

LEARNING

THE HTML

Markup Language



Contents

1. Introduction to HTML	03
2. Tags	06
3. What's in the HTML website?	09
4. How to add a text	13
5. Indentation and more tags	17
6. Lists	19
7. Hyperlinks	21
8. Images	24
9. Basic page structure	27
10. DIV and SPAN	31

1. Introduction to HTML

HTML is an abbreviation for Hypertext Markup Language. It may sound daunting but don't worry. It's simply the language used for building websites. We use HTML to define the content and structure of the website.

We use HTML to describe every element on a website. The browser will know how to interpret these elements and how to display them. Take a look at the following examples. Think for a moment about code and how it's interpreted by the browser:

First example:

```
<h1>Header</h1>
```

Header

Second example:

```
<p>The text in the paragraph</p>
```

Paragraph

Last example:

```
<button>Submit</button>
```

We write and publish the HTML code in a document with an ".html" extension. The browser knows that such a document contains HTML code and knows how to deal with its content.

For example: We want to create an HTML document named "index". So we create a document named "index.html" - with an ".html" extension.

HTML will allow you to add to your website any element imaginable. You can build:

- headings,
- paragraphs,
- forms,
- data tables,
- video and audio player,
- hyperlinks,
- graphics,
- and much more...

2. Tags

We already used some tags in the previous lessons. Now we will take a closer look at them. Here is an example of a simple tag:

```
<h1>Header</h1>
```

What's important:

- <h1> - this is an opening tag. The browser will know that it should interpret the following code as a level 1 heading.
- Welcome - this is the text that will be displayed as a level 1 heading
- </h1> - this is the closing tag. This is the end of our heading

Everything we put between <h1> and </h1> tags will be displayed as a level 1 heading.

There are some of different tags:

- <h1> - level 1 heading
- <p> - paragraph
- <a> - hyperlink
- <table> - table
- <section> - section
- <footer> - footer
- - important/bold text
- <nav> - navigation

These tags above are selected examples of vast possibilities offered by HTML.

Note that we write tag names in lowercase - it's for a convenience and readability reasons.

Majority of HTML elements include both opening and closing tags. That's because they usually include some content inside e.g. text. There are also special elements that use only an opening tag. We sometimes call them "self-closing elements".

Elements using both opening and closing tags:

- <p></p> - paragraph
- <h1></h1> - heading h1
- This is the content.

Elements using only opening tags (self-closing elements):

- <hr> - horizontal rule
- - image
-
- <input />
- <link />

3. What's in the HTML website?

In order to render our website correctly, the browser needs to know that it's dealing with a proper HTML document. It will allow the browser to correctly parse and display the contents of the website. For this task we need to start our HTML document with a special declaration.

Every website consists of two main parts:

- 1. important information for the browser**
- 2. content of the page accessible by humans**

They are wrapped with the: `<head>` and `<body>` tags. Both parts are always wrapped with an `<html>` tag. Take a look at the code:

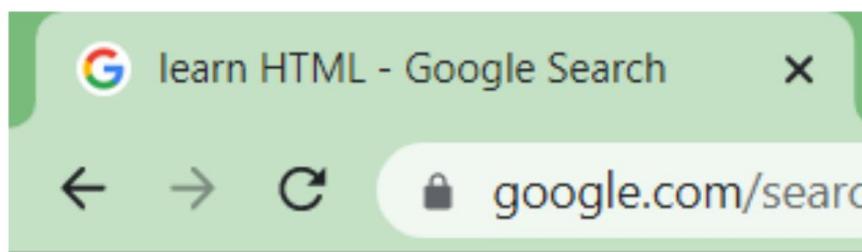
```
<!DOCTYPE html>
<html>
<head>
    HEAD section
</head>
<body>
    BODY section
</body>
</html>
```

Every webpage needs a title. We place the title in the `<head>` section of the page. It should be surrounded by the `<title>` tags also.

