Prework HTML

What is HTML

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Homepage</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

What you see above is a simple example of code. It is HTML, which is a markup language used to describe page content.
HTML is not a programming language. It does not contain any computational or logical expressions, and is only used to describe the structure of a document, i.e. to create various titles, links, text paragraphs, tables, lists, etc. With the help of HTML we can use plain text to make links to other pages, insert graphics on a page.

Information - Three technologies

Those three technologies that you can see at the top of the page - HTML, CSS and JavaScript, are an inseparable family. **HTML** describes a document, creates its logical structure, **CSS** gives the document a certain look, and **JavaScript** - adds dynamics and interaction to it.

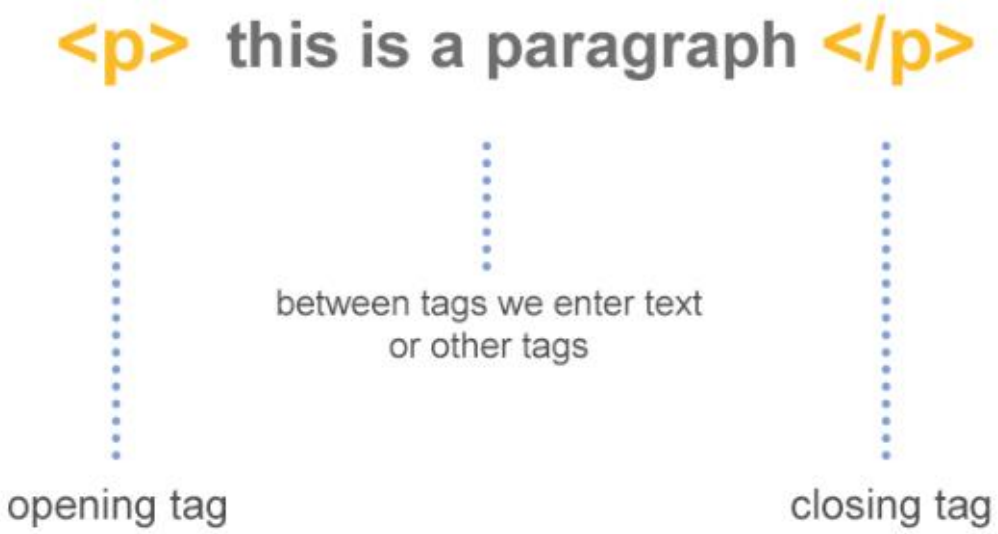
A bit of history

The HTML language was created by the physicist Tim Berners-Lee.
For many years, new, subsequent versions were created. We will create our pages in **HTML5**. Although its first draft was published by W3C (organization responsible for the development of this language) in 2008, it was officially recommended in 2014.
Version five not only introduced many new elements and removed obsolete ones, but also defined new page divisions, simplified many older parts (such as the declaration of a document to be found in future chapters), removed the need to close tags, etc.
You can say that after many years of development of new versions, the approach that has ultimately won is that the language that millions of people work in can be simple and enjoyable to use.

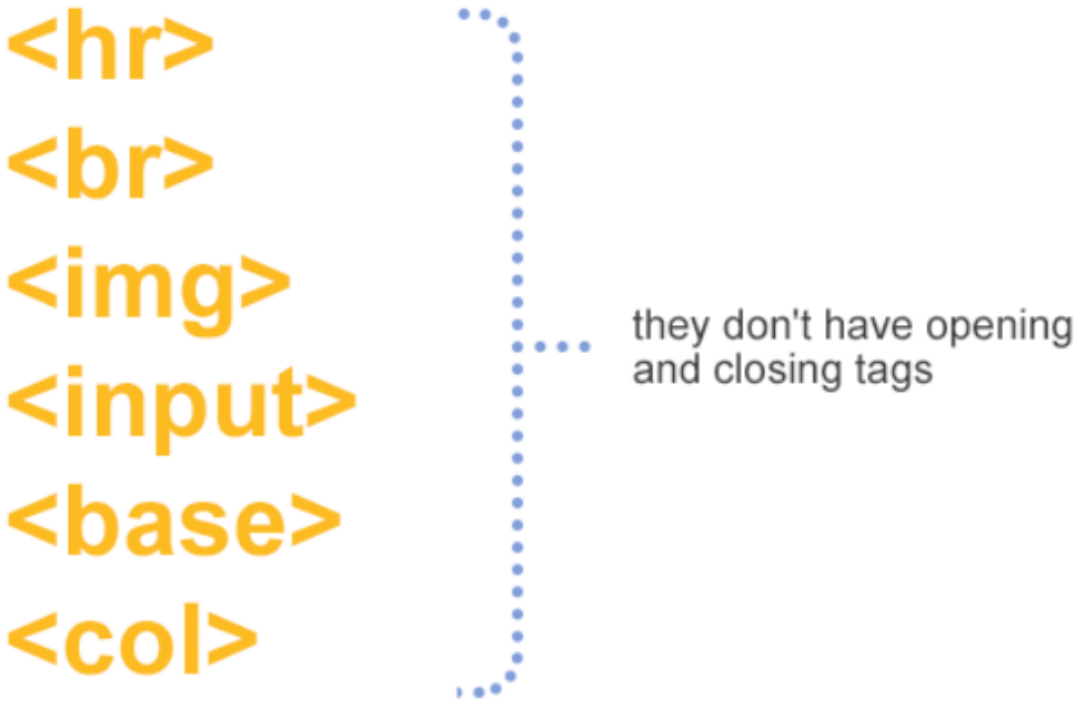
If you are interested in this topic, you can learn more at https://en.wikipedia.org/wiki/HTML.

HTML Tags

HTML language consists of tags. Each tag has a meaning and is used to represent something different. For example, the **<p>** tag is used to represent a piece of text, the **<h1>** tag is used to create a title on a page, and the **<a>** tag is used to create links. We will use tags very often throughout the course.
The basic structure of a tag is shown in the graphic below:



Most tags consist of an opening tag and a closing tag. Between such an opening and closing, there may be an additional document structure placed - text or other tags.
The second group of tags are the **empty** tags, which have no opening or closing and therefore cannot have any content inside.



Closing a tag in HTML4

In HTML4 all tags had to be closed - even the empty ones. This meant that when writing empty element's code, one had to add **/** as its end.

