**HTML Concept**

HTML (Hyper Text Markup Language) is a markup language for describing web pages. HTML documents are simple text files, described by HTML tags, which are standard keywords (tag names), surrounded by angle brackets: `<tag>`, e.g. `<body>`, `<table>`, `<img>`.

With few exceptions HTML tags come in pairs, called start and end tags. The end tag has a slash before the tag name: e.g. `<tag>...</tag>`. Tag interpretation goes like this: all the content encapsulated by the start and end tag inherits the properties of the tag.

Note that unpaired tags like `<img>, `<br>, `<hr>, etc (will discuss them later) are standalone objects, they do not apply any property to other content, so they do not have a stop tag.

Tags have default properties called attributes. We can modify a default attribute value of a tag by adding the attribute with its new value [surrounded by quotation marks] within the start tag: `<tag attribute="new value">...</tag>`. Remember that attributes go in the start tag and NEVER in the stop tag.

To conclude: we built HTML with tags (and their attributes) that “explains” the browser how we want the information to be presented.

**File setup**

In order to complete our HTML lesson you will need to use 3 programs that are already installed on our Windows PC:

1. Windows explorer – used for creating / modifying / moving files and folders
2. Notepad – used for modifying the HTML files content
3. A Browser (Internet Explorer, Chrome, Firefox) – used for visualization of the HTML files.

First, run Windows Explorer and create a folder called “startHTML” wherever you consider. This is going to be our working folder. In our example we create the folder directly on disc D “D:/startHTML/”.

Second, run Notepad (Start-> Accessories->Notepad). From the main menu select “File-> Save as” command. In the pop-up window that opens in the “Save as type” select “All Files” option and write in the “File name” box “page1.html”. In the “Save in” option select the folder you have created in the first step (D:/startHTML/), then click the “Save” button. This will save your first, empty HTML file. Note that the file name can be written with up case or down case letters and that the file extension can either be “.htm” or “.html”. We are going to use the html extension in our examples.

Third, return to Windows Explorer and double click the new created file. By now its icon should have the appearance of the default browser. The file opens within the default browser.
First page

Copy the following example in the notepad file we've just created.

```html
<!DOCTYPE html>
<html>
<head>
<title>My first HTML file</title>
</head>
<body>
My first HTML content.
</body>
</html>
```

Save the file “File->Save”, then move to (click/focus) the browser window and click the refresh/reload button or push the F5 button on your keyboard. This will reload the file with its new saved content. From now on we are going to call this operation “Save and Refresh”.

The result should look like this:

Now let’s try and interpret ourselves the HTML code like the browser dose.
The first (unpaired) tag, `<DOCTYPE html>`, tells the browser that we are reading a HTML file. We could determine this by the file extension, but, then again, some program could just mix up file extensions.

The `<html>`...

Within the html tags there are two sections: head and body.

Note: the head section contains meta-information (information that will not be displayed in the page content). In our example it contains only the title tag which is the actual browser’s window title.

Let’s continue with our encapsulating technique. So we have an html section determined by the `<html>`...

Let’s continue with our encapsulating technique. So we have an html section determined by the `<html>`...

Note that the header section can contain other information that does not address the user but it addresses web crawlers (computer programs) that also surf the internet.

The body section contains the information that will be displayed (if written in a correct HTML manner). In our example the body contains only some plain text.

Everything that we are going to display in the next examples will stand between the `<body>`...

This concludes are first cheaper: we’ve learned how to create a HTML file, how to display it in browser, the basic structure of a HTML document and the encapsulating concept of the HTML.

Text formatting

In this chapter we will learn how to format the text: paragraphs, breaks, headings, bold, italic, underline, true type and preformed text.

To get started let’s create a second file named “page2.htm”, copy the following example, then Save and Reload.

```html
<DOCTYPE html>
<html>
    <head>
        <title>Lacul de Mihai Eminescu</title>
    </head>
    <body>
        LACUL
        Mihai Eminescu
        Lacul codrilor albastru
        Nuferi galbeni il incarca;
        Tresarind in cercuri albe
        El cutremura o barca.
        Si au trec de-a lung de maluri,
        Parc-ascult si parc-astept
        Eadin trestii sa rasara
        Si sa-mi cada lin pe piept;
        Sa sarim in luntrea mica,
    </body>
</html>
```
Ingâñati de glas de ape,
Și sa scap din mâna cărma,
Și lopetele sa-n scape.

