Chapter 20: Sectioning Elements

Section 20.1: Nav Element

The **<nav>** element is primarily intended to be used for sections that contain **main navigation blocks** for the website, this can include links to other parts of the web page *(e.g. anchors for a table of contents)* or other pages entirely.

Inline items

The following will display an inline set of hyperlinks.

```
<nav>
    <a href="https://google.com">Google</a>
    <a href="https://www.yahoo.com">Yahoo!</a>
    <a href="https://www.bing.com">Bing</a>
</nav>
```

Use list items when needed

If the content represents a list of items, use a list item to show this and enhance the user experience.

Note the `role="navigation"`, *more on this below.*

```
<nav role="navigation">
    <ul>
        <li><a href="https://google.com">Google</a></li>
        <li><a href="https://www.yahoo.com">Yahoo!</a></li>
        <li><a href="https://www.bing.com">Bing</a></li>
    </ul>
</nav>
```

Avoid unnecessary usage

<footer> elements may have a list of links to other parts of the site (FAQ, T&C, etc.). The footer element alone is sufficient in this case, you don't *need* to further wrap your links with a **<nav>** element in the **<footer>**.

```
<!-- the <nav> is not required in the <footer> -->
<footer>
    <nav>
        <a href="#">...</a>
    </nav>
</footer>

<!-- The footer alone is sufficient -->
<footer>
    <a href="#">...</a>
</footer>
```

> **Notes:**
>
> - **<main>** element descendants are not allowed within a **<nav>**
>
> Adding a `role="navigation"` ARIA role to the **<nav>** element is advised to aid user agents that don't support HTML5 and also to provide more context for those that do.

> ```
> <nav role="navigation"><!-- ... --></nav>
> ```

> **Screen Readers:** *(software that allows blind or visually impaired users to navigate the site)*
>
> User agents like screen readers will interpret the `<nav>` element differently depending on their requirements.
>
> - It could give the `<nav>` element a higher priority when rendering the page
> - It could delay the rendering of the element
> - It could adapt the page in a specific way to tailor for the user's needs
>   *example:* make the text links within the `<nav>` elements larger for someone who's visually impaired.

Click here to read the official HTML5 Specification for the `<nav>` element

Section 20.2: Article Element

The `<article>` element contains **self-contained content** like articles, blog posts, user comments or an interactive widget that could be distributed outside the context of the page, for example by RSS.

- When article elements are nested, the contents of the inner article node should be related to the outer article element.

A blog (`section`) with multiple posts (`article`), and comments (`article`) might look something like this.

```
<section>
    <!-- Each individual blog post is an <article> -->
    <article>
        <header>
            <h1>Blog Post</h1>
            <time datetime="2016-03-13">13th March 2016</time>
        </header>

        <p>The article element represents a self contained article or document.</p>
        <p>The section element represents a grouping of content.</p>

        <section>
            <h2>Comments <small>relating to "Blog Post"</small></h2>

            <!-- Related comment is also a self-contained article -->
            <article id="user-comment-1">
                <p>Excellent!</p>
                <footer><p>...</p><time>...</time></footer>
            </article>
        </section>
    </article>

    <!-- ./repeat: <article> -->

</section>

<!-- Content unrelated to the blog or posts should be outside the section. -->
<footer>
    <p>This content should be unrelated to the blog.</p>
</footer>
```

Avoid unnecessary usage

When the main content of the page (excluding headers, footers, navigation bars, etc.) is simply one group of elements. You can omit the `<article>` in favour of the `<main>` element.

```
<article>
    <p>This doesn't make sense, this article has no real `context`.</p>
</article>
```

Instead, replace the article with a `<main>` element to indicate this is the `main` content for this page.

```
<main>
    <p>I'm the main content, I don't need to belong to an article.</p>
</main>
```

If you use another element, ensure you specify the `<main>` ARIA role for correct interpretation and rendering across multiple devices and non HTML5 browsers.

```
<section role="main">
    <p>This section is the main content of this page.</p>
</section>
```

> **Notes:**
>
> - `<main>` element descendants are not allowed within a `<article>`

Click here to read the official HTML5 Specification for the `<article>` element

Section 20.3: Main Element

The `<main>` element contains the **main content** for your web page. This content is unique to the individual page, and should not appear elsewhere on the site. Repeating content like headers, footers, navigation, logos, etc., is placed outside the element.

- The `<main>` element should only ever be used at most **once** on a single page.
- The `<main>` element must not be included as a descendant of an `article`, `aside`, `footer`, `header` or `nav` element.

In the following example, we're displaying a **single blog post** (and related information like references and comments).

```
<body>
    <header>
        <nav>...</nav>
    </header>

    <main>
        <h1>Individual Blog Post</h1>
        <p>An introduction for the post.</p>

        <article>
            <h2>References</h2>
            <p>...</p>
        </article>

        <article>
```

```
            <h2>Comments</h2> ...
        </article>
    </main>

    <footer>...</footer>
</body>
```

- The blog post is contained within the `<main>` element to indicate this is the main content for this page (and therefore, unique across the website).
- The `<header>` and `<footer>` tags are *siblings* to the `<main>` element.

> **Notes:**
>
> The HTML5 specification recognizes the `<main>` element as a **grouping** element, and not a *sectioning* element.
>
> - ARIA role attributes: `main` *(default)*, `presentation`
>
> Adding a `role="main"` ARIA role attribute to **other elements** intended to be used as main content is advised to aid user agents that don't support HTML5 and also to provide more context for those that do.
>
> The `<main>` element by default has the main role, and so does not need to be provided.

Click here to read the official HTML5 Specification for the `<main>` element

Section 20.4: Header Element

The `<header>` element represents introductory content for its nearest ancestor sectioning content or sectioning root element. A `<header>` typically contains a group of introductory or navigational aids.

> **Note:** The header element is not sectioning content; it doesn't introduce a new section.

Examples:

```
<header>
  <p>Welcome to...</p>
  <h1>Voidwars!</h1>
</header>
```

In this example, the `<article>` has a `<header>`.

```
<article>
  <header>
    <h1>Flexbox: The definitive guide</h1>
  </header>
  <p>The guide about Flexbox was supposed to be here, but it turned out Wes wasn’t a Flexbox expert
either.</p>
</article>
```

W3C Proposed Recommendation

Section 20.5: Footer Element

The `<footer>` element contains the footer part of the page.

Here is an example for `<footer>` element that contain `p` paragraph tag.

```
<footer>
    <p>All rights reserved</p>
</footer>
```

Section 20.6: Section Element

The `<section>` element represents a generic section to thematically group content. Every section, typically, should be able to be identified with a heading element as a child of the `section`.

- You can use the `<section>` element within an `<article>` and vice-versa.
- Every section should have a *theme* (a heading element identifying this region)
- Don't use the `<section>` element as a general styling 'container'.
  If you need a container to apply styling, use a `<div>` instead.

In the following example, we're displaying a **single blog post** with multiple chapters each chapter is a section *(a set of thematically grouped content, which can be identified by the heading elements in each section)*.

```
<article>
    <header>
        <h2>Blog Post</h2>
    </header>
    <p>An introduction for the post.</p>
    <section>
        <h3>Chapter 1</h3>
        <p>...</p>
    </section>
    <section>
        <h3>Chapter 2</h3>
        <p>...</p>
    </section>
    <section>
        <h3>Comments</h3> ...
    </section>
</article>
```

> **Notes:**
>
> Developers should use the **article** element when it makes sense to syndicate the contents of the element.

Click here to read the official HTML5 Specification for the `<main>` element