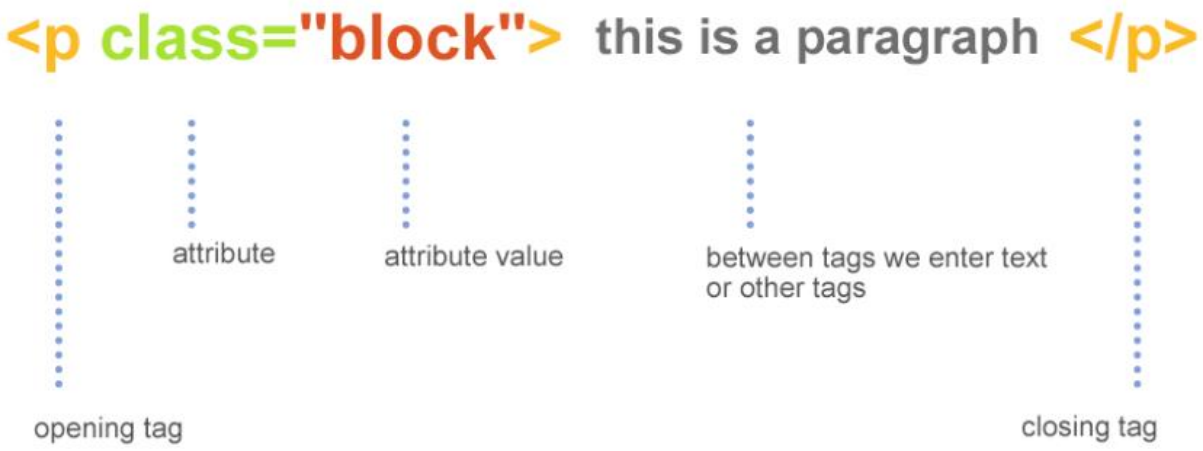
```
<!-- in HTML4 -->
<br />
<img />
<meta />
<link />
```

In HTML5 it is no longer required (even though some people habitually use it):

```
<!-- in HTML5 -->
<br>
<img>
<meta>
<link>
```

Attributes

Each tag can have additional attributes. These are name/value pairs separated by the “=” character, which are used for setting additional tag parameters (e.g. where a link leads, what graphic an img tag should display, etc).



The value assigned to an attribute can be enclosed in quotation marks or apostrophes but it can also be given without them (if it is a single word), or it may not be given at all.
It is recommended to use quotation marks when specifying values for an attribute.

```
<!-- attributes in double quotation marks -->
<input type="text" value="Text">

<!-- attributes in single quotes -->
<input type='text' value='Text'>

<!-- With single-word values the quotation marks
can be omitted, it is not recommended though -->
<input type=text value="Here quotation marks are necessary">

<!-- attributes with no value -->
<input type="text" readonly>
<input type="text" disabled>

<!-- another example of a tag with two attributes -->
<p class="text" data-id="top">

  <!-- element with multiple attributes -->
  <img src="cat.jpg" width="200" height="300" alt="cat" title="Kitty">

</p>
```

Comments in HTML

Just like in other languages, in HTML we can use comments.
Comments can be used to describe or deactivate parts of the code.
Inserting in a document an additional description that **is not displayed by the browser**.
To use a comment in a document, write:

```
<!-- This is a sample comment -->
<nav>

   <ul>
      <li>Home</li>
      <li>Contact</li>
   </ul>
   <div class="text">

   </div><!-- end .text -->
</nav>
```

A comment is just a tag, so it can be placed among other tags, but not inside a tag.
Comment tags can also be used to comment on an existing piece of code so that it is no longer displayed.

```
<nav>
   <ul>
      <li>Home</li>
     <!-- <li>Contact</li> -->
   </ul>
   <div class="text">

   </div>
</nav>
```

In the example above we commented on one element of the list. This can be used when you want a piece of unfinished or pending code not to be displayed to the user, but to keep the source code where it is.

Structure of HTML Document

HTML document structure

Each essay should start with an introduction, followed by a main body, and then a conclusion. Similarly, each website has its own structure. Let's consider this on a sample HTML document:

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Homepage</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

Quick structure creation

In the new HTML editors you can get a similar structure very quickly. Just create a new blank html document, type **!** or **html:5** and press **Tab**. If this trick does not work in your editor, it means that Emmet is not installed in it (https://emmet.io). In editors such as WebStorm or Visual Studio Code it is included by default.

Document declaration

```
<!doctype html>
```

The first tag that should be at the beginning of every **HTML** page is **doctype**. It informs the browser about the type of document to be displayed. In previous HTML versions there were several such declarations and they were complicated (see: https://www.s3schools.com/tag/tag_doctype.asp). Fortunately, version five shortened and unified this.

HTML language declaration

```
<!doctype html>
<html>
<!-- page content -->
</html>
```

The whole content of the page should be placed between the tags **<html>** and **</html>**. This tag can have an additional **lang** attribute which determines the language of the page.

```
<!doctype html>
<html lang="en-US">
<!-- page content -->
```

```
</html>
```

The first two letters (lowercase) are a declaration of the language used, the next two letters (uppercase) are a declaration of the country in which the language is used.

Head section

```
<!doctype html>
<html lang="pl-PL">
   <head>
      <!-- page settings here -->
   </head>
</html>
```

Page header is located in the **<head>** tag. It describes the HTML document settings such as the encoding used, tab icon, page title (displayed on the browser title bar), attached styles etc. Its content is not visible on the page. An example of such settings may look like this:

```
<!doctype html>
<html lang="pl-PL">
   <head>
      <meta charset="UTF-8">
      <title>Sample page</title>
      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
      <meta http-equiv="x-ua-compatible" content="ie=edge">
   </head>
</html>
```

There are not all possibilities, just the most commonly used ones. You can say that they should appear in every starting document. In addition to the above, there may also be a number of settings related to the social networking sites' graphics, attached scripts or styles, other resources, etc. Let's take a look at the most important ones below.

