

CLIZ WARE DEVELOPING NETWORK

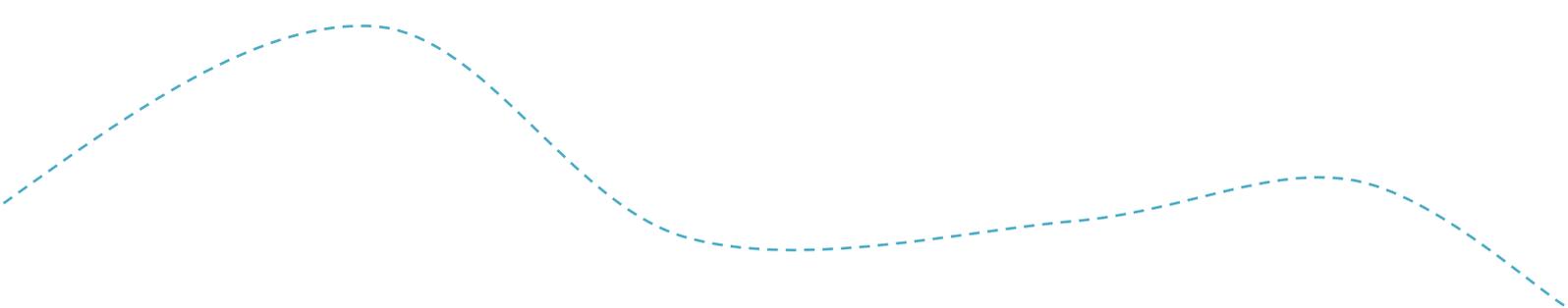
Cliz Ware

Developing Web

Basics of HTML

Artem Los

2011-11-26



Usage of “Dev Net”

To be able to use all feature in this course, you may use CliZ Ware HTML Editor. This is a free editor that might be found at www.editor.clizware.net/. The editor itself is actually designed for this course with all samples that are given in this book.

1. Let us start with figuring out how the editor works. First, go to the page given above. You may now add your source code in the left box. After that, you will see how it will look in a browser.
2. You can also view an example that is given in this book at internet. You will need to enter the number that is given under an example. To do that you have to add a suffix to the web-address. For example, try to open example **1** as follows:
www.editor.clizware.net/?id=1. The only thing we have added is **?id=1** to the address. So, the pattern is therefore **?id=number**.

Good Luck!

Artem Los, IB11C, Katedralskolan, Uppsala, Sweden

P.S. To view all examples, type www.editor.clizware.net/main/ in your web browser.

Chapter 1 – What is what?

So, today we will look at the difference between HTML, CSS, and Java Script. The basic difference between these languages is that HTML is used to design, CSS is used to create a template, and Java Script is used to make something happen. In this chapter, we will look at the structure of HTML, and some few tags.

HTML – Hypertext Markup Language, however, is based on tags, and looks kind of the same as XML. In a html document you have a specific structure, shown below:

```
<html>
  <head>
  </head>

  <body>
  </body>
</html>
```

You have a “head” and a “body” in this document. “Body” is that part that usually is the main page. In the “head” you might want to put information about the website, “meta”, maybe a “title”, or include some script. A “html” tag is actually telling the browser that the page contains HTML.

You may now want to make something happen, but first you have to save the document. If you are using notepad, just click on “Save As” in File menu, select Save As type to “All files”. Enter then “myfirstpage.html” as name of the document. Now your file is saved. Remember that the extension, format, of a html file is “html”. Let us just add some few things before we preview our page.

The most common phrase to use at the beginning of a course is – “Hello World!”. Now, save your document again, and double click at your page.

Hello World!

Hello World is now displayed on you page, congratulations!

As a summary, let us now go through those tags we have used:

“html” tag – should be included in all websites.

“head” tag – used for storing information about the website. Information is used by search-robots, such as Google.

“body” tag – is the main tag. Almost everything putted in-between this tag will appear in you page.

```
<html>

  <head>
  </head>

  <body>
  Hello World!
  </body>

</html>
```

Sample 2.

Questions:

- 1) What is HTML standing for?
- 2) Which format does a HTML document have?
- 3) In-between which tag should the information about the page be putted?
- 4) In-between which tag should the text that be putted to make it appear in a web browser?
- 5) What is the most common used phrase that always is used at the beginning of a course?

Tags – how they work, (part 1)

In this chapter we will look at how a tag is constructed. We will go from basic to more advanced tags. Let us look at a simple tag.

```
<p> Paragraph</p>
```

Here we have a simple paragraph, and we will use it to describe a tag deeper. As you might see, all tags start with a *start, opening, tag*, “<p>”, and ends with an *end, closing tag*, “</p>”. To summarize, “<name>...</name>”.

