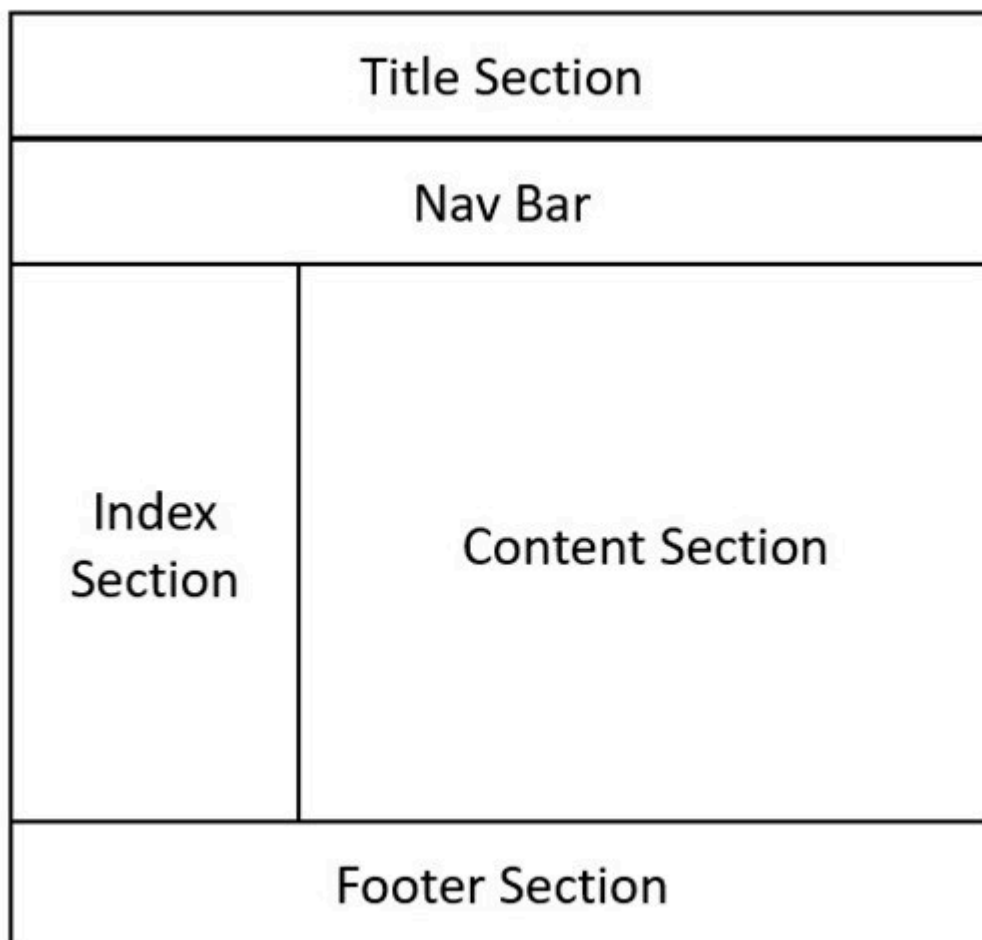# HTML - Layout Elements

What are HTML Layout Elements?

HTML layout elements are special semantic elements that are used to define the structure and layout of a web page. These layout elements are useful to arrange (divide) the content into logical sections that improve the readability and accessibility of the webpage for readers.

HTML Design Using Layout Elements

The below figure illustrates how a typical HTML webpage layout is designed. Most of the web pages have a title section, a nav bar, an index section, a content section, and a footer section, which can be defined using the `<header>`, `<nav>`, `<section>`, `<article>`, and `<footer>` elements, respectively.

Explore our **latest online courses** and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

Key HTML Layout Elements

Each layout element has a specific meaning and function and can be customized with attributes and styles. They describe the content they contain, not just the appearance of a web page. They are as follows:

HTML <header> Element

The **<header> element** is used to add a header section in an HTML web page. It serves as a container for the introductory and navigational part of a page. All the content inside this tag will be on top of the webpage. Providing a header section to your webpage helps search engines to understand your contents easily and rank the page accordingly.

Example

The following example demonstrates how you can create a header layout for an HTML webpage:

```
</>                                          Open Compiler

<!DOCTYPE html>
<html lang="en">

<head>
    <title>Tutorialspoint</title>
</head>

<body>
   <header>
   <div>
      <h1>Tutorialspoint</h1>
      <h3>Simply easy learning!</h3>
   </div>
   </header>
</body>

</html>
```

HTML <nav> Element

The **<nav> element** represents a section of a page within the webpage, where it has hyperlinks to other pages or parts within the page (just like the menu bar). It is usually included inside a **<header>** element or **<section>** element.

