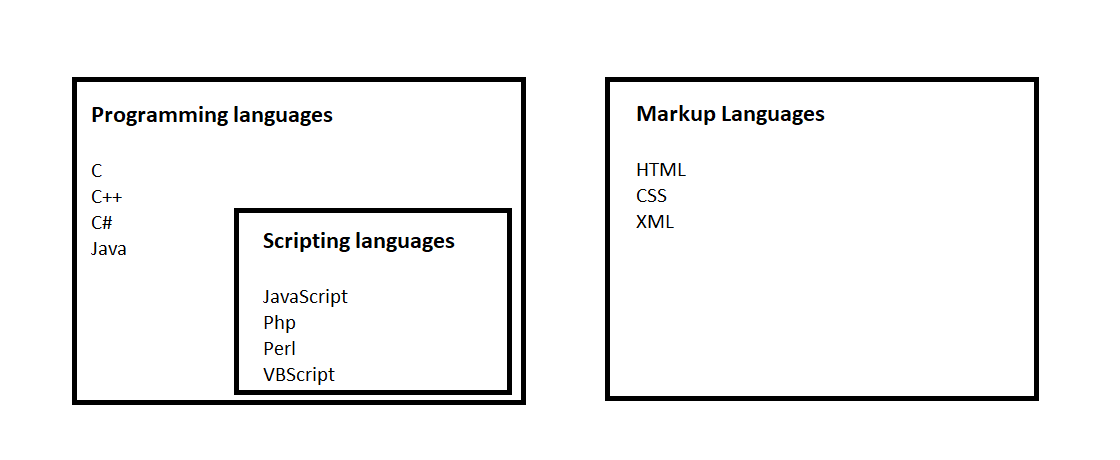
 HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (at the note for computer) text so that a machine can understand it and manipulate text accordingly. Most of the markup (e.g. HTML) languages are human readable. The language uses tags to define what manipulation has to be done on the text. It is used for structuring and presenting the content on the web pages. HTML5 is the fifth version of HTML. Many elements are removed or modified from HTML5. There are many differences between HTML and HTML5 which are discussed below:

|  |  |
| --- | --- |
| **HTML** | **HTML5** |
| **It didn’t support audio and video without the use of flash player support.** | **It supports audio and video controls with the use of <audio> and <video> tags.** |
| **It uses cookies to store temporary data.** | **It uses SQL databases and application cache to store offline data.** |
| **Does not allow JavaScript to run in browser.** | **Allows JavaScript to run in background. This is possible due to JS Web worker API in HTML5.** |
| **Vector graphics is possible in HTML with the help of various technologies such as VML, Silver-light, Flash, etc.** | **Vector graphics is additionally an integral a part of HTML5 like SVG and canvas.** |
| **It does not allow drag and drop effects.** | **It allows drag and drop effects.** |
| **Not possible to draw shapes like circle, rectangle, triangle etc.** | **HTML5 allows to draw shapes like circle, rectangle, triangle etc.** |
| **It works with all old browsers.** | **It supported by all new browser like Firefox, Mozilla, Chrome, Safari, etc.** |
| **<HTML>,<Body> , and <Head> tags are mandatory while writing a HTML code.** | **These tags can be omitted while writing HTML code.** |
| **Older version of HTML are less mobile-friendly.** | **HTML5 language is more mobile-friendly.** |
| **Doctype declaration is too long and complicated.** | **Doctype declaration is quite simple and easy.** |
| **Elements like nav, header were not present.** | **New element for web structure like nav, header, footer etc.** |
| **Character encoding is long and complicated.** | **Character encoding is simple and easy.** |
| **It is almost impossible to get true GeoLocation of user with the help of browser.** | **One can track the GeoLocation of a user easily by using JS GeoLocation API.** |
| **It can not handle inaccurate syntax.** | **It is capable of handling inaccurate syntax.** |
| **Being an older version , it is not fast , flexible , and efficient as compared to HTML5.** | **It is efficient, flexible and more fast in comparison to HTML.** |
| **Attributes like charset, async and ping are absent in HTML.** | **Attributes of charset, async and ping are a part of HTML 5.** |
|  |  |