Page encoding

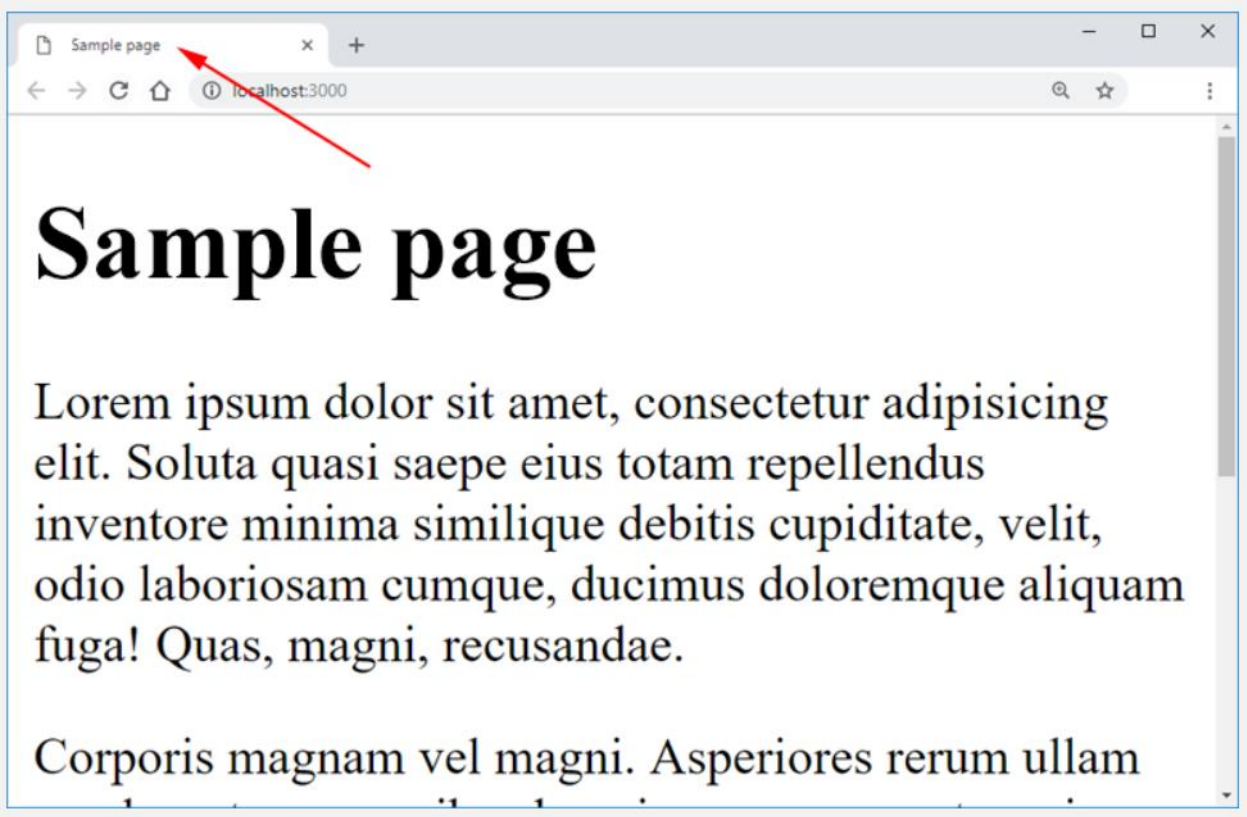
```
<!doctype html>
<html lang="pl-PL">
   <head>
      <meta charset="UTF-8">
   </head>
</html>
```

The **<meta>** tag with the charset attribute is used for setting appropriate page encoding. It is necessary for the correct display of e.g. country-specific characters with diacritical marks. In most cases, the encoding used is the **UTF-8**.
You can learn more about it at: https://developer.mozilla.org/en-US/docs/Glossary/character_encoding

Page title

```
<!doctype html>
<html lang="pl-PL">
   <head>
      ...
      <title>Sample page</title>
   </head>
</html>
```

The **title** tag is used for setting a page title. This is not a title displayed in html, but it appears on the tab (and on the main bar in older browsers).



It is a very important tag in terms of SEO, which affects the position of a website in the search engine. Recommended length of the title is not more than 65 characters.
You can learn more about it at: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/title

Viewport tag

```
<!doctype html>
<html lang="pl-PL">
   <head>
      ...
      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
      ...
   </head>
```

```
</html>
```

Until a dozen years ago nobody thought about displaying pages on screens of different sizes. We had desktop computers and mobile devices did not exist.
With the increase of popularity of mobile devices (smartphones especially) a problem appeared: how to display large pages on small screens.
Apple came up with the idea of a virtual rendering space. The page was rendered in such a space, and then scaled to the screen size. Thanks to that, the pages (which used to not adjust to the screen size) could be displayed entirely on the screen - without the need to scroll.

Although this solution worked, it caused the pages to appear scaled - very small in most cases.

In order for our website to be displayed in the right scale and its elements to adjust to the size of the screen, we need to use the Viewport tag in the settings.

More about this tag: https://developer.mozilla.org/en-US/docs/Web/HTML/Viewport_meta_tag.

x-ua-compatible tag

```
<!doctype html>
<html lang="pl-PL">
   <head>
      ...
      <meta http-equiv="x-ua-compatible" content="ie=edge">
      ...
   </head>
</html>
```

This tag is not necessary for the proper operation of your website (especially when you only support the latest versions of browsers), but it is worth including it in its settings.
With the development of successive versions of Internet Explorer, a mechanism was introduced that was supposed to select a rendering engine appropriate for the detected HTML code. Unfortunately, sometimes this detection did not work properly and the browser switched to reverse compatibility mode (e.g. IE11 would use the engine of the IE9).

The above tag says that IE browsers should always use the latest rendering engine.
More about this tag: https://github.com/h5bp/html5-boilerplate/blob/6.1.0/dist/doc/html.md#x-ua-compatible.

Body section

```
<!doctype html>
<html lang="pl-PL">
   <head>
      ...
   </head>
   <body>
      <!-- actual page content, displayed in the browser -->
   </body>
</html>
```

While the head section is used to declare different settings, the body section contains the actual content of the page that will be displayed in the browser window. It is here that we will save the whole content together with tags.

Other elements

```
<!doctype html>
<html lang="pl-PL">
   <head>
      ...
   </head>
   <body>

      <header class="header">
         <h1>
            This is the <span>first</span> level section heading
         </h1>
      </header>
      <div>
         <p>This is the paragraph content</p>
      </div>

   </body>
</html>
```

Several tags are used in the code above. Each of them has its own role. The **header** tag means a container for headers, the **h1** is a first level tile and the **p** means a paragraph of text. Of course there are many more such tags. You will get to know them in the following chapters. Just as in language grammar, we have to follow certain rules when writing HTML code. We should use appropriate tags in the right places, because each tag has its own purpose. However, two of them do not have special function:

- ** ** - a block element, which aims to enclose larger parts of code, such as sections. Its purpose is not specified as it is in the case of the <p> element, etc. It is mainly used for building parts of the layout, e.g. for later styling.

  ```
  <div>
    <h1>Sample title</h1>
    <p>Sample text</p>
    <div>
      Other content
    </div>
  </div>
  ```

  Additional information: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div.
- is an **inline** element (inline will be discussed later in the course). Its purpose is to enclose smaller parts of the code e.g. text, words, images. It is used mainly for formatting words (e.g. giving color, decorating) or creating small elements such as icon

or ornaments.
Additional information: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span.

```
<div>
  <h1>Sample title</h1>
  <p>Sample <span>text</span></p>
  <div>
    Other <span>content</span>
  </div>
</div>
```

Text formatting in HTML

Text formatting in HTML

We format text using a specific set of tags. They are used for organizing the page and determining what a given text is (e.g. header, paragraph, list). Browsers have predefined ways of displaying the content of such tags (e.g. the header will have a larger font size). With tags basic formatting can be achieved. More advanced formatting requires the use of cascading style sheets (CSS). In this part of the course, we will learn basic tags for formatting text.