Example

The following example demonstrates how you can create a navigation menu (nav section) in an HTML page:

```
</>                                          Open Compiler

<!DOCTYPE html>
<html lang="en">

<head>
    <title>Tutorialspoint</title>
</head>

<body>
  <header>
  <div>
    <h1>Tutorialspoint</h1>
    <h3>Simply easy learning!</h3>
  </div>
  <div>
    <nav>
      <ul>
        <li>
          <a href="#">
            Home
          </a>
        </li>
        <li>
          <a href="#">
            HTML
          </a>
        </li>
        <li>
          <a href="#">
            CSS
          </a>
        </li>
        <li>
          <a href="#">
            Python
          </a></li>
        <li>
          <a href="#">
            JavaScript
```

```
                </a>
            </li>
        </ul>
    </nav>
    </div>
</header>
</body>


</html>
```

HTML \<section> Element

The **\<section> element** defines a major part of the web page where all the important content will be displayed. There can be multiple sections in a single page

Example

The following example demonstrates how to use HTML layout elements like **\<header>**, **\<nav>**, and **\<section>** to structure a web page:

```
</>                                                    Open Compiler

<!DOCTYPE html>
<html lang="en">

<head>
    <title>Tutorialspoint</title>
</head>

<body>
    <header>
        <h1>Tutorialspoint</h1>
        <h3>Simply easy learning!</h3>
        <nav>navigation bar</nav>
    </header>
    <section>Section 1</section>
    <section>Section 2</section>
    <section>Section 2</section>
</body>

</html>
```

HTML \<footer> Element

The `<footer>` **element** defines the footer section of the webpage. This section contains copyright information, social media links, and other important details. It will always be at the bottom of the webpage.

Example

The following example demonstrates how to create a responsive HTML footer with copyright information and social media links, styled using CSS for a consistent layout at the bottom of the webpage:

```
</>                                                    Open Compiler

<!DOCTYPE html>
<html lang="en">

<head>
    <title>Footer Example</title>
    <style>
        body {
            display: flex;
            flex-direction: column;
            min-height: 100vh;
            margin: 0;
        }
        .content {
            flex: 1;
        }
        footer {
            background-color: #333;
            color: #fff;
            text-align: center;
            padding: 20px 0;
        }
        footer .social-media a {
            color: #fff;
            margin: 0 10px;
            text-decoration: none;
        }
        footer .social-media a:hover {
            color: #ddd;
        }
    </style>
</head>

<body>
    <div class="content">
```

```
    <h2>Footer Element of HTML Layout</h2>
    <p>
        The footer tag defines the footer section of
        the webpage. This section contains copyright
        information, social media links, and other
        important details. It will be always at the
        bottom of the webpage.
    </p>
</div>
<footer>
    <p>©
        2024 Tutorialspoint. All rights reserved.
    </p>
    <div class="social-media">
        <a href="#">Facebook</a>
        <a href="#">Twitter</a>
        <a href="#">Instagram</a>
        <a href="#">LinkedIn</a>
    </div>
</footer>
</body>

</html>
```

HTML <article> Element

The `<article>` **element** specifies independent, self-contained content such as a forum post, magazine, blog post, and so on. When an HTML article element is nested, the inner element represents the article related to the outer element. For example, comments on social media posts can be an article element nested in the article representing the social media post. It is mostly used for forum posts, magazine or newspaper articles, blog entries, product cards, etc.

Example

The following example demonstrates how to use the HTML `<article>` tag to structure a blog post with a title, metadata, and content, styled for a clean layout:

```
</>                                                    Open Compiler

<!DOCTYPE html>
<html lang="en">

<head>
    <title>Article Example</title>
    <style>
```

```
        article {
            background: #fff;
            margin: 20px 0;
            padding: 20px;
        }
    </style>
</head>

<body>
    <header>
        <h1>My Blog</h1>
    </header>

    <main>
        <article>
            <h2>Understanding the HTML Article Tag</h2>
            <p>
                Posted on <time datetime="2024-06-20">
                June 20, 2024</time> by Farhan
            </p>
            <p>
                The article tag is commonly used for content
                such as blog posts, news articles, and user
                comments.
            </p>
            <p>
                Lorem ipsum dolor sit amet, consectetuer a
                cing elit. Aenean commodo ligula eget dolor.
                Aenean massa. Cum sociis natoque penatib
                magnis dis parturient montes, nascet
                mus. Donec quam felis, ultric
        </article>
    </main>

</body>
</html>
```

HTML <aside> Element

The `<aside>` **element** **example** specifies a section of content that is directly or indirectly related to the main content (like a sidebar). If you remove aside content from a web page, the main content will not be impacted because aside content is a separate, optional component of the page. This tag is often used for advertisements.

