

HTML5 Semantic Elements



Content

1. What are HTML5 semantic elements?
2. Why use semantic elements?
3. Type of semantic elements.
 - i. Page Header - <header>
 - ii. Page Footer - <footer>
 - iii. Navigation Bar - <nav>
 - iv. Independent Content - <article>
 - v. Content Section - <section>
 - vi. Complementary Information - <aside>
 - vii. Main Content Area - <main>
 - viii. Visual Media Container - <figure>
 - ix. Figure Caption - <figcaption>
 - x. Highlighted Text - <mark>
 - xi. Expandable Details - <details>
 - xii. Details Summary - <summary>
 - xiii. Date or Time - <time>
 - xiv. Progress Indicator - <progress>
 - xv. Calculation Output - <output>

01.What are HTML5 semantic elements?

Semantic HTML, also known as semantic markup, refers to the use of HTML tags that convey the semantics meaning of the content contained within them. Semantic HTML provides more information that helps to convey the roles and importance of different parts of the web page. It makes the HTML more comprehensible by defining webpages' different sections and layouts.

Semantic HTML tags are used to define the meaning of the content they contain.

Tags like <header>, <article>, and <footer> are semantic HTML tags, they specify the role of the content present on them. While <div> and are typical examples of non-semantic HTML elements. They work only as content holders but do not indicate what type of content they contain or what role that particular content plays on the page.



Non-Semantic Elements

Semantic Elements

02. Why use Semantic Elements?

The main reason to use Semantic HTML tags is, it becomes easier to read and understand. Search engines weigh keyword importance by their placement in the HTML hierarchy. It means keywords enclosed in a <h1> tag are given more importance than those enclosed in an <p>. By placing the most important keywords higher up in the hierarchy, we're telling search what the page is about, hence improving the page.

- **Accessibility** :- Semantic elements make web pages more accessible. Screen readers and other assistive technologies can interpret the structure and navigate the content more efficiently.
- **SEO** :- Better structured data leads to better SEO. Search engines prioritize well-structured content that uses semantic elements correctly, as it's easier to index.
- **Maintainability** :- Semantic HTML helps create a logically structured document, which is easier to read and maintain.

Which is readable? Left or Right ?

```
<header> </header>
<section>
  <article>
    <figure>
      <img>
      <figcaption></figcaption>
    </figure>
  </article>
</section>
<footer> </footer>
```

Semantic elements

```
<div id="header"></div>
  <div class="section">
    <div class="article">
      <div class="figure">
        <img>
        <div class="figcaption">
        </div>
      </div>
    </div>
  </div>
  <div id="footer"></div>
```

Non-semantic element

03. Type of semantic elements.

1) Page Header - <header>

The <header> tag is a semantic HTML element that is used to define the introductory or navigational content of a webpage or a section.

Typically, a header contains elements like ,the website or page title, logo or branding , navigation menus, search bar and any introductory information relevant to the page or section.

Syntax :

<header>.....</header>

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <header>Internet Technology</header>
</body>
</html>
```

Internet Technology

2) Page Footer - <footer>

The <footer> tag in HTML is used to define the footer section of an HTML document.

The footer section typically contains information such as contact information, sitemap, back-to-top links, related documents, copyright, etc.

The footer tag is a semantic tag included in HTML (in 2014) along with other tags like article, nav, header, etc.

It is not mandatory, but adds to clear structure to the document and useful for SEO.

Syntax :

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

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <footer>
    <p>+94 701231025</p>
    <p>dilini@gmail.com</p>
    <p>No.145, Galler Road , Panadura</p>
  </footer>
</body>
</html>
```

+94 701231025

dilini@gmail.com

No.145, Galler Road , Panadura

3) Navigation Bar - <nav>

The <nav> tag in HTML is used to define navigation sections on a webpage. It typically contains links to important sections of the website, such as menus, tables of contents, or indexes, and is often structured within unordered lists () or as standalone links.

Syntax :

```
<nav>

    <!-- - navigation links -- >

</nav>
```

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <nav>
    <a href="">Home</a>
    <a href="">About</a>
    <a href="">Gallery</a>
    <a href="">Service</a>
    <a href="">Contact</a>
  </nav>
</body>
</html>
```

[Home](#) [About](#) [Gallery](#) [Service](#) [Contact](#)

4) Independent Content - <article>

The HTML <article> tag defines a self-contained, independent piece of content like a blog post, news article, or comment. It is designed for content that can be independently distributed, shared, or reused, providing semantic meaning to the content.

On a page with a single piece of content, a single <article> element can be used to contain the main content and set it off from the rest of the page.

On a page with multiple pieces of content (a blog index page, a search results page, a category page, news feed), multiple <article> elements can be used to contain each individual piece of content. It is similar to the <div> element and displays the stylish work the same. However, using the <article> element instead of <div> provides more semantic information to screen readers, search engines, and third-party applications.

Syntax :

```
<article>

    <!-- content -->

</article>
```

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <article>
    <h1>Hello, I'm Dilini Silva</h1>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusantium
      aliquid assumenda autem doloremque eveniet ex, hic labore non sunt veritatis! Ad
      animi dignissimos distinctio eaque, enim eum id nisi repudiandae.
    </p>
  </article>
</body>
</html>
```

Hello, I'm Dilini Silva

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusantium aliquid assumenda autem doloremque eveniet ex, hic labore non sunt veritatis! Ad animi dignissimos distinctio eaque, enim eum id nisi repudiandae.

