HTML TABLES

Content:

Tables

Table properties.

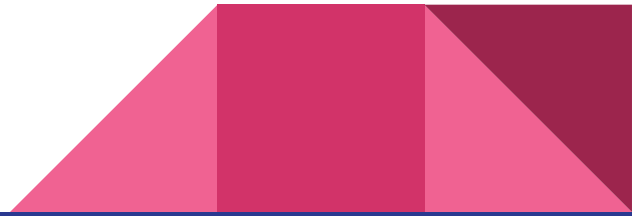
Heading, Phrasing, Interative sectioning tags.

Semantic tags in HTML5.

New Semantic elements: <dialog> <details> and <summary> .

Drag and drop.

Geolocation.



What is HTML Table ?

HTML Table is an arrangement of data in *rows* and *columns* in tabular format. Tables are useful for various tasks such as presenting text information and numerical data. A table is a useful tool for quickly and easily finding connections between different types of data. Tables are also used to create databases.

```
<table>
    <tr>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Age</th>
    </tr>
    <tr>
        <td>Priya</td>
        <td>Sharma</td>
        <td>24</td>
    </tr>
    <tr>
        <td>Arun</td>
        <td>Singh</td>
        <td>32</td>
    </tr>
    <tr>
        <td>Sam</td>
        <td>Watson</td>
        <td>41</td>
    </tr>
</table>
```

Output:

Firstname	Lastname	Age
Priya	Sharma	24
Arun	Singh	32
Sam	Watson	41

Tags used in HTML Tables

HTML Tags	Descriptions
<table>	Defines the structure for organizing data in rows and columns within a web page.
<tr>	Represents a **row** within an HTML table, containing individual cells.
<th>	Shows a table **header** cell that typically holds titles or headings.
<td>	Represents a standard **data** cell, holding content or data.
<caption>	Provides a title or description for the entire table.

<thead>	Defines the header section of a table, often containing column labels.
<tbody>	Represents the main content area of a table, separating it from the header or footer.
<tfoot>	Specifies the footer section of a table, typically holding summaries or totals.
<col>	Defines attributes for table **columns** that can be applied to multiple columns at once.
<colgroup>	Groups together a **set of columns** in a table to which you can apply formatting or properties collectively.

Heading content

Heading content, a subset of flow content, defines the title of a section. This definition applies both to sections marked by an explicit sectioning content elements and to those implicitly defined by the heading content itself.

Elements belonging to this category are `<h1>` - `<h6>` and `<hgroup>`.

> **Note:** Though likely to contain heading content, the `<header>` is not heading content itself.

> **Note:** The `<hgroup>` element is not recommended as it does not work properly with assistive technologies. It was removed from the W3C HTML specification prior to HTML 5 being finalized, but is still part of the WHATWG specification and is at least partially supported by most browsers.

Phrasing content

Phrasing content, a subset of flow content, refers to the text and the markup within a document. Sequences of phrasing content make up paragraphs.

Elements belonging to this category are:

- `<abbr>`
- `<audio>`
- `<b>`
- `<bdi>`
- `<bdo>`
- `<br>`
- `<button>`
- `<canvas>`
- `<cite>`
- `<code>`
- `<data>`
- `<datalist>`
- `<dfn>`
- `<em>`
- `<embed>`
- `<i>`
- `<iframe>`
- `<img>`
- `<input>`
- `<kbd>`
- `<label>`
- `<mark>`
- `<math>`
- `<meter>`
- `<noscript>`
- `<object>`
- `<output>`
- `<picture>`
- `<progress>`
- `<q>`
- `<ruby>`
- `<s>`
- `<samp>`
- `<script>`
- `<select>`
- `<slot>`
- `<small>`
- `<span>`
- `<strong>`
- `<sub>`
- `<textarea>`
- `<time>`
- `<u>`
- `<var>`
- `<video>`
- `<wbr>`

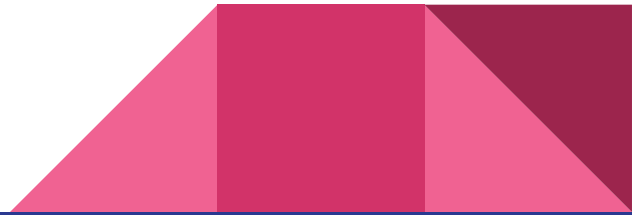
Interactive content

Interactive content, a subset of flow content, includes elements that are specifically designed for user interaction. Elements that belong to this category include:

- `<button>`
- `<details>`
- `<embed>`
- `<iframe>`
- `<label>`
- `<select>`
- `<textarea>`

Some elements belong to this category only under specific conditions:

- `<a>`, if the `href` attribute is present
- `<audio>`, if the `controls` attribute is present
- `<img>`, if the `usemap` attribute is present
- `<input>`, if the type attribute is not in the hidden state
- `<object>`, if the `usemap` attribute is present
- `<video>`, if the `controls` attribute is present



What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.