Now we will look at how this tag might be extended.

```
<p style="color:blue;"> </p>
```

Sample 3.

The tag above has now been extended with CSS, by the *style attribute*. An attribute is that you place inside the opening tag, to provide **additional information** about the tag. An attribute is constructed as follows, **name="value"**. For example, the code we have written inside the style attribute is telling the browser that the text in-between paragraph tag should be blue.

The most common attributes are, for instance, **class, id, style, title**.

Another example would be a link tag, **<a>**. To make sure that the web browser understands where to go when user enters the link, we need to provide additional information about the web address. In link tags, the name of attribute is **href**. Take a look on the code below:

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

When user presses the link, he will be redirected to google’s website.

Notice: you have to enter the proper address to the website, i.e. always include *http://*. If you skip this part, the web-browser will try to find the page somewhere on your server.

Now, we will look at tags, such as image-tag, **img**, where you do not need information in-between the tag.

```

```

As you might see, we do not have some closing tag there. Although it is a normal tag, we do not need to provide extra information. The closing part is therefore “/”>. The **src** attribute is the path to the image, and the **alt** attribute is the text that will be displayed when user holds his mouse over.

- 1) What are start tags and another name of them?
- 2) What are ending tags and another name of them?
- 3) What is an attribute and what is it used for?
- 4) Describe the construction of an attribute.
- 5) Name the attribute in a link tag, **a**, that is used to provide the address.
- 6) Name the attribute in a image tag, **img**, that is used to provide the path of an image?

Tags – common, (part 2)

In this chapter we will go through the most common tags and we will take a deeper look at tags. As you might discovered a tag starts with an opening tag and ends with a closing tag. That can be called with a single word – an **element**. HTML documents are defined by HTML elements. **Element content** is everything you put in-between a tag. As we already said in previous chapter, some elements might have an empty content, such as **img** tags. Elements like this are closed in the start tag. Almost all HTML elements can have **attributes**.

Let us now start with the common elements, such as **Heading**, **Paragraph**. A heading contains a series of elements. It depends on how big the heading should be, for example:

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
```

→

Heading 1
Heading 2
Heading 3

Note, the search engines use your data to collect information about your website.

You may continue till **h6!**

If you want to split up your paragraphs, you might use **Horizontal-Line**. The element name is therefore **HR**. This tag has no closing tag; the tag is closed already in the opening tag like this:

```
<h1>Heading 1</h1>
<hr />
<h2>Heading 2</h2>
<hr />
<h3>Heading 3</h3>
<hr />
```

→

Heading 1

Heading 2

Heading 3

The line will continue till the right side of the page.

Another element that is good to know is a **Paragraph** tag. It is defined by a **P**.

```
<p>
My first paragraph!
</p>
```

→

My first paragraph!

Nothing strange!

If you add another paragraph, as shown bellow, you will get a certain distance between them.

```
<p>
My first paragraph!
</p>
<p>
My second paragraph!
</p>
```

→

My first paragraph!
My second paragraph!

However, there will be a little problem with the space inside the document.

```
<p>
My first paragraph!
This is the second line.
</p>
```

→ My first paragraph! This is the second line.

As you might already see, the space inside the document will not affect the page. We can solve this issue in two ways, though. The basic way of solving this problem is to add a line **Break** tag, which is defined by **BR**.

```
<p>
My first paragraph! <br />
This is the second line.
</p>
```

→ My first paragraph!
This is the second line.

Another way of solving the space problem is to use an element called **preformatted text**, defined as **PRE**. The content inside this element will preserve all line breaks and spaces. The font style will usually be *Courier*.

```
<pre>
My first paragraph!
This is the second line.
</pre>
```

→ My first paragraph!
This is the second line.

The only change was in the font style, however. No more tags are required to tell the browser that there are line breaks, and spaces inside the document.

Note: If you want to add a space (usually more than one between two words/letters), use ` ` as a char for space. For example, `<p>Hello Guest!</p>`. The total amount of spaces between Hello and Guest will be three.

Char is a shorter version of character. In almost all programming languages, it defines all letters, symbols, and digits, for example, A, @, 3.

Try to experiment with this tags that we have gone through in this chapter. Elements are located below:

- **h1 – h6;** *Heading.*
- **p;** *Paragraph.*
- **br;** *Line break.*
- **pre;** *Preformatted text.*
- ** ** *Additional space*

- 1) Define the heading element.
- 2) Which kinds of heading element are there? Name all of them.
- 3) Define the paragraph element.
- 4) What is the function of line break tag (**BR**)?
- 5) What is the function of preformatted tag (**PRE**), and which differences in style are there?
- 6) Describe as much as you can the ` ` function. What is the main idea behind it?

Tables – their function, (part 1)

In this chapter we will define the table element and what it can be used for. Let us start with the structure of them.