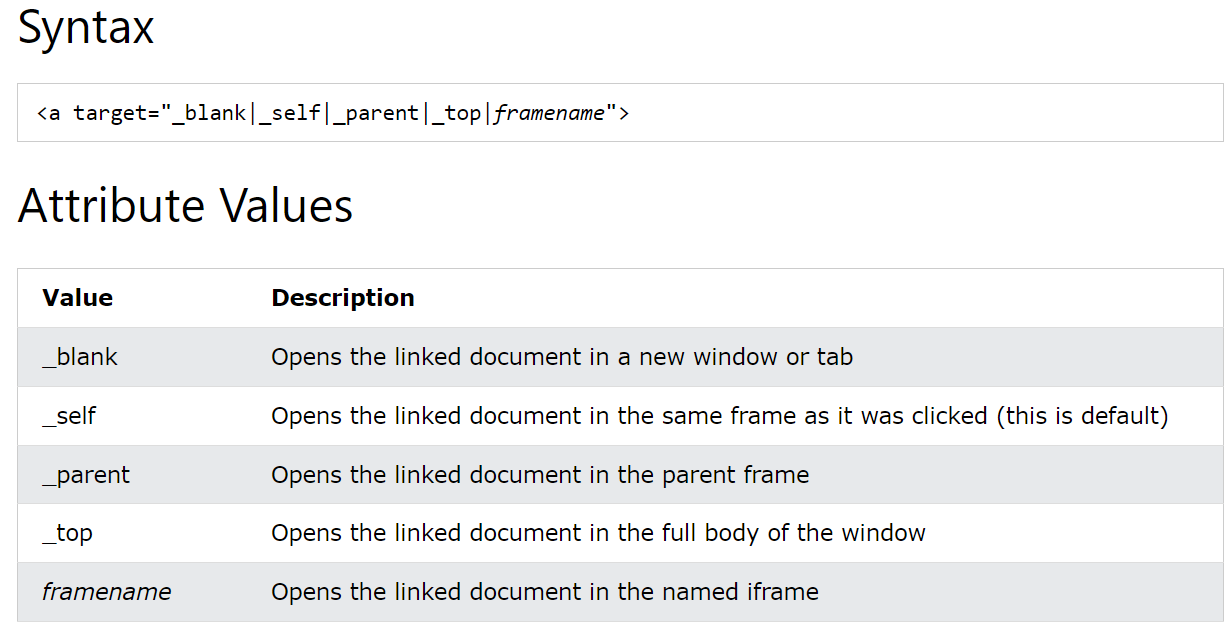


* **Programming language: In simple terms, programming languages are set of instructions or code which tells a computer what it needs to do. So basically, we provide a logic or instruction to the computer to perform some task to get the desired output from** it.
* Programming languages are high-level languages that need to be converted into machine level language because a computer can only understand machine level language or binary language (0 and 1).
* Scripting languages are basically the **subcategory of programming languages** which is used to **give guidance to another program** or we can say **to control another program**, so it also involves instructions.
* It basically **connects one language to one another languages** and doesn’t work **standalone**. **Javascript, PHP, Perl, Python, VBScript** these all are the examples of scripting language. Scripting languages need to be **interpreted** (Scanning the **code line by line**, not like compiler in one go) instead of compiled. There is no scope of compiler in scripting languages. Scripting languages are most widely used to create a website.

**Markup Languages**

* Markup Languages: Markup languages are **completely different from programming languages** and **scripting languages**. Markup languages prepare a **structure for the data** or prepare the look or design of a page.
* These are presentational languages and it doesn’t include any kind of logic or algorithm, for example, HTML. HTML is not asking any kind of question to the computer or it’s not comparing things and it’s not asking any logical question. It’s just used to represent a view inside a web browser. It tells the browser how to structure data for a specific page, layout, headings, title, table and all or styling a page in a particular way.
* So basically it **involves formatting data** or it **controls the presentation of data**. Examples of Markup languages are **HTML or XML**. These languages are most widely used to **design a website**.