They are wrapped with the: <head> and <body> tags. Both parts are always wrapped with an <html> tag. Take a look at the code:

```
<head>
    <title>HTML Course</title>
</head>
```

The title will be visible at the top of the browser window. It is also visible in the Google search results:



The title is a very important element for both the users of our website and the search engines bots.

In the previous exercise you may have noticed this declaration:

```
<meta charset="UTF-8">
```

It's a `<meta>` tag with a `charset` attribute. It's a very important element. It ensures that you can use characters from just about any language in your HTML document, and they will display reliably.

Note that this element is utilized by the browser and that's why we place it in the `<head>` section.

4. How to add a text

Paragraphs are blocks of text separated visually from adjacent blocks by a blank line. To create a paragraph we use this simple code:

```
<p>This is a paragraph</p>
```

It's simply a block of text surrounded with `<p>` tags. Usually it will include a few sentences of text.

Here is the output of the above code:

This is a paragraph

In most cases you will also need some headings. You can use them to declare a title of the article or a section.

Here is an example of level 1 heading (h1):

```
<h1>This is a level 1 heading</h1>
```

Headings usually are much shorter than paragraphs - in most cases they will contain a few words.

There six levels of headings available in the HTML:

- <h1>Heading level 1</h1>
- <h2>Heading level 2</h2>
- <h3>Heading level 3</h3>
- <h4>Heading level 4</h4>
- <h5>Heading level 5</h5>
- <h6>Heading level 6</h6>

<h1> is the most important heading.
<h6> is the least important heading.

Heading level 1

Heading level 2

Heading level 3

Heading level 4

Heading level 5

Heading level 6

Some parts of the text need to be emphasised:

```
This text <strong> is important</strong >
```

```
This text <em> has been emphasised </em>
```

You can emphasise the text by wrapping it with `` or `` tags.

This text is important.

This text has been emphasised.

Sometimes a piece of text needs to be marked as a quotation or citation.
Take a look at the code:

```
<blockquote>  
    "This is a great introduction to HTML"  
</blockquote>
```

That's how it looks in the browser:

Here is the quote:

"This is a great introduction to HTML"

In most cases the text of the quotation will be indented but in practice it's browser dependent.

5. Indentation and more tags

You may have already noticed that we indent tags that are placed inside another tag. Take a look at the example:

```
<body>
  <p>This is a paragraph</p>
  <blockquote> "It's a citation!"
  </blockquote><
    p>This is another paragraph</p>
</body>
```

What's the purpose of this indentation?

We use the indentations to make our code more readable. The browser will ignore them anyway.

But developers are another matter. They need a clean code that is easy to read. That's why you should always properly indent every line of code.

In practice we use Spaces or Tabs to add indentation.

We use the below tag to visually separate two distinct sections. Take a look:

<hr>

Here is the result in the browser:

This is a paragraph

This is another paragraph

Note that the <hr> tag is self-closing!

6. Lists

There are two basic kinds of lists.

An unordered list:

- bread
- butter
- milk

And an ordered list:

- 1.go straight ahead
- 2.turn right
- 3.then turn left

The code for both kinds of lists look like this:

an unordered list

```
<ul>
  <li>bread</li>
  <li>butter</li>
  <li>milk</li>
</ul>
```

an ordered list

```
<ol>
  <li>go straight ahead</li>
  <li>turn right</li>
  <li>then turn left</li>
</ol>
```

Note that some elements are common for both kinds of lists - namely the `` tag.

7. Hyperlinks

Let's a hyperlink to Google webpage.
Here is the code:

```
<a href="http://google.pl">Google</a>
```

The hyperlink usually looks like this:

Google

If we click on this link we will be redirected to the Google homepage.

Let's take a closer look at the hyperlink code:

```
<a href="http://google.com">Google</a>
```

- Hyperlink text is always wrapped with tags.
- We always add a "href" attribute to the opening tag ().
- We use the href attribute to declare a URL - the address of the webpage/document we will be redirected to after clicking the link.