Example

The following example demonstrates how to use the HTML `<aside>` tag to display related content, such as links to tutorials, alongside the main article content in a structured and responsive layout.

```
</>                                                    Open Compiler

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Aside Tag Example</title>
    <style>
        body {
            display: flex;
            flex-direction: column;
            background-color: #f4f4f4;
        }

        main {
            display: flex;
            flex: 1;
            padding: 20px;
        }
        article {
            flex: 3;
            background: #fff;
            padding: 20px;
            margin-right: 20px;
        }
        aside {
            flex: 1;
            background: #fff;
            padding: 20px;
            border-radius: 8px;
        }

    </style>
</head>

<body>
    <header>
        <h1>My Blog</h1>
    </header>
    <main>
        <article>
            <h3>Articles...</h3>
        </article>
```

```
    <aside>
        <h2>Related Articles</h2>
        <ul>
            <li>
                <a href="/html/index.htm">
                    HTML Tutorial
                </a></li>
            <li>
                <a href="/css/index.htm">
                    CSS Tutorial
                </a></li>
            <li>
                <a href="/python/index.htm">
                    Python Tutorial
                </a></li>
        </ul>
    </aside>
    </main>
</body>

</html>
```

Creating HTML Layout Using Layout Elements

The **layout elements** help to improve the readability and accessibility of the web page, as well as its SEO (search engine optimization) performance. Here, we are going to create a simple layout of a web page with the help of the above-mentioned semantic elements.

</>   Open Compiler

```
<!DOCTYPE html>
<html>

<head>
    <title>Layout structure of HTML</title>
    <style>
        * {
            box-sizing: border-box;
        }
        header {
            font-size: 25px;
            color: whitesmoke;
            padding: 1px;
            text-align: center;
            background-color: lightslategray;
```

```
        }
        nav {
            float: left;
            width: 20%;
            height: 350px;
            background: lightgray;
            padding: 20px;
        }
        nav ul {
            padding: 1px;
        }
        article {
            float: left;
            padding: 20px;
            width: 80%;
            background-color: lightyellow;
            height: 350px;
        }
        footer {
            background-color: lightslategray;
            padding: 10px;
            text-align: center;
            color: white;
            padding-top: 20px;
            padding-bottom: 10px;
        }
        footer a {
            margin: 10px;
        }
        footer p {
            margin-top: 30px;
        }
    </style>
</head>

<body>
    <!--header segment-->
    <header>
        <div>
            <p>Tutorialspoint</p>
            <p>Simply easy learning!</p>
        </div>
    </header>
    <section>
        <!--Menu Navigation segment-->
        <nav>
```

```html
      <ul>
         <li>
            <a href="#">Home</a>
         </li>
         <li>
            <a href="#">Jobs</a>
         </li>
         <li>
            <a href="#">Library</a>
         </li>
         <li>
            <a href="#">Articles</a>
         </li>
         <li>
            <a href="#">Certification</a>
         </li>
      </ul>
   </nav>
   <!--Content segment-->
   <article>
      <h1>Welcome to Tutorials point</h1>
      <p>
         Tutorialspoint.com is a dedicated page
         to provide quality online education in
         domains of Computer Science and other
         Technology, Programming Languages, and
         other Engineering subjects.
      </p>
   </article>
</section>
<!--Footer segment-->
<footer>
   <h2>Tutorialspoint</h2>
   <div>
      <a href="#"> About us </a>
      <a href="#"> Refund policy </a>
      <a href="#"> Terms of use </a>
      <a href="#"> Privacy policy </a>
      <a href="#"> FAQ's </a>
      <a href="#"> Affiliates </a>
      <a href="#"> Contact </a>
   </div>
   <div>
      <p>
         Copyrights © TUTORIALS POINT (INDIA)
         PRIVATE LIMITED. All rights reserved.
```

```
            </p>
        </div>
    </footer>
</body>

</html>
```