Headings

Headings are used to set titles for larger parts of page content. They can be compared to titles and section titles of press articles. The **<h1>** tag is at the top of the heading hierarchy (displayed in the latest font), **<h6>** - at the bottom of the hierarchy. In practice, the most commonly used are **<h1>, <h2>, <h3>**.

```
<!-- here !doctype and head -->
<body>
   <h1>Welcome to my page</h1>
   <h2>About me</h2>
   <h3>My projects</h3>
</body>
```

The sequence of the headings is an important issue. They should be used in the correct order. The first and most significant should be **h1**, followed by subsequent ones. This is shown by the code below:

```
 <!doctype html>
<html lang="pl-PL">
   <head>
      ...
   </head>
   <body>

      <header class="header">
         <h1>
            My page
         </h1>
      </header>

      <section>
         <h2>Latest articles</h2>
         <div>
            <article>
               <h3>Article title</h3>
```

```
            <p>...</p>
          </article>
          <article>
            <h3>Article title</h3>
            <p>...</p>
          </article>
        </div>
      </section>

      <section>
        <h2>Other articles</h2>
        <div>
          <article>
            <h3>Article title</h3>
            <p>...</p>
          </article>
          <article>
            <h3>Article title</h3>
            <p>...</p>
          </article>
        </div>
      </section>
    </body>
</html>
```

Paragraphs

The **<p>** tag (paragraph) is used for grouping text into paragraphs. In the case of HTML, this is not the same as the paragraph used in literature and in many cases this tag will be used for single sentences. This tag will be very useful when you start styling in CSS!

The following code:

```
<h1>Master Thaddeus</h1>
<p>O Lithuania, my homeland! thou art like health; Only he can truly appreciate thy worth Who has lost thee. Now I see and sing thy beauty In all of its glory, because I long for thee.</p>
```

… will yield the following result on the page:

Master Thaddeus

O Lithuania, my homeland! thou art like health; Only he can truly appreciate thy worth Who has lost thee. Now I see and sing thy beauty In all of its glory, because I long for thee.

Special characters

True to their name, special characters have special notation in HTML. They always begin with an ampersand **&**, and end with a semicolon **;** We use them if we want to display characters reserved in HTML or if it is not possible to enter a character from the keyboard.

-   - non-breaking space
- ™ - trade mark sign
- << - less-than sign
- >> - greater-than sign
- © - copyright sign
- ® - registered sign
- α - lowercase Greek alpha

Non-breaking space

If you put a few spaces (one after another) between the words in the code, they will be displayed as one. **To insert a few spaces so that all of them are displayed, we use the so-called non-breaking space character - **

```
Batman   and   Superman<br>
```

Batman and Superman

Batman and Superman
Batman and Superman

The sign **should not be used for making a space between elements** (as shown in the code above) - margins and paddings in CSS are used for this purpose.

New line

You write code, press Enter… and nothing happens. The new line you see in the editor will not be displayed on the page. To move to a new line, you have to use the **
** tag:

```
<p>
  This text is long
  and should be written
  in 3 lines
</p>

<p>
  This text is long<br>
  and should be written<br>
  in 3 lines
</p>
```

Result

This text is long and should be written in 3 lines

This text is long
and should be written
in 3 lines

Subscript and superscript

The **<sup>** tag places text in the upper index.
The **<sub>** tag places text in the lower index.
If you have trouble remembering them, imagine that “p” has a bello on top and “b” on the bottom. Another method is to associate words - sup for superman who flies high, sub for submarine that floats low.

```
<p>
The molecular formula for sulphuric acid is H<sub>2</sub>SO<sub>4</sub>.
Any mathematician knows that 2<sup>0</sup> = 1.
</p>
```

Result

The molecular formula for sulphuric acid is H_2SO_4. Any mathematician knows that $2^0 = 1$.

Formatting tags

We often want to give the text some meaning. There are several tags in HTML that are used for this purpose.

The strong tag is used for representing importance or seriousness for a fragment of the text, for marking a fragment to which we want to draw the reader’s attention. By default, text in this tag will be bold, but nothing stands in the way of changing this default appearance.

```
<p>
... in the mid-20th century, when the first electronic computers were built.
  <strong>They were the size of a large room and used
    several hundred times more energy than modern personal computers
  </strong>, and at the same time they had billions of times less computing power.
```

</p>

Result

... in the mid-20th century, when the first electronic computers were built. **They were the size of a large room and used several hundred times more energy than modern personal computers** , and at the same time they had billions of times less computing power.

Link to the documentation: http://w3c.github.io/html/textlevel-semantics.html#the-strong-element

Seems similar to the previous one? But the **** tag has a whole different meaning. **Strong** reflects importance, seriousness, or urgency of the fragment enclosed in it, while **** marks a gramet but does not increase its importance. This is important for text-to-speech readers. The **** tag will be read with a different accent, while **** is just a visual mark.

This text has no logical meaning, but is worth marking

Result

This text has no **logical meaning, but is worth marking**

Link to the documentation: http://w3c.github.io/html/textlevel-semantics.html#the-b-element

<cite>

The **cite** tag is used for indicating a quote in the text. It is most commonly used to include the author of the cited text.

First sentence in
<cite>
  Nineteen Eighty-Four
  by George Orwell (Part 1, Chapter 1)

</cite>.

Result

First sentence in *Nineteen Eighty-Four* by George Orwell (Part 1, Chapter 1) .

Link to the documentation: http://w3c.github.io/html/textlevel-semantics.html#the-cite-element

<code>

A tag used to mark a piece of text that is code. By default, the text in this tag is displayed in a monospace font (as the ones used in editors).

The <code>push()</code> method is used for adding new elements to an array.

Result

The `push()` method is used for adding new elements to an array.

Link to the documentation: http://w3c.github.io/html/textlevel-semantics.html#the-code-element

<del>

The **<del>** tag means deleted text. Most commonly used if you make corrections in the text by striking through the previously written text.

The dog was ~~brown~~ black.

Result

The dog was ~~brown~~ black.

Link to the documentation: http://w3c.github.io/html/edits.html#the-del-element

It gives the text stress emphasis, emphasizes the meaning of a word. It is usually marked in italics.

I am not convinced about this.

Result

I am *not* convinced about this.

Link to the documentation: http://w3c.github.io/html/textlevel-semantics.html#the-em-element

In the past it only meant text written in italics, but its meaning has changed in HTML5. This tag means text that is slightly different from the surrounding text, has a slightly different sound. It can be used for example for technical words or names.

The word bandwidth means the amount of data that can be transmitted across a given path in a unit of time.

The word *bandwidth* means the amount of data that can be transmitted across a given path in a unit of time.