There are two basic kinds of hyperlinks:

```
<a href="http://google.pl">Google</a>
```

Absolute hyperlink -

links to the external webpage e.g. We need to provide the full domain name e.g. <http://google.com>

```
<a href="contact.html">Contact Page</a>
```

Relative hyperlink -

links to the internal page or document. We need to provide the path to the document e.g. contact.html

8. Images

Take a look at the code:

```

```

- img tag is self-closing. So we use only an opening tag.
- src attribute is using to declare a path to the image file.

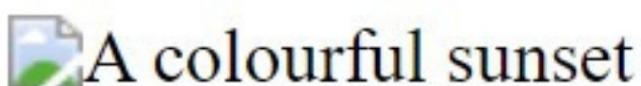
So using this simple code we can easily add the image to our webpage.

There is an additional, very important attribute that you should always add to every image:

```
<img src=sunset.jpg" alt="A colourful sunset">
```

- The alt attribute specifies an alternate text for an area, if the image cannot be displayed.
- The alt attribute is required if the href attribute is present.

It can look like this:



Let's get back to the alt attribute. There are several important reasons to use this attribute for every image we add to the website:

- If for some reason the image won't load we will still get to see the text of the alt attribute.

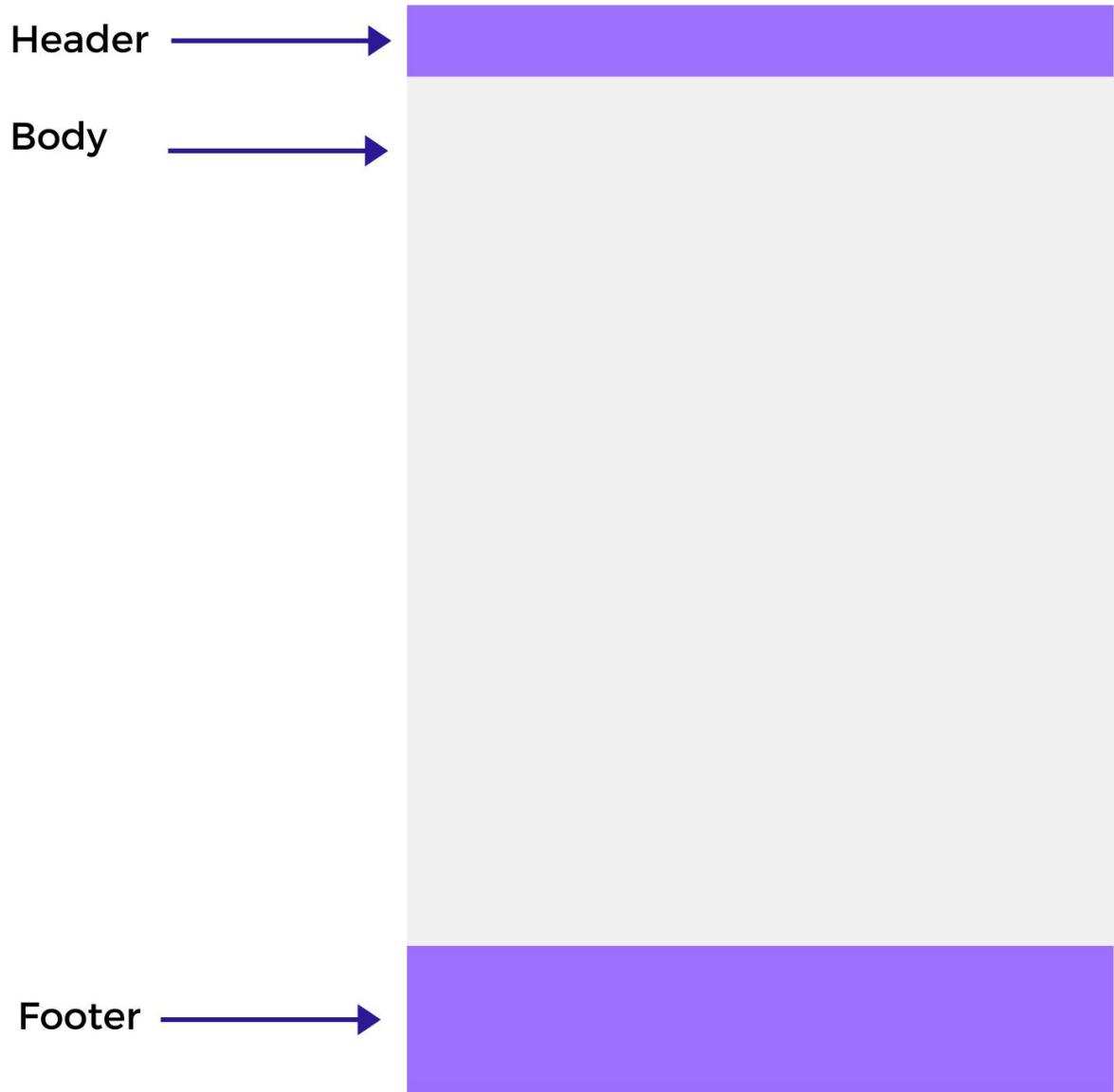
- There are numerous people with disabilities, poor vision etc. They browse the Web with a special screen reader software that read aloud the text content of the page. Images without alt text are useless for them.
- Search engines, like Google, need to know what's the content of the image. So they simply read the value of the alt text.

