HTML 5

André Restivo

Index

Introduction

and some History

What is it?

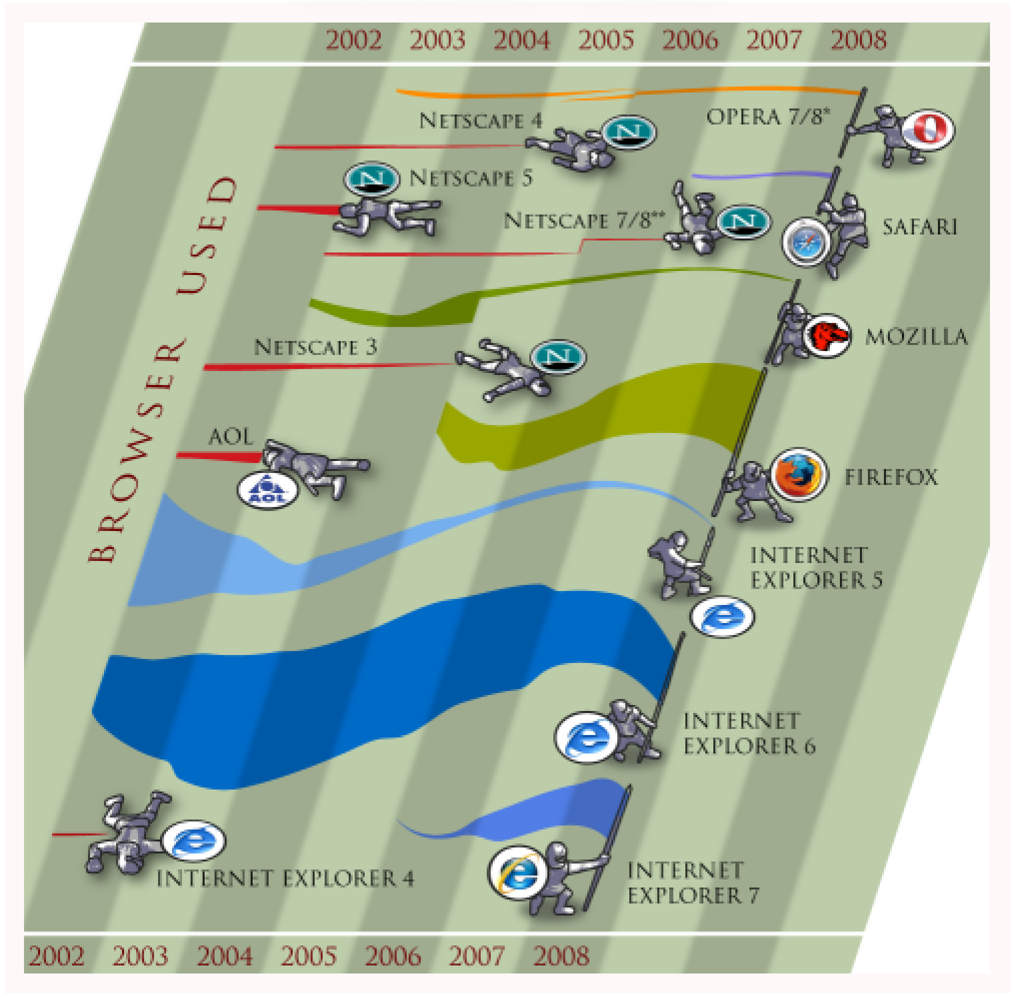
- **H**yper **T**ext **M**arkup **L**anguage;
- Markup language used to create web pages;
- Written using HTML **elements**;
- **Not** for design or presentation;
- All about structure and semantics.

History

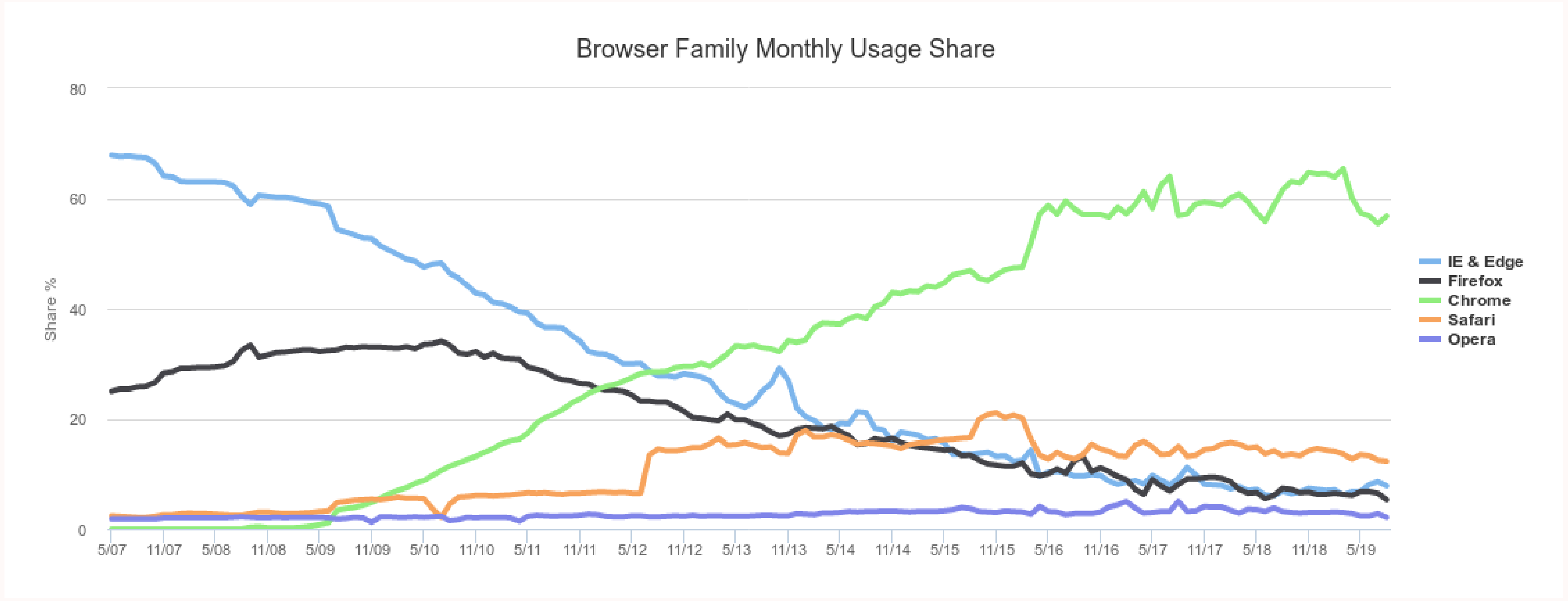
- 1989-92: **HTML 1.0**, Tim Berners-Lee original proposal
- 1993: **HTML+**, Dave Raggett's competing standard
- 1994: **HTML 2.0**, tables, file upload, ... (IETF)
- 1995: Non-standard Netscape features
- 1996: Competing Netscape and Internet Explorer features
- 1996: **HTML 3.2**, W3C standard, the Browser Wars end
- 1997: **HTML 4.0**, stylesheets are introduced
- 1999: **HTML 4.01**, we have a winner!
- 2000: **XHTML 1.0**, an XML version of HTML 4.01
- 2001: **XHTML 1.1**, modularization
- 2008: **HTML 5**, reduces the need for proprietary plug-in based apps

Learn more: http://en.wikipedia.org/wiki/HTML#History

Browser Wars



Browser Share



Source: http://www.w3counter.com/trends

HTML Elements

- HTML is composed of a **tree** of HTML elements;
- Elements can contain other elements and/or **text**;
- They are defined using **tags** and can have **attributes**;
- Browsers display each tag using a **predefined** style that can be changed using CSS.