5) Content Section - <section>

The Section tag defines the section of documents such as chapters, headers, footers, or any other sections. The section tag divides the content into sections and subsections.

The section tag is used when requirements of two headers or footers or any other section of documents are needed. Section tag grouped the generic block of related contents. The main advantage of the section tag is, it is a semantic element, that describes its meaning to both the browser and the developer.

Syntax :

<section>

<!-- content -->

</section>

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <section>
    <h1>About us</h1>
    <p>
      lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor
    </p>
  </section>
  <section>
    <h1>Projects</h1>
    <a href="">Project 1</a>
    <a href="">Project 2</a>
    <a href="">Project 3</a>
  </section>
  <section>
    <h1>Service</h1>
    <p>
      lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor
    </p>
  </section>
</body>
</html>
```

About us

lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor

Projects

[Project 1](#) [Project 2](#) [Project 3](#)

Service

lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor

6) Complementary Information - <aside>

The <aside> tag is used to describe the main object of the web page more shortly like a highlighter. It identifies the content that is related to the primary content of the web page but does not constitute the main intent of the primary page. The <aside> tag contains mainly author information, links, related content, and so on.

The aside content should be indirectly related to the surrounding content.

Syntax :

<aside>

<!-- content-- >

</aside>

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <h1>Java</h1>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquid beatae
    consequuntur culpa
    cum debitis dicta, enim facere magni maiores necessitatibus, nobis nostrum,
    quod repellat rerum sapiente totam unde. Commodi, quasi!
  </p>
  <aside>
    <h2>James Gosling</h2>
    <p>
      James Arthur Gosling OC is a Canadian computer scientist, best known as
      the founder and lead designer behind the Java programming language.
    </p>
  </aside>
</body>
</html>
```

Java

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquid beatae consequuntur culpa cum debitis dicta, enim facere magni maiores necessitatibus, nobis nostrum, quod repellat rerum sapiente totam unde. Commodi, quasi!

James Gosling

James Arthur Gosling OC is a Canadian computer scientist, best known as the founder and lead designer behind the Java programming language.

7) Main Content Area - <main>

The HTML <main> Tag defines a document's main content, which should be unique. It excludes content like sidebars, navigation, logos, and copyright info, ensuring unique document-specific material within.

The document must not contain more than one <main> element. The <main> element should not be a child element of an <article>, <aside>, <footer>, <header>, or <nav> element.

Syntax :

<main>

<!-- content -->

</main>

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <main>
    <article>
      <h1>Java</h1>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquid
        beatae consequuntur culpa
        cum debitis dicta, enim facere magni maiores necessitatibus, nobis
        nostrum, quod repellat rerum sapiente totam unde. Commodi, quasi!
      </p>
    </article>
    <article>
      <h1>Java Script</h1>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquid
        beatae consequuntur culpa
        cum debitis dicta, enim facere magni maiores necessitatibus, nobis
        nostrum, quod repellat rerum sapiente totam unde. Commodi, quasi!
      </p>
    </article>
  </main>
</body>
</html>
```

Java

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquid beatae consequuntur culpa cum debitis dicta, enim facere magni maiores necessitatibus, nobis nostrum, quod repellat rerum sapiente totam unde. Commodi, quasi!

Java Script

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquid beatae consequuntur culpa cum debitis dicta, enim facere magni maiores necessitatibus, nobis nostrum, quod repellat rerum sapiente totam unde. Commodi, quasi!

8) Visual Media Container - <figure>

The <figure> tag is used to insert self-contained content such as illustrations, diagrams, photos, or code listings in a document. It can be placed at any position in the document and is related to the main flow. The content inside the <figure> tag goes with the flow of the document, so if it is removed, it won't affect the flow of the document.

Syntax :

<figure>

<!-- image content-- >

</ figure >

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <main>
    <figure>
      
    </figure>
  </main>
</body>
</html>
```



9) Figure Caption- <figcaption>

The <figcaption> tag in HTML is used to provide a caption or description for a <figure> element.

This tag is typically used to describe an image, illustration, chart, or any other content that requires an explanation.

It helps to improve accessibility and search engine optimization (SEO) by associating descriptive text with visual content.

The <figcaption> element can be placed as the first or last child of the <figure> element.