Sa plutim cuprinsii de farmec
Sub lumina blândeii lune
Vântul-n trestii lin fosnească,
Undulioasa apa sune.

Dar nu vine... Singuratic
În zadar suspin si sufar
Lânga lacul cel albastru
Încarcat cu flori de nufar.

(1876/10/01)
</body>
</html>

The result looks like this:

Note that the window title is changed.

In order to display as much information as possible browsers do not interpret new lines (enter key) and do not display more than one space between the words (space key).

Let’s structure the text in paragraphs (def: A paragraph (from the Greek paragraphos, “to write beside” or “written beside”) is a group of sentences with one topic. It is also a self-contained unit of a discourse in writing dealing with a particular point or idea. A paragraph consists of one or more sentences.). So we shall consider each strophe as a paragraph. The HTML tag for paragraphs is `<p>`...`</p>`, and its interpretation is: everything that is between the start and end tag is part of the same paragraph.

Now embrace every strophe with the paragraph tags, than Save and Refresh

E.g.

...

<p>Lacul codrilor albastru
Nuferi galbeni îi încarca,
Tresarind în cercurii albe
El cutremura o barca.</p>
The result should look like this.

Though each strophe is separated the rows are not. We’ll have to use a break tag that tells the browser to go on the next line after each row. The break tag is `<br>` and because it is a standalone object (a break) it does not have a pair tag. Add a `<br>` tag at the end of each row. Save and Refresh.

E.g.

```
<p>Lacul codrilor albastru<br>
Nuferi galbeni il incarca; <br>
Tresarind in cercuri albe<br>
El cutremura o barca.</p>
```

The result should look like this.
HTML has 6 levels of headings tags: `<h1>.../h1>`, `<h2>.../h2>`, `<h3>.../h3>`, `<h4>.../h4>`, `<h5>.../h5>`, `<h6>.../h6>.
Most important heading is `<h1>` and its importance decreases to `<h6>`. Let's make the poem title LACUL the most important heading and Mihai Eminescu the second most important heading. Save and Refresh.

E.g.

```
...<h1>LACUL</h1>
<h2>Mihai Eminescu</h2>
```

The result should look like this. Note that it is not necessarily to add a break at the end of a heading.

```
<p>Lacul</p> <i>codrilor</i> <u>albastru</u><br />
<tt>Nuferi</tt> galbeni îl încarca, <br />
Tresarind în cercuri albe<br />
El <i>cutremura ox</i> <b>barca</b></p>
```

The result should look like this.
Just to have a final check the HTML file content should be:

```html
<!DOCTYPE html>
<html>
<head>
<title>LACUL de Mihai Eminescu</title>
</head>
<body>
<h1>LACUL</h1>
<h2>Mihai Eminescu</h2>
<p>În codriilor</p>
<p>În albastru</p>
<p>Tresarind în cercuri albe</p>
<p>În cutremura a o barca</p>
<p>În albastru</p>
<p>În lung de maluri</p>
<p>Parc-ascult și parc-astept</p>
<p>Ea din trestii sa rasara</p>
<p>Si sa-mi cada lin pe piept</p>
<p>În luntrea mica</p>
<p>Îngânați de glas de ape</p>
<p>Si sa scap din mâna cărma</p>
<p>Si iopetile sa-mi scape</p>
<p>Sub lumina blânde iune</p>
<p>Vântul tresti lin fosneaaca</p>
<p>Unde foios apa sunet</p>
<p>Dar nu vine...</p>
<p>Singuratic</p>
<p>În zadar suspin si sufar</p>
<p>Lângă lacul cel albastru</p>
```
Incarcat cu flori de nufer.<br/>

[1876/10/01]
</body>
</html>

Links

Links are the spark of web pages. With a single click you can move to another page... and again and again. They are represented by special appearance text with different mouse-over cursor.

The tag name for a link is `<a>`...`</a>` and it is the abbreviation of anchor. With the attribute “href” (abbreviation from hyper reference) we change the destination resource (web page/ other file). Between the start and the end tag stands the visible object/objects, that become links to the hyper reference.