Link to the documentation: http://w3c.github.io/html/textlevel-semantics.html#the-i-element

<kbd>

Text that denotes keyboard input.

The best shortcut in Chrome debugger is <kbd>Ctrl + Shift + P</kbd>

Result

The best shortcut in Chrome debugger is `Ctrl + Shift + P`

Link to the documentation: http://w3c.github.io/html/textlevel-semantics.html#the-kbd-element

<small>

Tag used for footnotes, license texts, copyright.

```
<footer>
  <small>&copy; Copyright OurFirm</small>
</footer>
```

Result

© Copyright OurFirm

Link to the documentation: http://w3c.github.io/html/textlevel-semantics.html#the-small-element

<u>

A tag that indicates an annotation that is not context related (e.g. a proper name in Chinese text, or incorrect words in the text when spell check is enabled). By default, the text enclosed in this tag is underlined.

```
<p>Japanese, Hungarian and Chinese names are written with first letters of the family name: <u>Wu</u> Xiaoqian.</p>
<p><u>Warsaw</u> is the capital of Poland.</p>
```

Result

Japanese, Hungarian and Chinese names are written with first letters of the family name: Wu Xiaoqian.

Warsaw is the capital of Poland.

Link to the documentation: http://w3c.github.io/html/textlevel-semantics.html#the-u-element

A bit more on the and <i> tags

By default, the tags **** and **<i>** look similar on the page. Remember that tags are not the appearance! Both of these tags can be styled so that they will look like bold text, a square box on the page, or even full screen animated graphics. The most important thing is…the meaning of the tag.

About **<b>**, **<u>** and **<i>**

In the past, tags such as **<b>**, **<u>** and **<i>** had no semantic meaning. They only gave a different look to the text they contained. With the creation of HML5, they gained their meanings, which are described above. Nowadays, they are used every day.
In most cases, **** and **** tags are used. This is because it is not always clear what the meaning of a word or text is (and sometimes it is debatable). The second reason is more ordinary: developers are used to using the **** tag.

Hyperlinks

Hyperlinks

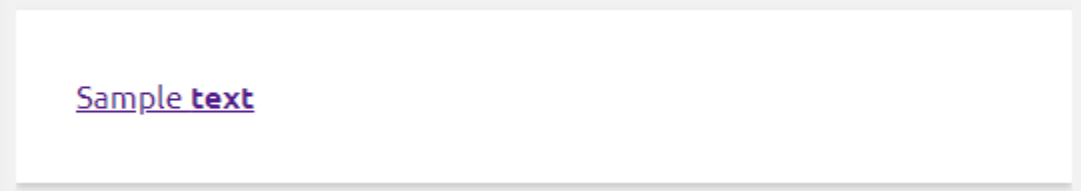
Hyperlinks (links) are elements that connect individual pages on the Internet and allow you to move from page to page.

Take me to the page

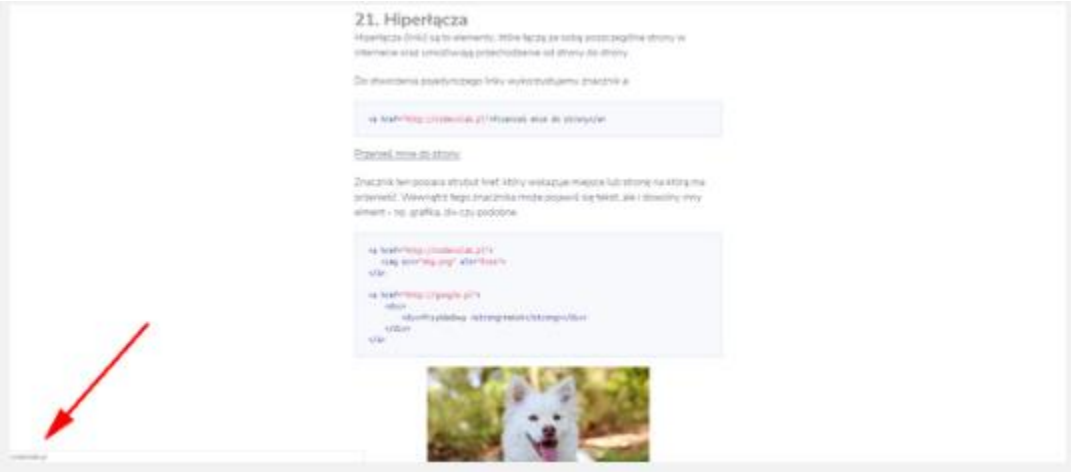
This tag has a **href** attribute which indicates the place or page to which it should redirect. Inside this tag tere can not only be text, but also any other element such as an image, a div, etc.

Take me to the page

```
<a href="https://www.google.com/">
   <div>
      <div>Sample <strong>text</strong></div>
   </div>
</a>
```



A distinctive feature of the links is that if you hover over them with your cursor, the address to which the link leads will appear in the bottom corner of your browser and the cursos will change its appearance to a “hand”.



Types of links

Links are divided into internal and external ones. External links start with a protocol which is usually **http** or **https** and redirect to external pages (the user moves to another website).

```
<a href="https://google.com">Go to Google website</a>
```

Internal links move the users within the range of a given website. We divide them into:

- links leading to another subpage within the same page,
- links leading to a section on the same page.

Hyperlinks to subpages

If a given hyperlink does not start with a protocol, when clicked it will take us to the subpage on our website that is indicated by the href attribute:

```
<a href="contact.html">I will redirect to the page contact.html</a>
<a href="info.html">I will redirect to the page info.html</a>
<a href="song.mp3">I will redirect to the page song.mp3</a>
```

If a link does not start with http, then even if it points to a potentially valid address, the browser will want to redirect to a subpage with this name:

```
<!-- The link will redirect to the subpage www.google.com on our website - probably it doesn't exist -->
<a href="www.google.com">Click</a>
```

Hyperlink to a given place on the page

Links that move to a given section on the same subpage (also called anchors) start with #, followed by the id of the element to which the link leads:

```
<a href="#third-title">Go to the third title</a>

<h1>First title</h1>
<p>Example of a very long text</p>

<h2 id="second-title">Subheading</h2>
<p>Example of a very long text</p>

<h2 id="third-title">Subheading</h2>
<p>Example of a very long text</p>
```

After clicking on such a link, the page will jump to the indicated place and the value of the href attribute of the link will be added to the address in the browser. This allows you to easily send your friend a page address so that they can start reading from the indicated place.

```
<a href="#bottom">Move to the bottom of the page</a>

<div class="#bottom"></div>
```

A similar principle also applies to links to external pages. If such an address ends with a # and a corresponding ID, when loaded, the page will jump to the indicated place.

```
<a href="https://www.smashingmagazine.com/2018/10/photoshop-workflows-shortcuts-digital-artists/#6-working-with-layers">
  Go the the indicated part of the article
</a>
```