Tags

Tags start with a **<** and end with a **>** and always contain a name.

They are case insensitive but lowercase is recommended.

```
<html>
```

Most tags come in pairs. An opening tag and a closing tag.

Closing tags have a **/** after the **<**.

```
<html> ... </html>
```

Tag content

The content of a tag is everything between the opening and closing tags.

```
<p>Some content</p>
```

Some tags don't have content and do not need to be closed.

```
<br>
```

Tags can have attributes. Some are optional and some are mandatory.

```
<img src="dog.png">
```

Resources

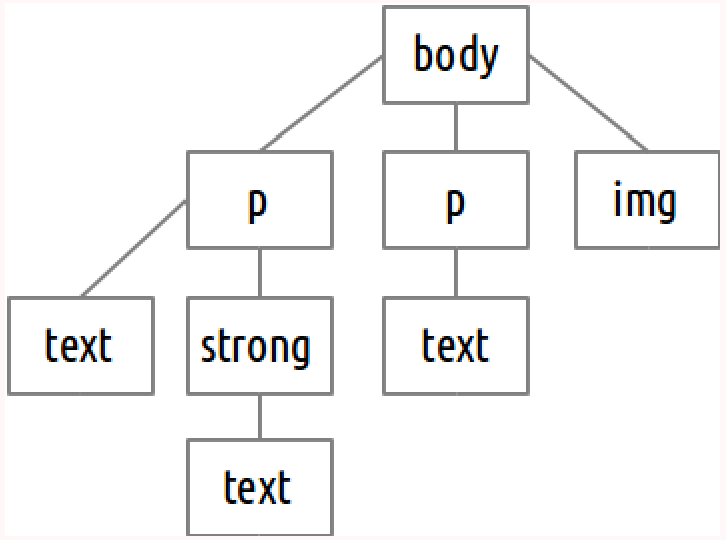
- References:
  - W3C Standard
  - WHATWG Living Standard
  - Mozilla Developer Network (MDN) Reference
- Books:
  - Dive into HTML 5
- Tutorials:
  - https://webplatform.github.io/docs/html/tutorials/
  - http://www.htmldog.com/guides/html/
  - http://html5tutorial.info/

Document

Tree Structure

- HTML elements form a **tree** like structure;
- Some HTML elements can have children. Those have **start** and **end** tags;
- HTML elements that do not allow children only have an opening tag.

Example



```
<body>
  <p id="first">Some <strong>text</strong></p>
  <p>Some other text</p>
  <img src="image.png">
</body>
```

Basic Document

- All HTML documents have a **html** root tag.
- They all contain a **head** and **body** sections.
- The head section *must* contain a **title** tag.
- The html tag *should* contain a **lang** attribute (e.g. "pt-PT" or "en-US").

```
<html lang="en-US">
<head>
 <title>Title</title>
</head>
<body>
</body>
</html>
```

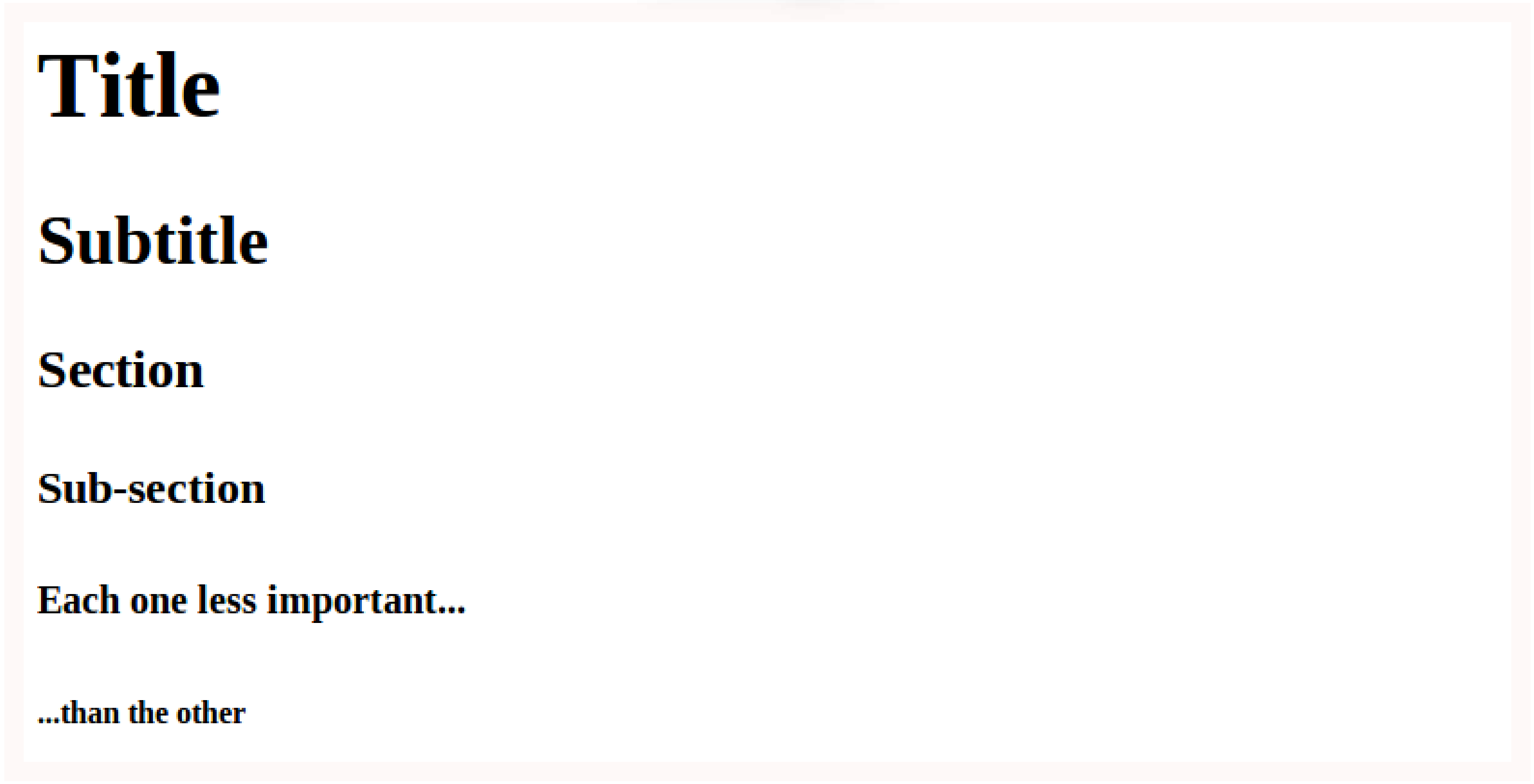
Basic Tags

Headings

- HTML implements six levels of document headings. **`<h1>`** is the most important and **`<h6>`** is the least.
- A heading element briefly describes the topic of the section it introduces.

```html
<h1>Title</h1> <!-- only one per document -->
<h2>Subtitle</h2> <!-- this is a comment btw -->
<h3>Section</h3>
<h4>Sub-section</h4>
<h5>Each one less important...</h5>
<h6>...than the other</h6>
```

