical sense to you!

If you want a grouping of tags but can't find a semantic tag that makes sense, use a div or span.

You will see in practice a lot of people just using div tags, but technically you do get a small SEO hit when not using semantic tags.

The goal is to try and make a more inclusive web page so that people with disabilities can enjoy your page as well.



KNOWLEDGE CHECK

Let's see if we can make a nice semantic website.

1. Create a folder called HtmlSemantics in your Scratch folder
2. Follow the Git workflow and get this connected to GitHub
3. Re-create the standard 3 tier website, but this time use semantic tags instead of div when appropriate.
 1. Create a title and welcome message at the top
 2. Create links to external websites you find interesting
 3. Create content in the middle of the site (can include images, text, videos, iframes, etc.)
 4. Create some contact information at the bottom of the site

Again you should be using semantic tags When possible! There might still be times when a div is needed as it doesn't make sense to use others, but lean towards the semantic tags when possible.

Not only do they make your code more readable (you can understand the logical flow of the source code better) but they also make using your site a much better experience for more users.