Fun fact

If you want the link to direct to a specific place in the text, just look for the closest item with the ID attribute in the page code. If you add this **#foundID** to the page address, after receiving the link your friend will start reading from the given place.
When creating pages that are read frequently, it is a good idea to add IDs to subsequent subheadings in the text so that potential users can share relevant links more easily.

href attribute

As you have noticed above, the **href** attribute is used to indicate where a link should redirect to:

```
<!-- will redirect to the page youtube.com -->
<a href="http://youtube.com">click</a>

<!-- will redirect to the subpage contact.html -->
<a href="contact.html">click</a>

<!-- will redirect to the element with an id that equals "end" -->
<a href="#end">click</a>

<!-- will redirect to the beginning of the page -->
<a href="#">click</a>

<!-- will redirect to the indicated place in the article -->
<a href="https://en.wikipedia.org/wiki/HTML#Attributes">click</a>
```

rel attribute

The **rel** attribute is used for positioning purposes. It determines how Googlebot indexing a given page should behave. This attribute can take the following values:

- **dofollow** - default
- **nofollow**.

```
<a href="http://www.coderslab.com" rel="nofollow">
   IT school
```


If the indexing bot encounters a link with the rel attribute set to **nofollow**, it will not follow the link and the page value in terms of indexing will not change. If a given link has the rel attribute set to **dofollow**, the bot will follow the link and the value will be transferred to the page to which the link leads.

dofollow links	nofollow links
Improve positioning	Less important for positioning
Pass the value of the page	Do not pass the value of the page
Pass the value of the link text	Positive impact on the link structure

What is Googlebot?

Both attributes are used by Googlebot. It is a web crawler used by Google. It moves from one page to another and thus creates an index for the Google search engine. Thanks to this we can later search for things in Google's search engine.

target attribute

The target attribute determines whether a link should be opened in a given tab or in a new one.

- _**blank** - opens a page in a new tab,
- _**self**- opens a page in the same tab.

```
<!-- the link will be opened in a new tab -->
<a href="http://some-another-page.com" target="_blank">click</a>

<!-- the link will be opened in the same tab -->
<a href="http://some-another-page.com" target="_self">click</a>
```

We have discussed the rel attribute above. Apart from the presented values, you can also set it to **noopener**.

```
<a href="http://some-another-page.com" target="_blank" rel="noopener">click</a>
```

This is particularly important for links with the target attribute set to _**blank**. By default, when the page to which a given link redirects is opened in a new tab, it has the ability to redirect to the page from which it was opened using the opener property. This exposes the page to potential attacks.

In short, such an attack may look more or less like this. Imagine that on your large portal you place a link to an external page that opens in a new tab (**target="_blank"**). Author of this page uses the opened property to replace the page in your tab with his own page (e.g. one that

suggests that the user was logged out). The user returns to the old tab and, seeing tht he has been logged out, tries to log in again with his data…

Inline frame

The **iframe** element (inline frame, floating frame) allows you to include an HTML document inside another HTML document. The **src** attribute of the iframe element you write the URL of the page to be displayed in this floating frame.

```
<iframe src="http://www.youtube.com/embed/tAGnKpE4NCI" width="100%"
height="400"></iframe>
```

Images in HTML Documents

Images in HTML documents

For inserting an image into a page, use the **img** element.

<img src="apple.jpg" width="600" height="400" alt="apple" title="Apple">

The most commonly used attributes for this element are:

src attribute

The **src** attribute contains the path to a given graphic file.

<!-- relative path to image -->
<img src="images/lovely-flowers.png" alt="flowers">

<!-- graphics downloaded from an external server -->
<img src="http://other-server.com/pears.jpg" alt="pears">

This attribute can also contain a string, which is an image encoded in base64. This second method is much less frequently used.

Attributes width and height

Optional attributes **width** and **height** set the initial size of an image. This ensures that a place in the layout remains reversed for the image while it is loading. This way the rest of the layout “does not jump” when loading.

```
<!-- 200px wide, 100px high -->
<img src="dog.png" width="200" height="100">

<!-- 100% of parent's width, 200px high -->
<img src="dog.png" width="100%" height="200">

<!-- 100% of parent's width, 50% of parent's height -->
<img src="dog.png" width="100%" height="50%">
```

Nowadays, these attributes are used only occasionally because CSS styling has taken over their role.

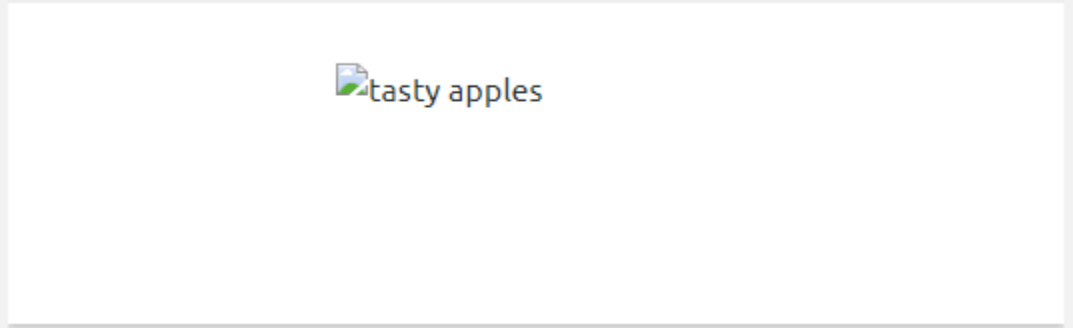
alt attribute

One of the most important attributes of every image on the page. The **alt** attribute contains alternative text, which is displayed when the image is not loaded. **This is especially important for visually impaired people who use special text-to-speech readers for Internet browsing**.

Alt attributes are read by the speakers, so that visually impaired users can find out what an image shows. Do not produce complicated descriptions. Simplicity is the best approach here:

```
<!-- wrong -->
<img src="..." alt="This image shows apples">

<!-- much better -->
<img src="..." alt="tasty apples">
```