**Target Attribute**

****

**Semantic HTML cheat sheet**

There are hundreds of semantic tags available to help describe the meaning of your HTML documents. Below is a cheat sheet with some of the most common ones you’ll use in this course and in your development career.

Sectioning tags

Use the following tags to organize your HTML document into structured sections.

<header>

The header of a content section or the web page. The web page header often contains the website branding or logo.

<nav>

The navigation links of a section or the web page.

<footer>

The footer of a content section or the web page. On a web page, it often contains secondary links, the copyright notice, privacy policy and cookie policy links.

<main>

Specifies the main content of a section or the web page.

<aside>

A secondary set of content that is not required to understand the main content.

<article>

An independent, self-contained block of content such as a blog post or product.

<section>

A standalone section of the document that is often used within the body and article elements.

<details>

A collapsed section of content that can be expanded if the user wishes to view it.

<summary>

Specifies the summary or caption of a <details> element.

<h1><h2><h3><h4><h5><h6>

Headings on the web page. <h1> indicates the most important heading whereas <h6> indicates the least important.

Content tags

<blockquote>

Used to describe a quotation.

<dd>

Used to define a description for the preceding <dt> element.

<dl>

Used to define a description list.

<dt>

Used to describe terms inside <dl> elements.

<figcaption>

Defines a caption for a photo image.

<figure>

Applies markup to a photo image.

<hr>

Adds a horizontal line to the parent element.

<li>

Used to define an item within a list.

<menu>

A semantic alternative to <ul> tag.

<ol>

Defines an ordered list.

<p>

Defines a paragraph.

<pre>

Used to represent preformatted text. Typically rendered in the web browser using a monospace font.

<ul>

Unordered list

Inline tags

<a>

An anchor link to another HTML document.

<abbr>

Specifies that the containing text is an abbreviation or acronym.

<b>

Bolds the containing text. When used to indicate importance use <strong> instead.

<br>

A line break. Moves the subsequent text to a new line.

<cite>

Defines the title of creative work (for example a book, poem, song, movie, painting or sculpture). The text in the <cite> element is usually rendered in italics.

<code>

Indicates that the containing text is a block of computer code.

<data>

Indicates machine-readable data.

<em>

Emphasizes the containing text.

<i>

The containing text is displayed in italics. Used to indicate idiomatic text or technical terms.

<mark>

The containing text should be marked or highlighted.

<q>

The containing text is a short quotation.

<s>

Displays the containing text with a strikethrough or line through it.

<samp>

The containing text represents a sample.

<small>

Used to represent small text, such as copyright and legal text.

<span>

A generic element for grouping content for CSS styling.

<strong>

Displays the containing text in bold. Used to indicate importance.

<sub>

The containing text is subscript text, displayed with a lowered baseline.

<sup>

The containing text is superscript text, displayed with a raised baseline.

<time>

A semantic tag used to display both dates and times.

<u>

Displays the containing text with a solid underline.

<var>

The containing text is a variable in a mathematical expression.

Embedded content and media tags

<audio>

Used to embed audio in web pages.

<canvas>

Used to render 2D and 3D graphics on web pages.

<embed>

Used as a containing element for external content provided by an external application such as a media player or plug-in application.

<iframe>

Used to embed a nested web page.

<img>

Embeds an image on a web page.

<object>

Similar to <embed> but the content is provided by a web browser plug-in.

<picture>

An element that contains one <img> element and one or more <source> elements to offer alternative images for different displays/devices.

<video>

Embeds a video on a web page.

<source>

Specifies media resources for <picture>, <audio> and<video> elements.

<svg>

Used to define Scalable Vector Graphics within a web page.

Table tags

<table>

Defines a table element to display table data within a web page.

<thead>

Represents the header content of a table. Typically contains one <tr> element.

<tbody>

Represents the main content of a table. Contains one or more <tr>elements.

<tfoot>

Represents the footer content of a table. Typically contains one <tr> element.

