COURSEWARE

Structural elements

Contents

Overview

HTML5 has a series of **structural elements** within the **Document Object Model (DOM)**, which allows for Web developers to create semantically structured pages.

These **semantic elements** and **non-semantic elements** allow for easy separation of content.

Tutorial

A **semantic element** clearly describes its meaning to both the browser and the developer - these include:

- `<header>`
- `<nav>`
- `<section>`
- `<main>`
- `<article>`
- `<aside>`
- `<footer>`
- `<form>`
- `<table>`

There are also **non-semantic** elements, which are designed purely to inform the browser of content-separation:

- `<div>`
- `<span>`

These do not have any bearing on the content of the Web page itself.

`<div>`

- The `<div>` tag defines a *division* within an HTML document.
- The `<div>` tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.
- The `<div>` tag is easily styled by using the `class` or `id` attributes.
- Any sort of content can be put inside the `<div>` tag!

```
<!Doctype html>
<html lang="en">
<head>
    <style>
        .myDiv{background-color:red;}
    </style>
</head>
<body>
    <div class="myDiv">
        <p>This is some text in a paragraph</p>
    </div>
</body>
</html>
```

(*note: By default, browsers always place a line break before and after the `<div>` element*)

Most browsers will display `<div>` element with the following default CSS values:

```
div {
    display:block;
}
```

Semantic elements

The **Document Object Model (DOM)** is the structure of a Web document, generated by the document's headings, form titles, table titles, and any other appropriate landmarks.

The user can apply this information to generate a table of contents, for example.

This table of contents could then be used by assistive technology to help the user, or be parsed by a machine like a search engine to improve search results.

The sectioning elements `<section>`, `<article>`, `<aside>` and `<nav>` can all help to create a more logical structure in the DOM.

`<header>`

The `<header>` element is normally the first element of the document, which generally acts as a container for hyperlinked images or text.

Headers contain levels from `<h1>` (highest level, largest font size) to `<h6>` (lowest level, smallest font size).

(*note: there can be one `<header>` per sectioning block.*)

```
<header>
  <a href="/">
    <img src="logo.png" alt="QA Home">
  </a>
  <h1>My Main Title</h1>
  <h2>My Sub title></h2>
</header>
```

`<nav>`

The `<nav>` tag is used to mark up *internal* navigation on a Web page, as opposed to linking to external sites.

However, not all groups of links on a page need to be in a `<nav>` element; it is primarily intended for sections that consist of major navigation blocks.

It is common for `<footer>`s to contain a list of links to various pages of a Web site (e.g. terms of service, homepage, contact).

The `<footer>` element alone is sufficient for such cases; while a `<nav>` element can be used in such cases, it is usually unnecessary.

`<nav>` elements can be nested inside other sectioning elements (most commonly the `<header>` and `<footer>`) and are typically surrounded by a `<li>` within a `<ul>`.

As such, multiple `<nav>`s are allowed, but each should contain a related category of links.

```
<nav>
  <h2>Main site navigation</h2>
  <ul>
    <li>
      <a href="/">Home</a>
    </li>
    <li>
      <a href="/about-us.html">About Us</a>
    </li>
  </ul>
</nav>
```

Sections

`<section>`s are used to break semantic elements like `<article>` or `<nav>` into smaller chunks. There are a few simple rules as to their use:

- They should not be used as containers for neither styles nor scripts (use `<div>` instead)
- They are the most generic and least meaningful non-semantic element, so use them sparingly
- They must be followed by a header (`<h1>` through `<h6>`) element

`<section>`, unlike `<div>`, provides a child level in the DOM:

```
<body>
...
  <section>
    <h2>heading level = section nesting level</h2>
    <p>rest of the content.....</p>
  </section>
...
</body>
```

A common mistake from new HTML5 Web developers is to suggest that the `<section>` tag renders the `<div>` tag obsolete, which is not the case - they are simply specialised for different uses:

- `<div>` is used for presentational structure - how it looks to the user
- `<section>` is used for informational structure - how it looks to the DOM

`<main>`

The `<main>` tag specifies the main content of a document.

The content inside the `<main>` element should be unique to the document. It should not contain any content that is repeated across documents, such as sidebars, navigation links, copyright information, site logos, and search forms:

```
<main>
  <h1>Some articles</h1>
  <article>
      <p>First Article Text</p>
  </article>
  <article>
      <p>Second Article Text</p>
  </article>
  <article>
      <p>Third Article Text</p>
  </article>
</main>
```

(*note: there must only be one `<main>` element in a document, and it must not be a descendant of an `<article>`,`<aside>`, `<footer>`, `<header>`, or `<nav>` element.*)

`<article>`

The `<article>` element represents an indivisible unit of work that makes up a significant section of data.

This may include a `<video>` element, a blog entry or news story, or results from a dynamic server page.

As is common with all of our sectioning elements, the first tag inside an `<article>` must be a heading:

```
<article>
    <h2>Yesterday</h2>
    <p>Some stuff goes here</p>
</article>
<article>
    <h2>Today</h2>
    <p>Some more stuff goes here</p>
</article>
```

`<aside>`

The `<aside>` element is used for providing related, but non-essential, tangential information to data which is presented within the `<main>` block:

When used within an `<article>` element, the contents should be specifically related to that article - here, we use it to enclose a glossary:

```
<article>
    <h1>My Blog Post</h1>
    <p>...</p>
    <aside>
       <h1>Glossary</h1>
        <dl> ... </dl>
    </aside>
</article>
```

When used as a child of the `<body>` tag, the contents of the `<aside>` tag should instead be specific to the purpose of the Web site itself - here, it is used to enclose an entire news feed and external navigation links:

```
<body>
    <aside>
        <h2>News feed</h2>
        <ul>
        <li><a href="#">Link 1</a></li>
        <li><a href="#">Link 2 Friend</a></li>
        </ul>
    </aside>
</body>
```

`<footer>`

The `<footer>` element represents a footer for a document or a `<section>` - generally it is used at the foot of a Web page, rather than a specific section, though they do see infrequent use. As such, many `<footer>` elements can appear in a Web page.

They can also *contain* entire `<sections>`, usually in the form of appendices, indices, sitemaps, and other less noteworthy Web page content.

Traditionally, `<footer>`s should contain `<small>` elements:

```
<footer>
  <small>
    Small footer
  </small>
</footer>
```

A more unconventional use of the `<footer>` element is a **fat footer**, which might contain images, external links, feedback or review boxes, advertising, etc.:

```
<footer>
  <p>Fat footer</p>
</footer>
```

Crib sheet

Element	Typical content	Typical parent element	Typical child element
`<header>`	Title, logo, banner, introductory information	Body,Section,Article	Nav, Section
`<nav>`	Primary navigation content	Body	Section, Nav
`<section>`	Generic page section	Body	Article, Header, Footer, Aside, Nav
`<article>`	Story, subsection, blog post	Body, Section	
`<aside>`	Sidebar content, tip, quotations	Body	Section, Article
`<footer>`	Footer, summary , copyright info, secondary navigation	Body, Section, Article	Nav, Section
`<main>`	Unique content, central to the topic of the document.	Body	Section, Article, Aside (non repetitive content like nav, header or footer)

Exercises

Use the base code below to change the elements to suit the new HTML5 structural element layout. Your task is as follows:

1. Replace any element with an appropriate structural element.
2. Once you have finished, check your outline matches the one below by testing the HTML structure on the HTML outliner website.

▶ Base code
▶ Solution