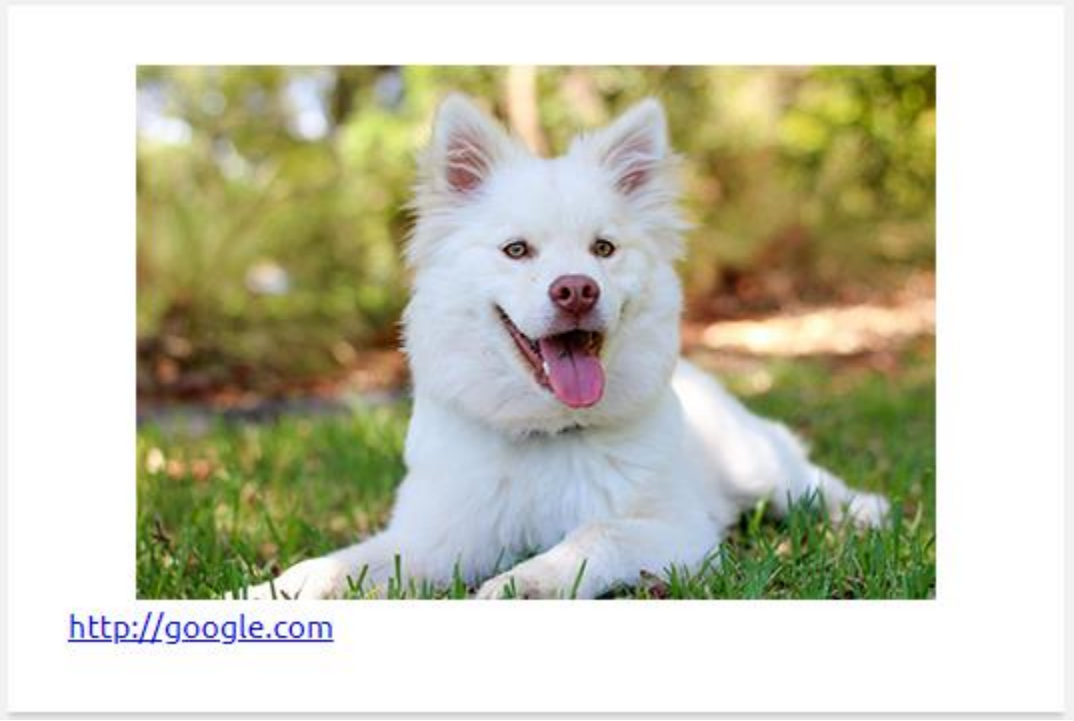
This attribute is also important from the positioning perspective, because Google robot does not see an image depicts, but the text in alt will be indexed.

title attribute

The **title** attribute contains text that will appear in a tooltip when you hover over the element. It can be used not only for images, but also for other elements on the page.

<img src="dog.png" alt="white dog" title="White dog">

http://google.com



Just like alt, this attribute is useful from the page positioning viewpoint.

IMG and background in CSS

In the following chapters you will also learn about placing graphics with CSS. What should you choose and when? Remember that the graphics placed on the page with the **img** tag are **page content**, while graphics placed with CSS are only decoration.
A nice corner for a div will rather be CSS, but a photo of a person adding a comment, a product in a shop or a car at auction is **information**.
Sometimes it will not be so obvious, but these are the rules you should follow when adding graphics to a page.

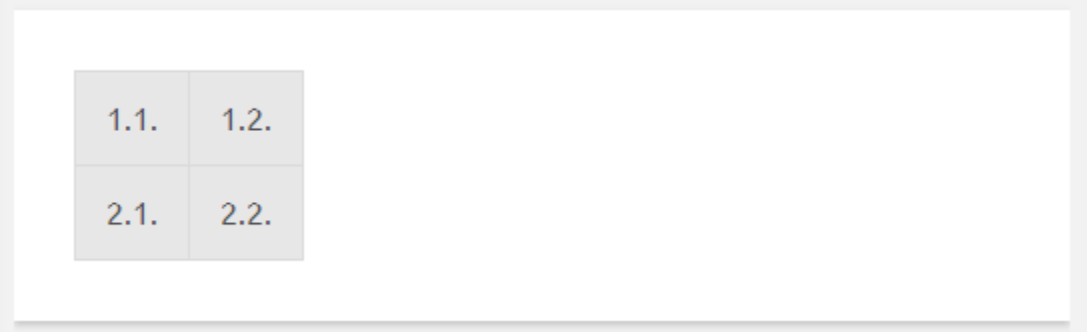
Tables in HTML

Tables

```
<table>
  <tbody>
    <tr>
       <td>1.1.</td>
       <td>1.2.</td>
    </tr>
    <tr>
       <td>2.1.</td>
       <td>2.2.</td>
    </tr>
  </tbody>
</table>
```

Above we see the code of a simple table. Each table consists of several basic tags:

- **table** - a tag in which the whole table is nested
- **tr** - table row - means successive rows of the table
- **td** - table data - means single table cells



After inserting the code of a table into a document you will see the data of this table. The whole appearance (border, cell color, etc.) will be given by CSS styling. In the following text, for better visibility, a border for tables and cells was added by means of styling.
It is important that, just like in the case of lists, in tales we also have a certain structure in place. Only selected tags (**tr, tbody, thead, tfoot**) can be placed directly in the table tag, so it is a mistake to place e.g. a **div** tag directly there.

Headers

Apart from the **td** cells, each table can have headers cells that are written using the **th** tag. By default, their contents are centered and highlighted in bold text, but with the help of styling we can give them any other appearance.

```
<table>
  <thead>
    <tr>
       <th>User name</th>
       <th>User points</th>
    </tr>
  <thead>
  <tbody>
    <tr>
       <td>Robert Newman</td>
       <td>210</td>
    </tr>
    <tr>
       <td>Michael Diaz</td>
       <td>110</td>
    </tr>
    <tr>
       <td>Peter Robins</td>
       <td>230</td>
    </tr>
  </tbody>
</table>
```

User name	User points
Robert Newman	210
Michael Diaz	110
Peter Robins	230

Table headers do not have to be placed only at the top:

```
<table>
  <tbody>
    <tr>
       <th>Car name</th>
       <td>BMW</td>
       <td>Audi</td>
       <td>Fiat</td>
    </tr>
    <tr>
       <th>Color</th>
       <td>Black</td>
       <td>Red</td>
       <td>Blue</td>
    </tr>
    <tr>
       <th>Kilometers driven</th>
       <td>320.000</td>
       <td>120.000</td>
```

Car name	BMW	Audi	Fiat
Color	Black	Red	Blue
Kilometers driven	320.000	120.000	240.000

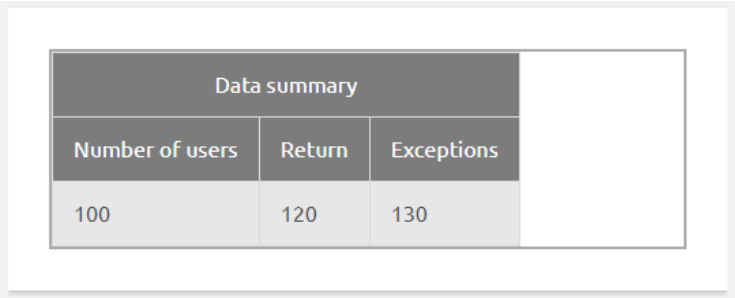
```
      <td>240.000</td>
    </tr>
  </tbody>
</table>
```