Example



Title

Subtitle

Section

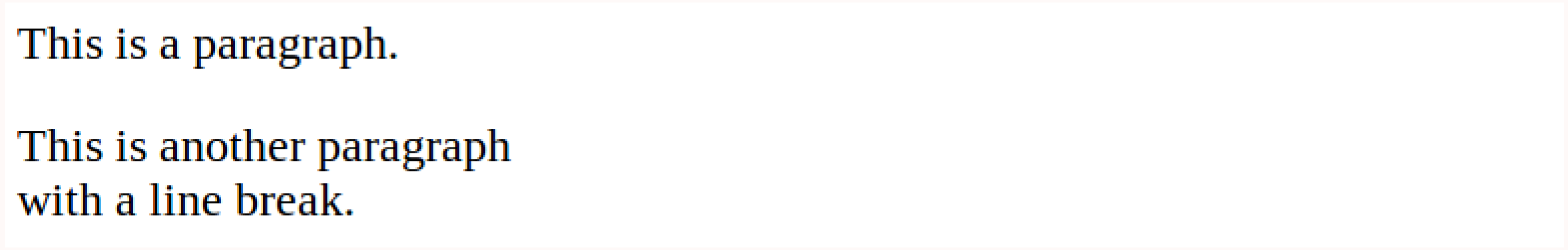
Sub-section

Each one less important...

...than the other

Paragraphs and Line Breaks

```html
<p>This is a paragraph.</p>
<p>
This is another paragraph<br> <!-- empty tag -->
with a line break.
</p>
```



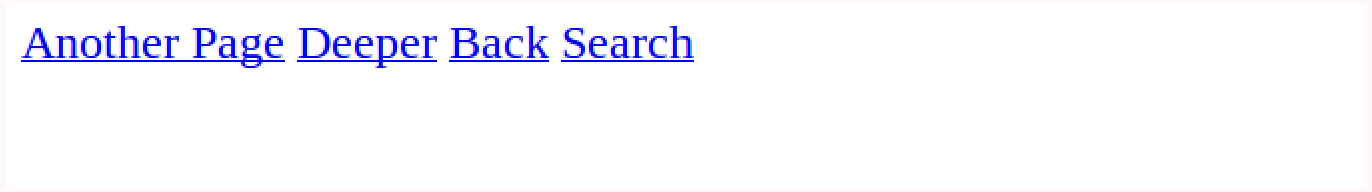
This is a paragraph.

This is another paragraph
with a line break.

Anchor

Anchors (or links) can be relative or absolute.

```html
<a href="anotherpage.html">Another Page</a>
<a href="somewhere/deeper.html">Deeper</a>
<a href="../start.html">Back</a>
<a href="http://www.google.com">Search</a>
```


Another Page Deeper Back Search

Images

The **alt** attribute is mandatory and represents an alternative image description for browsers incapable of showing images.

> *Omitting this attribute indicates that the image is a key part of the content, but no textual equivalent is available. Setting this attribute to the empty string indicates that this image is not a key part of the content; non-visual browsers may omit it from rendering.*

The **width** and **height** attributes are optional and should not be used to resize images on the fly.

```html
<img src="dog.png" alt="A dog" width="300" height="200">
```

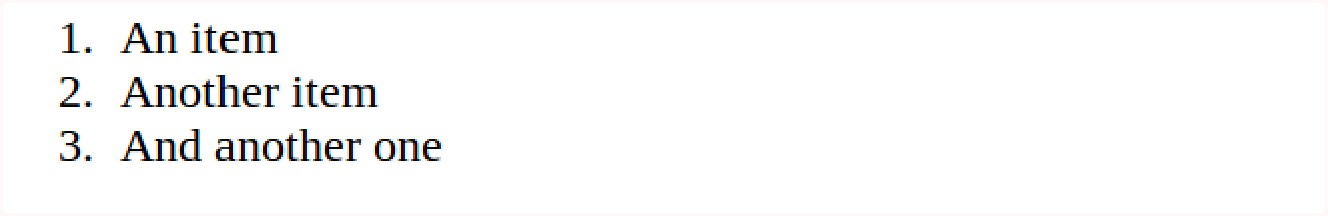
Common Attributes

- **hidden**: Specifies that an element is not yet, or is no longer, relevant.
- **accesskey**: Specifies a shortcut key to activate/focus an element.

Lists

Ordered Lists

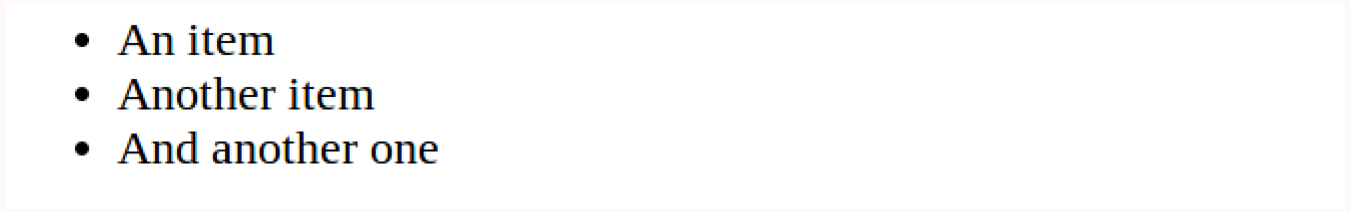
```html
<ol>
  <li>An item</li>
  <li>Another item</li>
  <li>And another one</li>
</ol>
```



