TASK

Semantic HTML

Visit our website

Introduction

WELCOME TO THE SEMANTIC HTML TASK!

Let’s dive into the latest and most enhanced version of HTML. You guessed it, I’m talking about Semantic HTML!

In this task, you will learn what Semantic HTML is and how to improve the way you create the markup for all your web pages.

Get in touch

Connect for support

Remember that with our courses, you’re not alone! You can contact an expert code reviewer to get support on any aspect of your course.

The best way to get help is to login to Discord at **https://discord.com/invite/hyperdev** where our specialist team is ready to support you.

Our code reviewers are happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!

WHAT IS SEMANTIC HTML?

In the previous task we introduced you to HTML, which is the World Wide Web's core markup language. HTML5 is currently the newest version of HTML which adds new elements and attributes.

One of the most important features of HTML5 is its semantics. The word **semantic** means *"relating to meaning"*. HTML was originally designed to semantically describe scientific documents; it has since evolved to describe much more. Semantic HTML refers to writing HTML in a way that is more comprehensible by better defining the different sections and layout of webpages.

When using Semantic HTML, we choose HTML elements based on their meaning, not on how they are presented. It also makes webpages more informative and adaptable, allowing browsers and search engines to interpret content better. Elements such as `<div>` and `<span>` are not semantic elements since they provide no context as to what is inside of those tags.

LIST OF COMMON SEMANTIC HTML ELEMENTS

<article> - This defines independent and self-contained content in a webpage, which is indented to be reusable.

Examples use cases:

- A forum post
- A magazine or newspaper article
- A blog entry
- A product card
- A user-submitted comment

Here is a demo of how to use the `<article>` element.

```
<article class="forecast">
    <h1>Weather forecast for Seattle</h1>
    <article class="day-forecast">
        <h2>03 March 2022</h2>
        <p>Rain.</p>
    </article>
    <article class="day-forecast">
        <h2>04 March 2022</h2>
        <p>Periods of rain.</p>
    </article>
    <article class="day-forecast">
        <h2>05 March 2022</h2>
        <p>Heavy rain.</p>
    </article>
</article>
```

<aside> - This defines a portion of a webpage whose content is only indirectly related to the webpage's main content.

The following example uses `<aside>` to mark up a paragraph in an article which is only indirectly related to the main article content.
<cite> - This defines the title of a creative work.

```
<article>
  <p>
    The Disney movie entitled <cite>The Little Mermaid</cite> was
    first released in theatres in 1989.
  </p>
  <aside>
    <p>
      The movie earned $87 million during its initial release.
    </p>
  </aside>
  <p>
    More info about the movie...
  </p>
</article>
```

<details> - This creates a widget in which information is visible only when the widget is toggled into an “open” state. A summary or label must be provided using the `<summary>` element.

Here is an example of the `<details>` element.

```
<details>
    <summary>Details</summary>
    Something small enough to escape casual notice.
</details>
```

<summary> - This specifies a summary, caption, or legend for a `<details>` elements’s disclosure box. Clicking on the `<summary>` element toggles the state of the parent `<details>` element between *open* and *closed*. See the example above under the `<details>` heading for an example of the `<summary>` element.

<figcaption> -This represents a caption or legend describing the rest of the contents of its parent `<figure>` element.

Here is an example of the `<figcaption>` element.

```
<figure>
    <img
src="https://hips.hearstapps.com/hmg-prod/images/womanyellingcat-1573233850.jpg"
         alt="A cat meme template">
    <figcaption>A cat meme template</figcaption>
</figure>
```

<figure> - This represents a self contained content like illustrations, diagrams, photos, and code listings which is specified using the `<figcaption>` element. You can see an example of using the `<figure>` element above.

<footer> - This defines a footer for a webpage or section. A `<footer>` typically contains information about the author of the section, copyright data, or links to related web pages.

Here is an example of the `<footer>` tag.

```
<article>
    <h1>How to be a wizard</h1>
    <ol>
        <li>Grow a long, majestic beard.</li>
        <li>Wear a tall, pointed hat.</li>
        <li>Have I mentioned the beard?</li>
    </ol>
    <footer>
        <p>(c) 2018 Gandalf</p>
    </footer>
</article>
```

<header> - The `<header>` element represents introductory content, typically a group of introductory or navigational aids.

The `<header>` element may contain:

- Heading elements
- A logo
- A search form
- An author name

Here is an example of the `<header>` element

```
<header class="page-header">
    <h1>Cute Cat Express!</h1>
</header>

<main>
    <p>I love cats <em>so</em> much! Like, really, a lot. They're adorable and so,
so snuggly soft!</p>
</main>
```

<main> - The `<main>` element represents the content of the `<body>` of a webpage. The main content area consists of content that is directly related to or expands upon the central topic of a webpage, or the central functionality of an application. Please look at the example under the `<header>` element for an example of the `<main>` element.

<nav> - The `<nav>` element represents a section of a webpage whose purpose is to provide navigation links, either within the current webpage or to other web pages.

Example use cases of navigation sections:

- Menus
- Tables of contents
- Indexes

```
<nav>
    <a href="/">Home</a>
    <a href="#about">About us</a>
    <a href="#articles">Articles</a>
    <a href="#contact">Contacts</a>
</nav>
```

<section> - This represents a generic standalone section of a webpage, which doesn't have a more specific semantic element to represent it. Sections should always have a heading, with very few exceptions.

Here is an example of the `<section>` element.

```
<section>
  <h1>Pink Flamingo</h1>
  <p>Plastic flamingos have been used as garden ornaments since the late 1950s.</p>
</section>
```

WHY USE SEMANTIC HTML?

ACCESSIBILITY

Screen readers and browsers can interpret Semantic HTML better, which makes webpages more accessible for people with disabilities.

SEO

Using Semantic HTML can improve website Search Engine Optimisation (SEO). SEO refers to the process of increasing the number of people that visit your webpage. With better SEO, search engines are better able to identify the content of your website and weigh the most important content appropriately.

EASY TO UNDERSTAND

Writing Semantic HTML makes code easier to understand, making the source code more readable for other developers.

A note from the
Hyperion Team

For more information about semantic HTML, please refer to this **playlist** of video tutorials. The entire tutorial series takes just over an hour, and should provide a lot of helpful context!

Compulsory Task

Create a webpage about your favourite hobby using semantic HTML.

- Your webpage needs to include, at a minimum, the following semantic elements: <footer>, <header>, <main>, <nav>, and <section>.
- Include at least three other HTML 5 tags on your webpage.

If you are having any difficulties, please feel free to contact our specialist team **on Discord** for support.

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we've done a good job?

Click here to share your thoughts anonymously.

REFERENCE

MDN Web Docs. Author unknown. (2022). **https://developer.mozilla.org**