Syntax :

```
<figure>

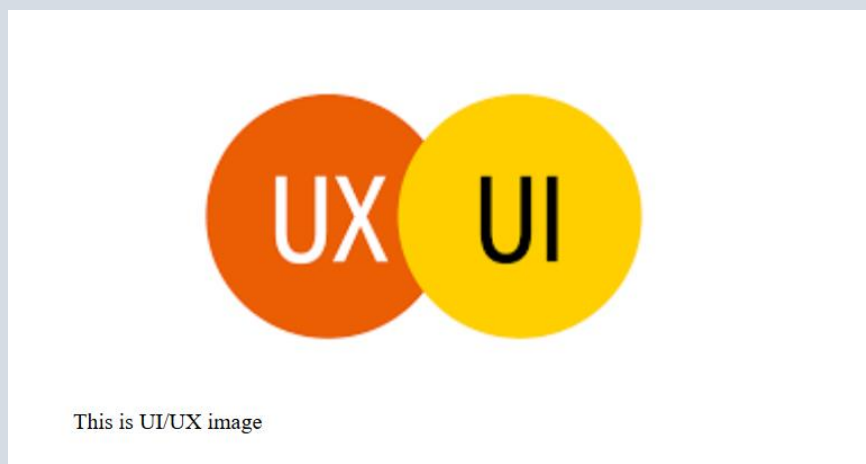
    <!-- -image content-- >

    <figcaption>.....</figcaption>

</ figure >
```

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <main>
    <figure>
      
      <figcaption>This is UI/UX image</figcaption>
    </figure>
  </main>
</body>
</html>
```



10) Highlighted Text- <mark>

The <mark> tag in HTML is used to define the marked text. It is used to highlight the part of the text in a paragraph.

Syntax :

<mark>

<!-- content-- >

</ mark >

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <h1>Java</h1>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit.
    Ab, autem <mark>fugiat harum id</mark>, illo in itaque magni mollitia
    placeat quo recusandae, saepe. Aperiam aspernatur culpa dolorum facere neque
    officiis, sunt?
  </p>
</body>
</html>
```

Java

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ab, autem **fugiat harum id**, illo in itaque magni mollitia placeat quo recusandae, saepe. Aperiam aspernatur culpa dolorum facere neque officiis, sunt?

11) Expandable Details - <details>

The <details> tag in HTML is used to create a disclosure widget from which the user can view or hide additional information. It is used with the <summary> tag, which provides the title or header for the details section. It's generally used for FAQs, dropdown menus, or to show/hide additional content.

Syntax :

```
<details>

    <!-- content-- >

</ details >
```

12) Details summary - <summary>

The HTML <summary> tag defines a summary for the <details> element. The <summary> tag is used along with the <details> element and provides a summary visible to the user. When the user clicks the summary, the content placed inside the <details> element becomes visible which was previously hidden.

The <summary> tag was added in HTML 5. The <summary> tag requires both starting and ending tags. The <summary> tag should be the first child element of the <details> element.

Syntax :

```
<details>

    <summary>...</summary>

</ details >
```

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    <details>
        <summary>UI/UX Project</summary>
        <p>
            lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod
            tempor incididunt ut labore et dolore magna aliqua.
        </p>
    </details>
</body>
</html>
```

► UI/UX Project

▼ UI/UX Project

lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

13) Date or Time - <time>

The <time> semantic HTML element represents a time-related piece of information, including:

- A specific date in the Gregorian calendar.
- The time on a 24-hour clock.
- A valid time duration. The <time> element accepts a datetime attribute that translates the info to a machine-readable format.

Its datetime attribute should be in a valid format for it to be translated to machine-readable info, see below examples.

Syntax :

<time>

<!--content-->

</time>

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <p>
    Exam is scheduled for
    <time datetime="2025-01-25">25th of January 2025</time> at
    <time datetime="19.00">7.00 pm</time>
  </p>
</body>
</html>
```

Exam is scheduled for 25th of January 2025 at 7.00 pm

14) Progress Indicator - <progress>

The <progress> tag is a semantic HTML element that is used to display a progress bar, a graphic indicator that represents the status of an event or process such as the level in a game, a question in an online exam, or parts of a form.

When using the max attribute in the <progress> element, the value must be greater than 0 and be a valid floating point number. If the value is not specified, the default value is 1.

Syntax :


```
<progress value = " " max = " " >
```

```
<!-- -percentage- - >
```

```
</ progress >
```

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<form action="/upload" method="POST">
  <label for="mp4-file">Upload MP4 file:</label>
  <input type="file" id="mp4-file" name="mp4-file">
  <br>
  <label for="mp4-progress">Upload progress:</label>
  <progress id="mp4-progress" value="50" max="100">50%</progress>
  <br>
</form>
</body>
</html>
```

Upload MP4 file: No file chosen
Upload progress: 

15) Calculation Output - <output>

The <output> HTML element is a container element into which a site or app can inject the results of a calculation or the outcome of a user action.

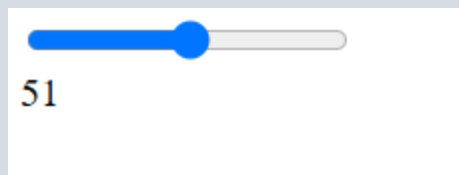
This element includes the global attributes. (for , id , name)

Syntax :

<output>....</ output >

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<form oninput="result.value=parseInt(c.value)">
  <input type="range" id="c" name="c">
  <br>
  <output name="result">0</output>
</form>
</body>
</html>
```



Thank You!