1. An item
2. Another item
3. And another one

Unordered Lists

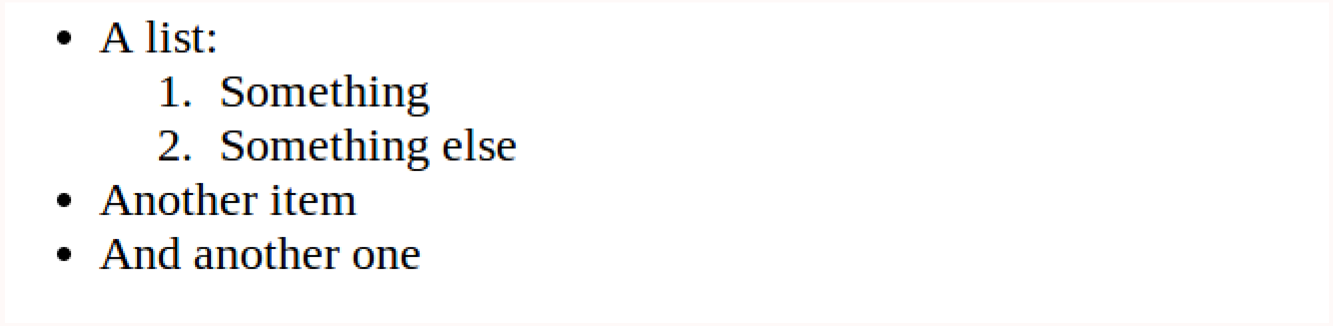
```
<ul>
  <li>An item</li>
  <li>Another item</li>
  <li>And another one</li>
</ul>
```



- An item
- Another item
- And another one

Nested Lists

```
<ul>
  <li>A list:
    <ol>
      <li>Something</li>
      <li>Something else</li>
    </ol>
  </li>
  <li>Another item</li>
  <li>And another one</li>
</ul>
```

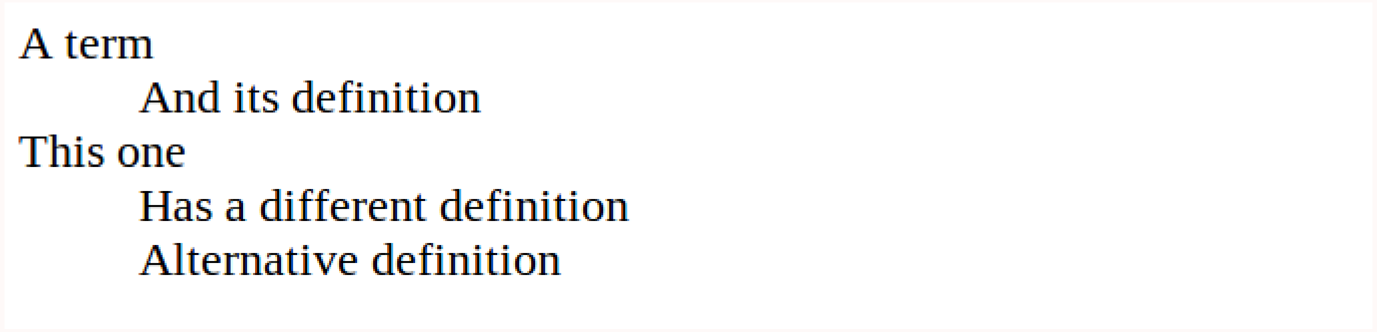


- A list:
    1. Something
    2. Something else
- Another item
- And another one

Description Lists

- Define terms and descriptions.
- A term can have several descriptions.
- Several terms can have the same description.

```html
<dl>
  <dt>A term</dt>
  <dd>And its definition</dd>
  <dt>This one</dt>
  <dd>Has a different definition</dd>
  <dd>Alternative definition</dd>
</dl>
```



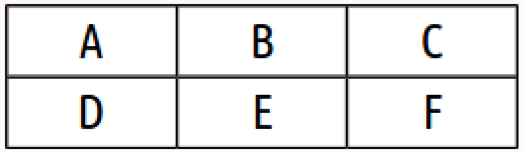
```
A term
        And its definition
This one
        Has a different definition
        Alternative definition
```

Tables

Rows and Data

A table is organized using rows (tr) that contain data cells (td).

```
<table>
  <caption>Just some letters</caption>
  <tr>
    <td>A</td><td>B</td><td>C</td>
  </tr>
  <tr>
    <td>D</td><td>E</td><td>F</td>
  </tr>
</table>
```



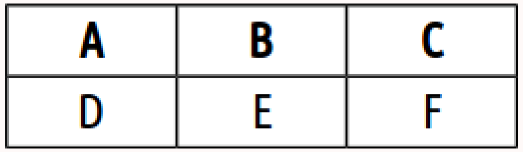
A	B	C
D	E	F

A table can have an optional *caption*.

Headers

Some data cells can be headers (**th** instead of **td**)

```
<table>
  <tr>
    <th scope="col">A</th><th scope="col">B</th><th scope="col">C</th>
  </tr>
  <tr>
    <td>D</td><td>E</td><td>F</td>
  </tr>
</table>
```



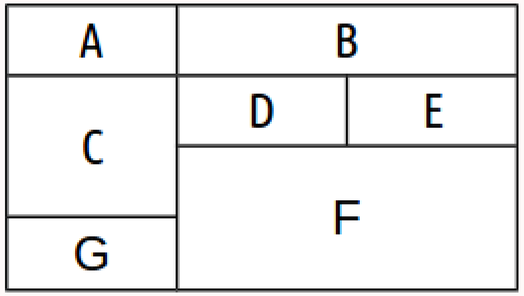
A	B	C
D	E	F

Table headers can have an optional **scope** attribute that indicates the cells the attribute relates to. Values for this attribute can be **col** or **row**.

Cell Merging

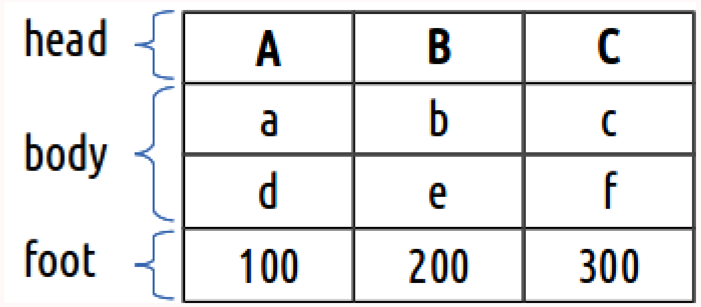
Cells can be merged horizontally or vertically

```
<table>
  <tr>
    <td>A</td><td colspan="2">B</td>
  </tr>
  <tr>
    <td rowspan="2">C</td><td>D</td><td>E</td>
  </tr>
  <tr>
    <td colspan="2" rowspan="2">F</td>
  </tr>
  <tr>
    <td>G</td>
  </tr>
</table>
```



Sections

```
<table>
  <thead>
    <tr>
      <th>A</th><th>B</th><th>C</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td>100</td><td>200</td><td>300</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>a</td><td>b</td><td>c</td>
    </tr>
    <tr>
      <td>d</td><td>e</td><td>f</td>
    </tr>
  </tbody>
</table>
```



A	B	C
a	b	c
d	e	f
100	200	300

Column Groups

So that we don't have to repeat the same information for each cell in a column, we can define column groups using the **colgroup** and **col** tags.