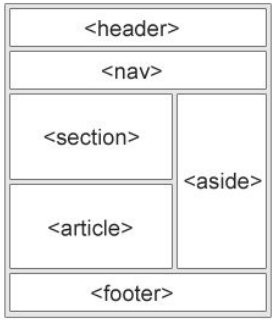
Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

Semantic Elements in HTML

Many web sites contain HTML code like: to indicate navigation, header, and footer.

In HTML there are some semantic elements that can be used to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



<details> ▶ HTML semantic elements

The additional information on the web page that the user can choose to view or hide is specified using the <details> tag in HTML semantics. It serves as a disclosure widget through which users can access more data or exert control. It is used in conjunction with the <summary> tag. Technically, a summary tag is not required, but if you choose to omit it, the browser will use some default content. The <details> element encloses all the text that you want to display or hide, while the <summary> tag includes the section title and summary.

HTML

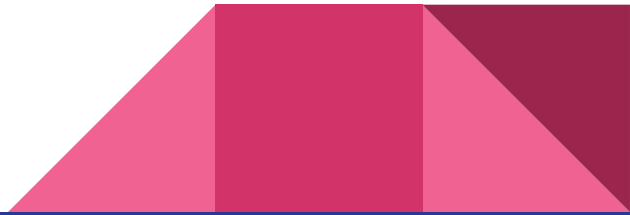
```
<details>
    <summary>Details</summary>
    Something small enough to escape casual notice.
</details>
```

OUTPUT

▼ **Details**

Something small enough to escape casual notice.

`<summary></summary>`	Specifies the summary inside of the <details> </details> semantic tag	Used for the text in the disclosure widget
`<dialog></dialog>`	Represents an interactive dialog box	To create modal or popup dialogs



Drag and Drop

Drag and drop is a very common feature. It is when you "grab" an object and drag it to a different location.

Browser Support

The numbers in the table specify the first browser version that fully supports Drag and Drop.

API					
Drag and Drop	4.0	9.0	3.5	6.0	12.0

```
<!DOCTYPE HTML>
<html>
<head>
<script>
function allowDrop(ev) {
  ev.preventDefault();
}

function drag(ev) {
  ev.dataTransfer.setData("text", ev.target.id);
}

function drop(ev) {
  ev.preventDefault();
  var data = ev.dataTransfer.getData("text");
  ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>

<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>
```

```
<img id="drag1" src="img_logo.gif" draggable="true" ondragstart="drag(event)" width="336" height="69">

</body>
</html>
```

Drag the W3Schools image into the rectangle:

The HTML Geolocation API is used to locate a user's position.

Locate the User's Position

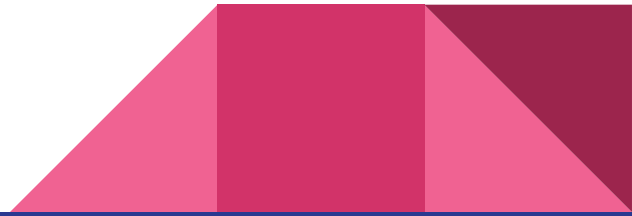
The HTML Geolocation API is used to get the geographical position of a user.

Since this can compromise privacy, the position is not available unless the user approves it.

Using HTML Geolocation

The `getCurrentPosition()` method is used to return the user's position.

The example below returns the latitude and longitude of the user's position:



```
<!DOCTYPE html>
<html>
<body>
<h1>HTML Geolocation</h1>
<p>Click the button to get your coordinates.</p>

<button onclick="getLocation()">Try It</button>

<p id="demo"></p>

<script>
const x = document.getElementById("demo");

function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
  } else {
    x.innerHTML = "Geolocation is not supported by this browser.";
  }
}

function showPosition(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}
  </script>

  </body>
  </html>
```

Thanks