```
<table border="1">
  <tr>
    <td>row 1, cell 1</td>
    <td>row 1, cell 2</td>
  </tr>
  <tr>
    <td>row 2, cell 1</td>
    <td>row 2, cell 2</td>
  </tr>
</table>
```

(Internet Explorer 9)

(Chrome, Firefox)

The border can actually differ in each web-browser, but it might be fixed with CSS though. As it is shown above, the table consist of a **TABLE** element, a **TR** (defines a row) element, and a **TD** (defines a column).

If we want to assign a name to each column, we can do it by adding a header, defined as **TH**.

```
<table border="1">
  <tr>
    <th>Header 1</th>
    <th>Header 2</th>
  </tr>
  <tr>
    <td>row 1, cell 1</td>
    <td>row 1, cell 2</td>
  </tr>
  <tr>
    <td>row 2, cell 1</td>
    <td>row 2, cell 2</td>
  </tr>
</table>
```

(Chrome, Firefox)

A table might be used in several ways: either to create *a simple table*, or a *layout of a page*. It is strongly recommended to use CSS; however, we will go through the way of doing it with tables in the second part of this chapter.

As a conclusion of this chapter, please remember, before you publish your page; use several major browsers to see how they will show your page!

- 1) Name all tags that have to do with tables.
- 2) Why is it good to test your page in all major browsers?
- 3) How do you add a header, define the tag
- 4) Name the way of using tables.

Lists – another good tag

A list is used to simply create a list where you put your stuff. There are different kinds of lists. Let us look at an **unordered list** first, **UL**. Inside the list you might put list items, **LI**.

<pre>Hobbies Football Basket Box </pre>	→	Hobbies <ul style="list-style-type: none">• Football• Basket• Box	A simple, unordered list of hobbies
--	---	---	-------------------------------------

However, if we want to add an **ordered list**, **OL**, we just change the **UL** to **OL**.

<pre>Hobbies Football Basket Box </pre>	→	Hobbies <ol style="list-style-type: none">1. Football2. Basket3. Box	A simple, ordered list of hobbies
--	---	--	-----------------------------------

There is also another kind of lists, a **definition list**, **DL**. Inside a definition list we have two kinds of tags, a **DT**, which defines the item in the list, and **DD**, which describes the item in the list.

<pre>Hobbies <dl> <dt>Football</dt> <dd>Is a sport where the idea is to hunt the ball, and earn points</dd> <dt>Basket</dt> <dd>Is a sport where you have to do the same as in Football, but, there are some differences</dd> <dt>Box</dt> <dd>It is a brutal sport, and maybe dangerous.</dd> </dl></pre>	→	Hobbies Football Is a sport where the idea is to hunt the ball, and earn points Basket Is a sport where you have to do the same as in Football, but, there are some differences Box It is a brutal sport, and maybe dangerous.
--	---	---

- 1) What is the difference between an unordered list and ordered list?
- 2) How to define an ordered and unordered list?
- 3) How to define a list item.
- 4) What is so special about a definition list?
- 5) Which element is used to define an item in the list?
- 6) Which element is used to describe an item in the list?

Summary of elements

Name	Tag	Example
Link	a	<code>My link</code>
Image	img	<code></code>
Paragraph	p	<code><p>content</p></code>
Heading	h1,h2,h3,h4,h5,h6	<code><h1>My Heading</h1></code>
Line break	br	Hello <code>
</code> World!
Horizontal line	hr	<code><hr /></code>
Preformatted text	pre	<code><pre>text here</pre></code>
Table	table	<code><table border="1"> <tr> <th>Header 1</th> <th>Header 2</th> </tr> <tr> <td>row 1, cell 1</td> <td>row 1, cell 2</td> </tr> <tr> <td>row 2, cell 1</td> <td>row 2, cell 2</td> </tr> </table></code>
Table row	tr	See table
Table column	td	See table
Table heading	th	See table
Unordered list	ul	<code> Football Basket Box </code>
List item	li	See unordered list
Ordered list	ol	<code> Football Basket Box </code>
Definition list	dl	Hobbies <code><dl> <dt>Football</dt> <dd>Is a sport where the idea is to hunt the ball, and earn points</dd> <dt>Basket</dt> <dd>Is a sport where you have to do the same as in Football, but, there are some differences</dd> <dt>Box</dt> <dd>It is a brutal sport, and maybe dangerous.</dd> </dl></code>
Definition of list item	dt	See definition list
Description of list item	dd	See definition list

This was a short overview of HTML. For more info: <http://www.w3schools.com/html/default.asp>