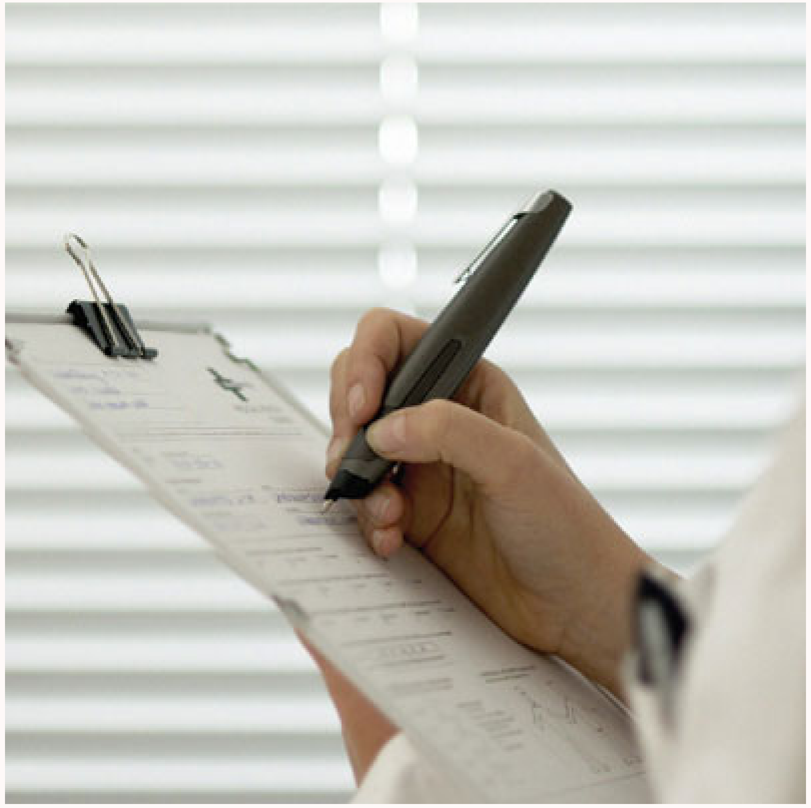
```
<table>
  <colgroup>
    <col span="2" class="first">
    <col>
  </colgroup>
  <tr>
    <td>A</td><td>B</td><td>C</td>
  </tr>
  <tr>
    <td>D</td><td>E</td><td>F</td>
  </tr>
</table>
```

Mainly used to set the *class* of each column. We will talk about classes later on.

Forms

What are they?

Forms allow users to enter data that is sent to a server for processing



Action and Method

The **form** tag defines a form that can contain controls.

```
<form action="save.php" method="get">
  <!-- form controls go here -->
</form>
```

- **action**: the web page that receives and processes the form results
- **method**: either **get** (values are sent in the URL) or **post** (values are sent inside the HTTP header)

Form Controls

Four main types of form controls:

- **input**: Several types of user editable fields;
- **textarea**: A big editable text field;
- **select**: A dropdown list.
- **button**: A generic button.

Input

An input field can vary in many ways, depending on the **type** attribute.

```
<input type="text" name="address">

<input type="password" name="password">

<input type="email" name="email">
```

Input Common Attributes

- **type**: the type of the input
- **name**: name of the field to be passed to the action
- **placeholder**: hint for the user
- **autocomplete**: value of the control can be automatically completed by the browser (on/off)
- **readonly**: input value cannot be modified (boolean)
- **required**: input must be filled out (boolean)
- **disabled**: input is disabled (boolean)

Boolean attributes: If the attribute is present, its value must either be the empty string or a value that is an ASCII case-insensitive match for the attribute's canonical name, with no leading or trailing whitespace.

```
<input type="text" name="address" required="required" disabled>
```

Learn more: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input

Text Inputs

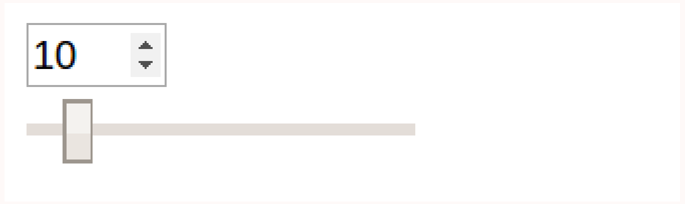
```html
<input type="tel" name="phone" value="555-555-555">
```

- **type**:
  - **text**: text input with no constraints
  - **password**: characters are not shown
  - **tel**: input value is a telephone number
  - **search**: input value is used to perform a search
  - **url**: input value is an URL
  - **email**: input value is used an e-mail
- **value**: the initial value

Number Inputs

```
<input type="number" value="10" min="0" max="100" step="5">
```

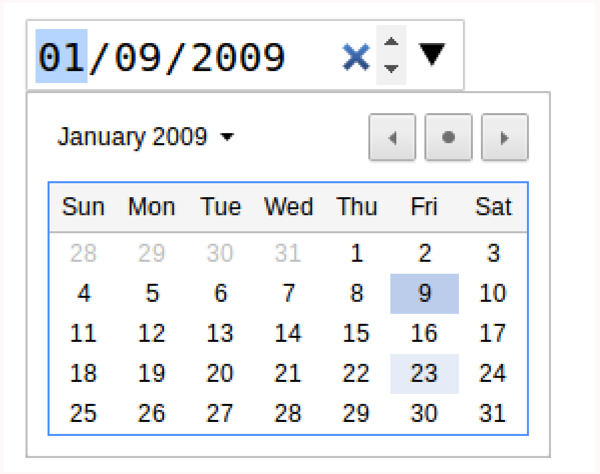
- **type**:
  - **number**: a precise control for setting a number
  - **range**: imprecise control for setting a number
- **value**: the initial value
- **min**: the minimum value
- **max**: the maximum value
- **step**: limits the increments at which a value can be set



Date/Time Inputs

```
<input type="date" value="2009-01-09">
```

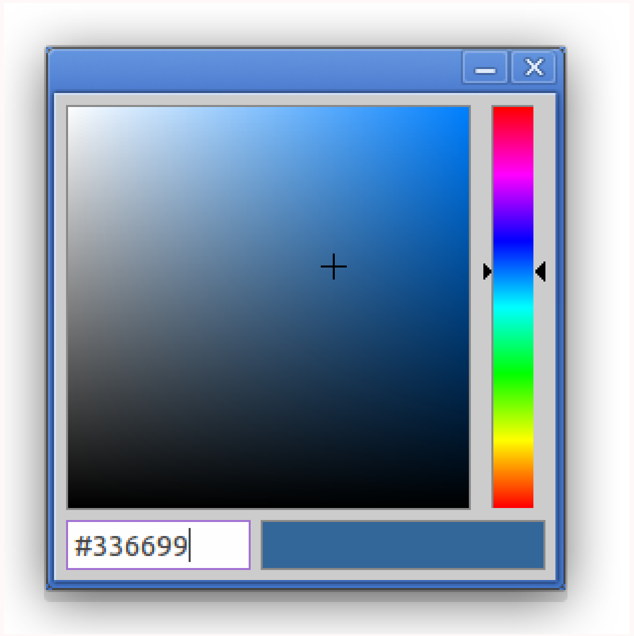
- **type**:
    - **time**: control to select a time of the day
    - **date**: control to select a date
    - **datetime**: control to select a time in a certain day
    - **week**: control to select a week
- **value**: the initial value according to: RFC3339 (obligatory XKCD)