The alt attribute is very important! You should always add it to every image on your website.

9. Basic page structure

Every website needs a basic structural skeleton. Think of some websites you visit frequently. You could without a doubt describe some elements that are common for every site e.g. header, menu or footer. Now you will learn how to build such elements using a few HTML tags.

We could build an unlimited number of websites based on this simple schema:



- **Header** - located at the very top. It usually includes elements like: logo, menu etc.
- **Main content** - well...main content of the website: articles, videos, images etc.

- Footer - located at the very bottom.
Usually includes some basic info about the website, contact information etc.

You can easily recreate this structure using these tags:

- `<header></header>` - Header (don't confuse with headings h1-h6!)
- `<main></main>` - Main content
- `<footer></footer>` - Footer

Properly built website should be based on a semantic markup. Using semantic markup we make sure that all elements on our website are properly described. In other words: every element should have a meaning. It's very important for several reasons:

- Google search engine needs to know what elements on a webpage are the most important ones.
- People with disabilities are often using a screen reader. Such a software in order to work needs to know what's the purpose of a given HTML element.

The elements you've just seen are good examples of semantic tags:

- `<header>` - we use this tag to define The Header
- `<main>` - we use this tag to define The Main Content
- `<footer>` - we use this tag to define The Footer

So these are not ordinary tags. They give meaning to the elements of our website.

10. DIV and SPAN

Sometimes there won't be a suitable semantic tag for a given element.

That's why the HTML comes with two very useful tags: DIV and SPAN. They are pretty common so you should learn how to use them.

We have a code for a simple shopping cart. We need to group all elements together. Since HTML doesn't have a semantic tag for a cart we need to use a generic tag:

```
<div>
  <h2>Cart</h2>
  <ul>
    <li>Product no.1</li>
    <li>Product no.2</li>
  </ul>
</div>
```

We grouped all elements with the `<div>` tag. It's a generic container for elements that for some reason belong together.

There are two main reasons for using the `<div>` tag:

- we need to style the elements as one group
- there is no suitable tag for a given element

Remember that the `<div>` tag itself won't change the appearance of elements. It's simply a way to group all elements together.

Sometimes we need to group some parts of the text for styling purposes. We can use a `` or `` tags but in some cases their usefulness is limited.

That's why we got a `` tag. It's a generic container for text:

```
<span>Tekst</span>
```

We wrap a piece of text with a pair of `` tags. It will allow us to style this text with CSS stylesheets.

We added a `` tag in the previous examples to target a piece of text with a CSS rule. That way we can easily style any part of the text. Here is a simple example:

My mother has **blue** eyes and my father has **dark green** eyes.

In order to make it work we need to add a class attribute:

```
<span class="blue">Some text</span>
```

Then we can easily style this “blue” class with a suitable color using CSS stylesheet. We won’t go into writing the CSS for now - it is a complex topic outside of the scope of this course.