<tr>

Represents a row in a table. Contains one or more <td> elements when used within <tbody> or <tfoot>. When used within <thead>, contains one or more <th> elements.

<td>

Represents a cell in a table. Contains the text content of the cell.

<th>

Defines a header cell of a table. Contains the text content of the header.

<caption>

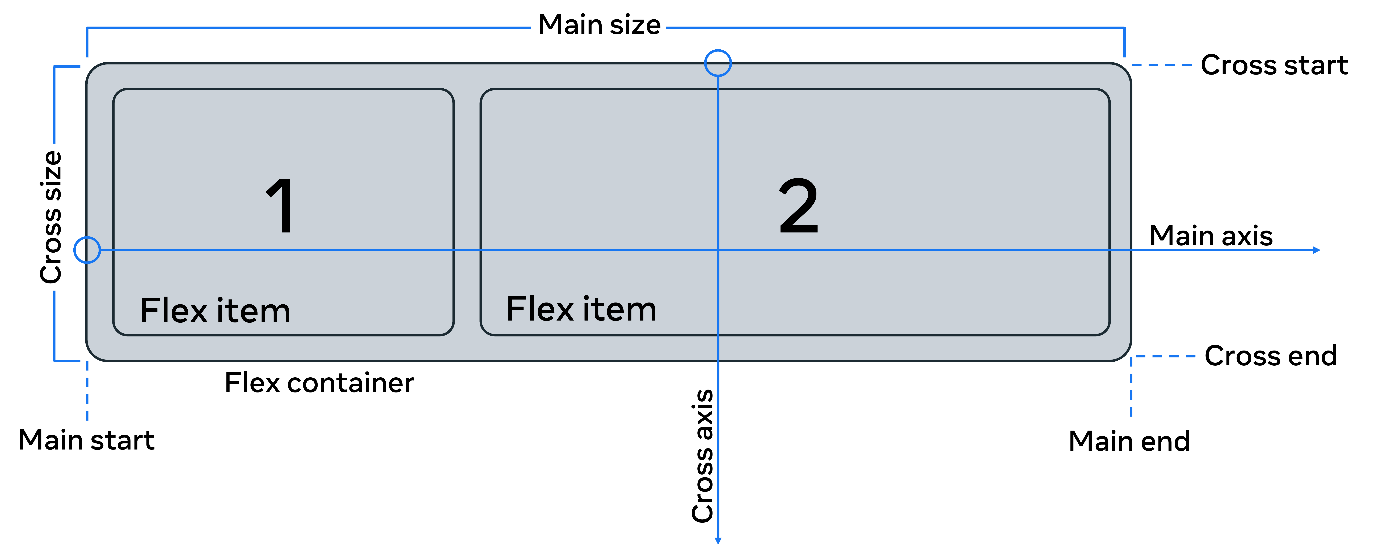
Defines the caption of a table element.

<colgroup>

Defines a semantic group of one or more columns in a table for formatting.

<col>

Defines a semantic column in a table.



Flexbox properties

Original HTML code:

<body>

  <div class="flex-container">

    <div class="box box1">  One..</div>

    <div class="box box2">  Two..</div>

    <div class="box box3">  Three..</div>

    <div class="box box4">  Four..</div>

    <div class="box box5">  Five..</div>

    <div class="box box6">  Six..</div>

    <div class="box box7">  Seven..</div>

  </div>

</body>

.box{

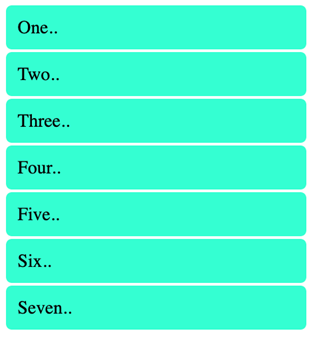
    background-color: aquamarine;

    border-radius: 5px;

    margin: 2px;

    padding: 10px;

}



here are seven div containers inside the HTML file.

The corresponding CSS file contains rules for all seven div tags that have the box class. Note how two class names are given for each of the tags, one that is common among all classes and another independent of it. The style is applied to all the containers.