Color Input

```
<input type="color" value="#336699">
```

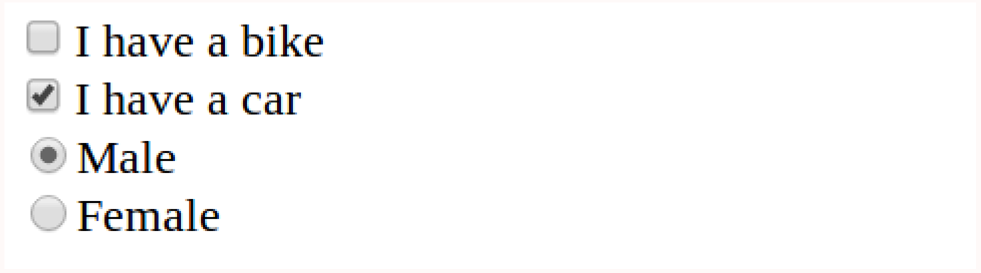
- **type**: **color**: control to select a color
- **value**: initial color in hexadecimal format



Checkbox and Radio

```
<input type="checkbox" name="vehicle" value="Bike">Ride a bike
<input type="checkbox" name="vehicle" value="Car" checked="checked" >Drive a car
<!-- user can select several -->

<input type="radio" name="gender" value="male" checked="checked">Male
<input type="radio" name="gender" value="female">Female
<!-- only one can be selected             -->
<!-- name must be the same for each group -->
```



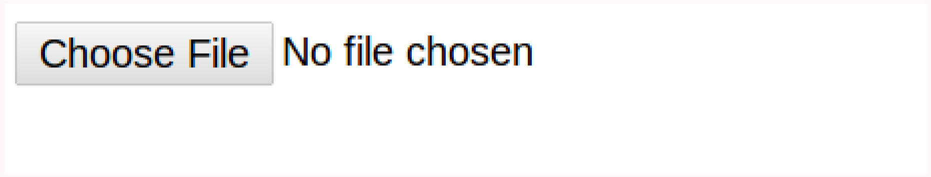
I have a bike
[✓ I have a car]
[✓ Male]
Female

File Upload

Allows file uploading for storing or processing

```
<form action="upload_file.php" method="post" enctype="multipart/form-data">
  <input type="file" name="file">
</form>
```

To use file uploading in a form, *method* must be **post** and *enctype* must be **multipart/form-data**



Hidden Input

```
<input type="hidden" name="?">
```

The same as a text field but it does not show on the browser. We'll find what these are used for later...

Submit

- A **button** that allows the user to submit the form for processing.
- The **value** contains the text to be used for the submit button. A multilingual default will be used if left blank.

```
<input type="submit" value="Send">
```

The form will be submitted using the *method* and *action* defined in the **form** tag.

Button

An alternative way of creating a button inside a form is by using the **button** tag.

```
<button formaction="login.php" formmethod="post">Login</button>
<button formaction="register.php" formmethod="post">Register</button>
```

This way you can have different buttons with **different** actions and methods.

Select

Dropdown boxes that allow users to select options from a list.

```
<select name="fruit">
  <option value="orange">Orange</option>
  <option value="banana" selected>Banana</option>
  <option value="tomato">Tomato</option>
  <option value="apple">Apple</option>
</select>
```

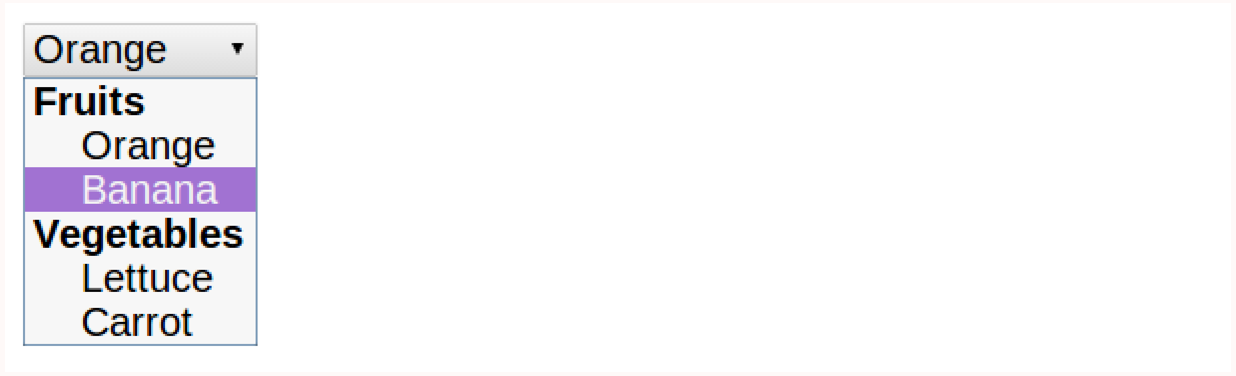
- **value**: The value sent to the server
- **content**: The text presented to the user
- **selected** (boolean): The option is the selected one

Tomato
Orange
Banana
Tomato
Apple

Option Groups

Options in select controls can be grouped to make selecting them easier.

```
<select name="food">
  <optgroup label="Fruits">
    <option value="orange">Orange</option>
    <option value="banana" selected>Banana</option>
  </optgroup>
  <optgroup label="Vegetables">
    <option value="lettuce">Lettuce</option>
    <option value="carrot">Carrot</option>
  </optgroup>
</select>
```



Select Attributes

- **name**: name of the field to be passed to the action
- **multiple**: allow multiple selections (boolean)
- **required**: field must be filled out (boolean)
- **disabled**: input is disabled (boolean)

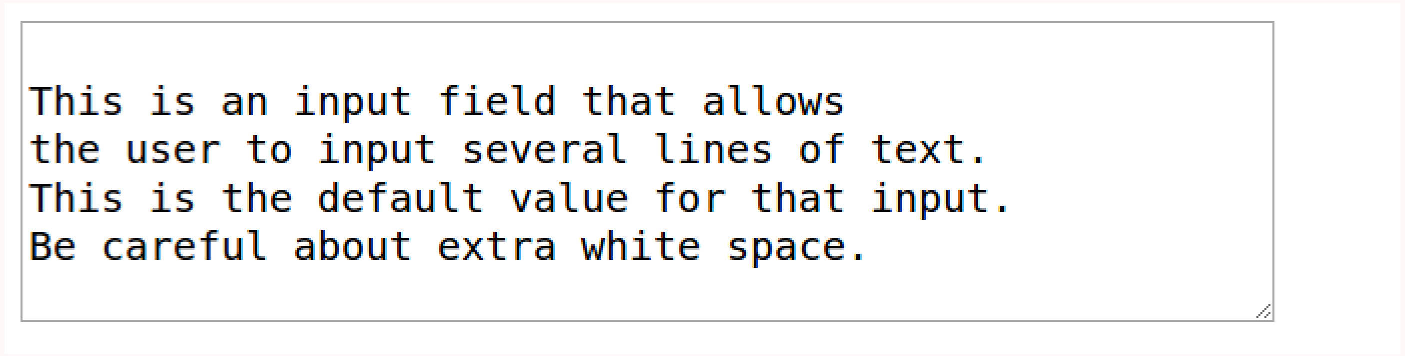
```
<select name="food" multiple="multiple" required>

</select>
```

Text Area

A text input field for larger texts.

```html
<textarea name="description" rows="4" cols="50">
This is an input field that allows
the user to input several lines of text.
This is the default value for that input.
Be careful about extra white space.
</textarea>
```



```
This is an input field that allows
the user to input several lines of text.
This is the default value for that input.
Be careful about extra white space.
```

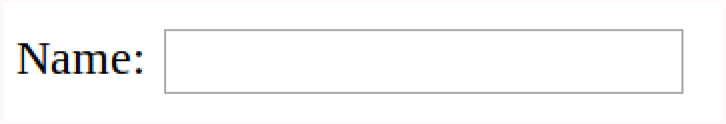
Text areas also allow the common attributes **name**, **disabled**, **readonly** and **required**.