Merging cells

Each **td** and **th** cell has two attributes that allow it to merge with other cells:

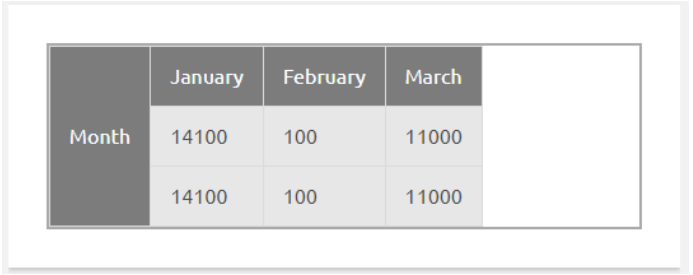
- **colspan** - determines how many columns a given cell should occupy
- **rowspan** - determines how many lines a given cell should occupy

```
<table class="tab">
  <tbody>
    <tr>
      <th colspan="3">Data summary</th>
    </tr>
    <tr>
      <th>Number of users</th>
      <th>Return</th>
      <th>Exceptions</th>
    </tr>
    <tr>
      <td>100</td>
      <td>120</td>
      <td>130</td>
    </tr>
  </tbody>
</table>
```



Data summary		
Number of users	Return	Exceptions
100	120	130

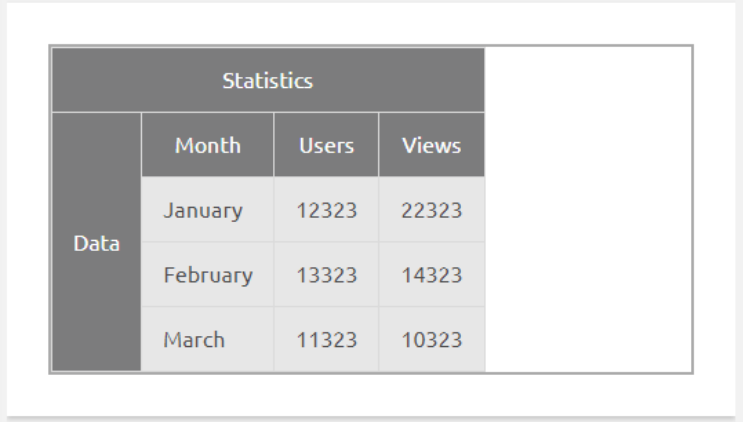
```
<table class="tab">
  <tbody>
    <tr>
      <th rowspan="3">Month</th>
      <td>January</td>
      <td>February</td>
      <td>March</td>
    </tr>
    <tr>
      <td>14100</td>
      <td>100</td>
      <td>11000</td>
    </tr>
    <tr>
      <td>14100</td>
      <td>100</td>
      <td>11000</td>
    </tr>
  </tbody>
</table>
```



Month	January	February	March
	14100	100	11000
	14100	100	11000

Combination of both parameters:

```
<table class="tab">
  <thead>
    <tr>
      <th colspan="4">Statistics</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th></th>
      <th>Number of users</th>
      <th>Return</th>
      <th>Exceptions</th>
    </tr>
    <tr>
      <th rowspan="3">Month</th>
      <td>January</td>
      <td>February</td>
      <td>March</td>
    </tr>
    <tr>
      <td>14100</td>
      <td>100</td>
      <td>11000</td>
    </tr>
    <tr>
      <td>14100</td>
      <td>100</td>
      <td>11000</td>
    </tr>
  </tbody>
</table>
```



Statistics			
	Month	Users	Views
Data	January	12323	22323
	February	13323	14323
	March	11323	10323

About table generators

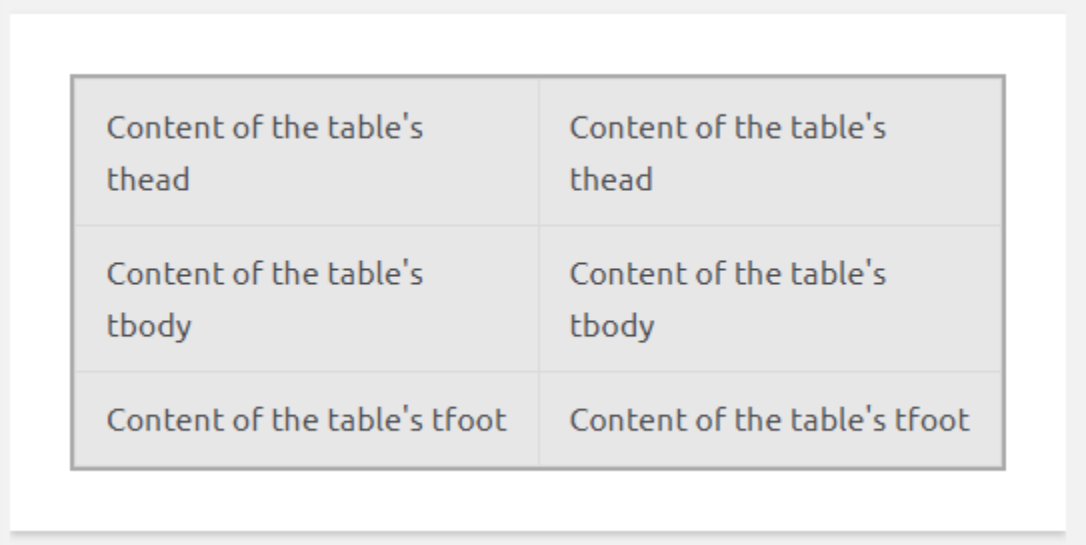
As you can see, even with such simple tables the code does not always seem obvious. For more complex constructions, it is with using table generators such as https://www.tablesgenerator.com/html_tables.

Table sections

Each table can be additionally divided into three sections:

- **thead** - sections with table header
- **tbody** - main body of the table
- **tfoot** - table footer

```
<table>
   <thead>
      <tr><!-- Content of the table header --></tr>
   </thead>
   <tbody>
      <tr><!-- Content of the table body --></tr>
   </tbody>
   <tfoot>
      <tr><!-- Content of the table footer --></tr>
   </tfoot>
</table>
```



Such division is not required, but is useful for more complicated table styling. The interesting thing is that even if you do not add **<tbody>** in your table, the browser will do it for you while rendering the page.

A bit about the history

In the past, tables were used to build entire page layouts. A table with invisible borders was created and the whole layout was sliced into pieces by placing appropriate parts into subsequent cells. Hence the term “slicing layout” which you can still encounter sometimes. Nowadays, this method is wrong because we should create the layout using other techniques such as FlexBox (you will learn about them later in the course). Remember that the purpose of tables is **to display tabular data**.
Interestingly, the method of building layouts based on tables is still used to this day for encoding mailings. This is mainly due to the primitiveness of some readers that do not support most CSS properties.