Label

- Allows the association between a **label** and its corresponding **input**.
- Clicking the **label** activates the **input**.
- Important for disabled people.
- Two ways of using it:

```html
<label for="id_name">Name:</label>
<input type="text" name="name" id="id_name">
```

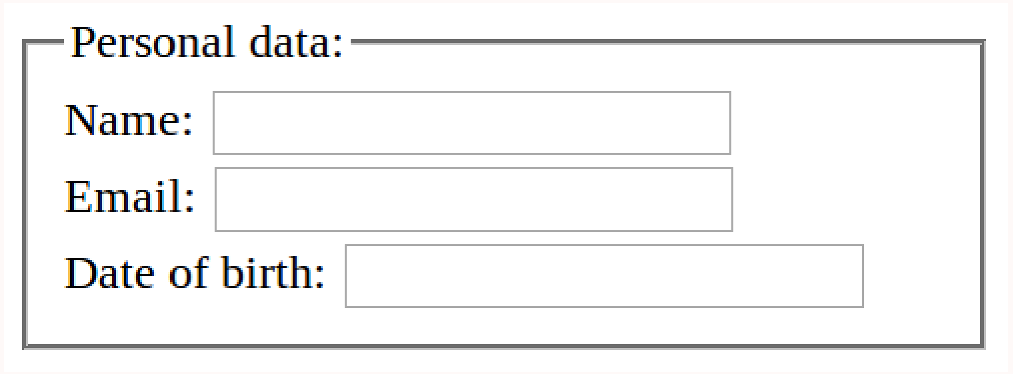
```html
<label>Name:
  <input type="text" name="name">
</label>
```



Field Set

- Allows grouping inputs in larger forms.
- The **legend** tag contains the title of the group.

```
<form>
  <fieldset>
    <legend>Personal data:</legend>
    <label>Name: <input type="text"></label>
    <label>Email: <input type="text"></label>
    <label>Date of birth: <input type="text"></label>
  </fieldset>
</form>
```

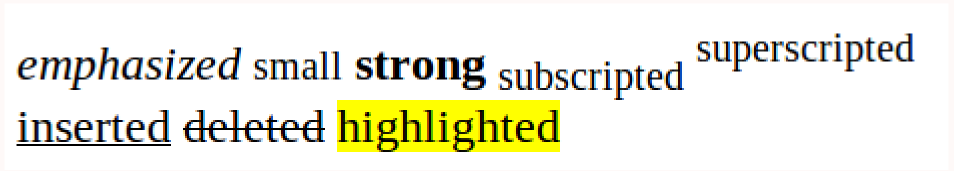


Text Tags

Text tags define portions of text as having a special meaning.

Formatting Tags

```html
<em>emphasized</em> <!-- Defines emphasized text -->
<small>small</small> <!-- Defines smaller text -->
<strong>strong</strong> <!-- Defines important text -->
<sub>subscripted</sub> <!-- Defines subscripted text -->
<sup>superscripted</sup> <!-- Defines superscripted text -->
<ins>inserted</ins> <!-- Defines inserted text -->
<del>deleted</del> <!-- Defines deleted text -->
<mark>highlighted</mark> <!-- Defines marked/highlighted text -->
```



emphasized small **strong** subscripted superscripted
inserted ~~deleted~~ highlighted

Output Tags

```
<code>...</code> <!-- Defines computer code text -->
<kbd>...</kbd> <!-- Defines keyboard text  -->
<samp>...</samp> <!-- Defines sample computer code -->
<var>...</var> <!-- Defines a variable -->
<pre>...</pre> <!-- Defines preformatted text -->
```

Special Tags

```
<abbr></abbr> <!-- Defines an abbreviation or acronym -->
<address></address> <!-- Defines contact information for someone -->
<time></time> <!-- Defines a time of the day -->
<progress></progress> <!-- Defines a progress of a task -->
<bdo></bdo> <!-- Defines the text direction -->
<blockquote></blockquote>  <!-- Quoted from another source -->
<q></q> <!-- Defines an inline (short) quotation -->
<cite></cite> <!-- Defines the title of a work -->
<dfn></dfn> <!-- Defines a definition term -->
```

Character Entities

Character Entities

A given character encoding may not be able to express all characters of the document character set.

Some characters might have some special meaning (<, >, " and &) and be confused by the browser as markup.

Character references in HTML may appear in two forms:

- Numeric character references (either decimal or hexadecimal).
- Named character entity references.

Character Entities

Character entities always start with a **&** and end with a **;**

For example, the ampersand (&):

- Decimal character: &#38;
- Hexadecimal character: &#x26;
- Named character entity: &amp;

Most important character entities:

- Less than sign (<): &lt;
- Greater than sign (>): &gt;
- Ampersand (&): &amp;
- Double quote sign ("): &quot;
- Non-breaking space ( ): &nbsp;

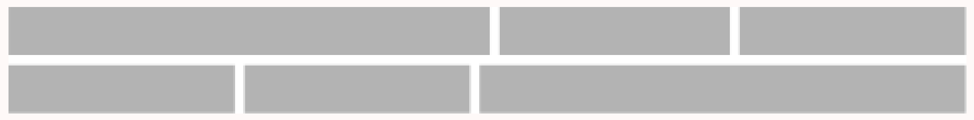
Other character entities | Character entity search

Elements

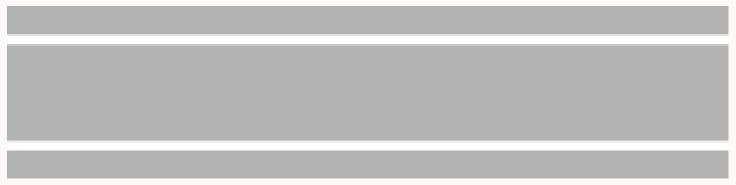
Elements

There are two major types of HTML elements according to the way they display on the browser:

- **Inline** elements occupy only the space they need and don't force line changes. Example: strong, a, ...



- **Block** elements, by default, use all the horizontal space they can get and force a line change before and after themselves. Example: p, h1, ...



Id and Class

The **id** and **class** attributes are used to easily identify a tag for manipulation (using javascript) or styling (using CSS).

A HTML document **cannot** have two elements with the same **id**:

```
<img id="logo" src="logo.png>
```

A HTML element can have more than one **class** (separated by whitespace).

```
<p class="first important">Some text</p>
```

We can create an anchor to an element with a specific **id** within a page:

```
<a href="anotherpage.html#introduction">Another page</a>
```

You can think of the **id** as the name of the element and the **class** as its type.

Span

Span is a generic **inline** tag that can be used, for example, to mark specific parts of text:

```html
<p>This book has been written
by <span class="author person">Arthur C. Clark</span>.</p>
```

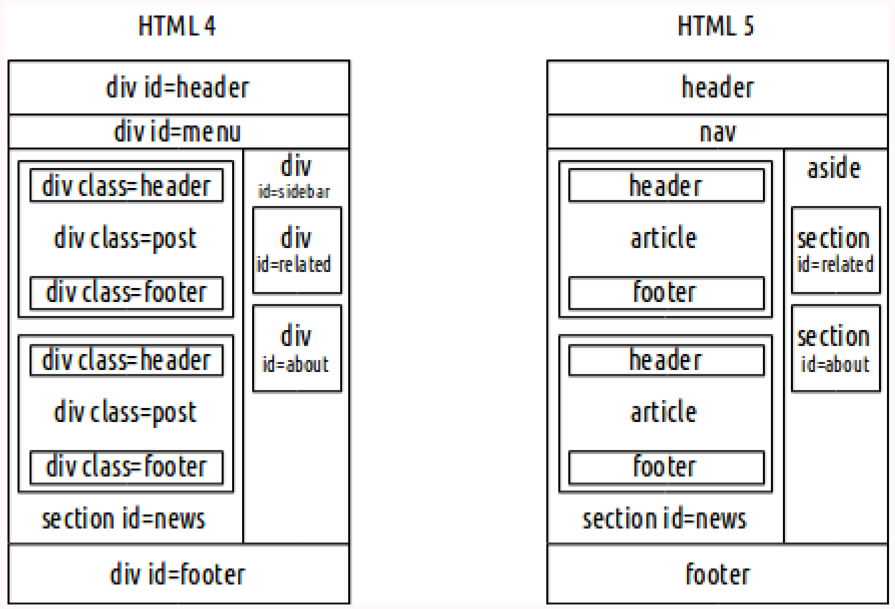
Div

Div is a generic **block** tag that can be used to define sections of a website:

```
<div id="menu">
    <ul>
        <li><a>Home</a></li>
        <li><a>Contacs</a></li>
        <li><a>Register</a></li>
    </ul>
</div>
```

Semantic Elements

- Semantic tags tell the browser something about what's inside them.
- Tags like **div** and **span** tell nothing!
- HTML 5 brings a whole new set of semantic tags to replace commonly used div blocks.



Semantic Elements

header

Represents a group of introductory or navigational aids. It may contain some heading elements but also other elements like a logo, wrapped section's header, a search form, and so on. Many different elements can contain the **header** tag: body, section, article, form, ...

nav

Represents a section of a page that links to other pages or to parts within the page: a section with navigation links.

aside

Represents a section of the page with content connected tangentially to the rest, which could be considered separate from that content. These sections are often represented as sidebars or inserts.

Semantic Elements

section

Represents a generic section of a document, i.e., a thematic grouping of content, typically with a heading (**header**).

article

Represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable, e.g., in syndication. Each *article* should be identified, typically by including a heading (**header**) as a child.

footer

Represents a footer for its nearest sectioning content or sectioning root element. A footer typically contains information about the author of the section, copyright data or links to related documents. Many different elements can contain the **footer** tag: body, section, article, form, ...

Example

```html
<html>
  <head><title>Science News</title></head>
  <body>
    <header>
      <img src="logo.png">
      <form action="search.php">...</form>
      <nav>
        <ul>
          <li><a href="other.php">Other</a></li>
          ...
        <ul>
      </nav>
    </header>
    <div id="main">
      <section id="news">
        <h2>News</h2>
        <article>
          <h3>Great news everyone!</h3>
          <p>...</p>
          <footer>Author: Hubert J. Farnsworth</footer>
        </article>
      </section>
    </div>
    <footer>
      Copyright: Mad Scientists News 2018
    </footer>
  </body>
</html>
```

Media

Canvas

A **canvas** is an empty rectangle that can be used to draw on the fly using *javascript*.

```
<canvas width="?" height="?"></canvas>
```

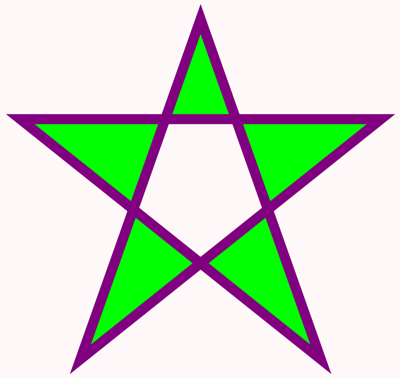
Some examples

SVG

- **S**calable **V**ector **G**raphics.
- SVG images can be created and edited with any text editor.
- SVG images can be searched, indexed, scripted, and **compressed**.
- SVG images are **scalable**.
- SVG images can be printed with high quality at **any resolution**.
- SVG images are **zoomable** without degradation.

SVG Example

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1" width="200" height="200">
  <polygon
        points="100,10 40,180 190,60 10,60 160,180"
        style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;"
  >
</svg>
```



Other Media Tags

HTML 5 also includes specific tags for:

- **audio**: defines sound, such as music or other audio streams
- **video**: specifies video, such as a movie clip or other video streams
- **source**: specify multiple media resources for media elements
- **track**: text tracks for video and audio elements

Learn more: Using HTML5 Audio and Video

Document Type

Document Type

- The Document Type declaration is **not** an HTML tag;
- It must be the first thing on your document;
- It tells the browser which HTML version you are using;
- For HTML 5 just use:

```
<!DOCTYPE html>
```

- HTML 4.01 & XHTML 1.0 (strict):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
                      "http://www.w3.org/TR/html4/strict.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
                      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Metadata

You can define metadata for your document inside the head tag.

Meta Content

```
<head>
  <meta name="?" content="?">
</head>
```

- **name:**
  - **application-name**, defining the name of the web application running in the webpage.
  - **author**, defining, in a free format, the name of the author of the document.
  - **description**, containing a short and accurate summary of the content of the page.
  - **generator**, containing, in a free format, the identifier to the software that generated the page.
  - **keywords**, containing, as strings separated by commas, relevant words associated with the content of the page.
- Learn more: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta

Character Set

Defining the character set used by the document.

```html
<head>
  <meta charset="utf-8">
</head>
```

- **UTF-8** Character encoding for Unicode (recommended)
- **ISO-8859-1** Character encoding for the Latin alphabet

Validation

http://validator.w3.org/