Create a new file named “page3.htm”, copy the following example, then Save and Reload.

```html
<!DOCTYPE html>
<html>
<head>
    <title>My first LINK example</title>
</head>
<body>
    <a href="page2.html">Visible (clickable text)</a>
</body>
</html>
```

The result should look like this:

![Image of a browser window showing a link to page2.html](image)

So the visible part of the link is between the start and end tags. The target page is specified by the href attribute, and in our case it takes us to the text formatting page example.

Fast exercise: modify page2.html; the date of the poem (1876/10/01) should become a link to page3.html

Lists

Lists are less important for humans, but they are very important to search engines. Within the same list there must be objects of the same type.
There are two types of lists: ordered lists (the position in the list is meaningful) and unordered lists. Different degrees of a disease should stay in an ordered list, but the possible color of eyes should stay in an unordered list.

The tags for lists are <ol>...</ol> and <ul>...</ul> standing from abbreviation from order list and unordered list. Inside each list there are tags that specify items <li>...</li>, abbreviation from list item. Note that the list item tag is the same for both types of lists.

Create a new file named “page4.htm”, copy the following example, then Save and Reload.

```html
<DOCTYPE html>
<html>
  <head>
    <title>Lists</title>
  </head>
  <body>
    <ol>
      <li>First stage</li>
      <li>Second stage</li>
      <li>Third stage</li>
    </ol>
    <ul>
      <li>black</li>
      <li>green</li>
      <li>blue</li>
    </ul>
  </body>
</html>
```

The result should look like this

Note that HTML automatically adds numbers before each item in an order list.

Images

As said in the introduction the tag name for images is <img>. It is an unpaired tag, a standalone object that does not alter/add properties of other elements.
With the tag an attribute specifies the file that will be loaded and displayed. The attribute is “src” and it is the abbreviation of source.

There are two types of source specification: explicit and implicit.

The implicit specification of source file demands that the file should be copied in the same directory with the HTML file.

The explicit specification of source file demands the full pass of the file. So instead of copying the file to the working folder, we specify the full disk path.

To better understand the concept let’s take the Desert.jpg file from “My Computer -> Pictures -> Sample Pictures -> Desert.jpg” and copy it to our working folder.

Create a new file named “page5.htm”, copy the following example, then Save and Reload.

```html
<DOCTYPE html>
<html>
<head>
<title>image manipulation</title>
</head>
<body>
    Implicit <img src="Desert.jpg"><br>
    Explicit <img src="C:\Users\Public\Pictures\Sample Pictures\Desert.jpg"><br>
</body>
</html>
```

The result should look like this

![Image of Desert.jpg](file:///D:/startHTML/page5.htm)

Note that the files are the same though they are loaded from different disk locations. If you delete the file in our current folder and reload the page you will see and error message.

It seems that the images are too big. We can resize them with two attributes width="100px" and height="80px". Remember that attributes always stand in the start tag: `<img src="Desert.jpg" width="100px" height="80px">`. Add the attributes to each image. Save and Reload.

The result should look like this

![Image of Desert.jpg resized](file:///D:/startHTML/page5.htm)
Tables

Tables are a little more complicated. The tag name for table is <table>...</table>. Who would have guessed? Like in list this declaration this is of an empty table. Next we have to add rows and columns. Exactly in this order: first a row, and inside a row columns; then another row. The row tag name is <tr>...</tr> and it is the abbreviation from table row. The column tag name for columns is <td>...</td> and it is the abbreviation for column data.

Create a new file named "page6.htm", copy the following example, then Save and Reload.

```
<html>
<head>
    <title>Tables. Complicated...</title>
</head>
<body>
    <table>
        <tr>
            <td>First cell</td>
            <td>Second cell</td>
        </tr>
        <tr>
            <td>Third cell</td>
            <td>Forth cell</td>
        </tr>
    </table>
</body>
</html>
```

The result should look like this. If it doesn’t add the border="1" attribute and value to the table start tag. Default value for borders is 0.
Exercises

Create three HTML files named 1.html, 2.html and 3.html. In 1.html file has the window title "Lists" and the h1 level page title “A list with 3 elements”. It contains an unordered list with three elements, the third being a link to the file 2.html. The 2.html file will contain a full table of X and 0. The last cell will contain a link to the file 3.html. We have the window title “X and 0” and no page title. In the 3.html file will have a table with three rows and three columns, each cell will contain the line number and column number. Last cell will contain an image (Desert.jpg) link to the 1.html file. Window title will be "Links with images" and the title page, "Back to the first page" of h3 level.

File 1.html

<html>
<head>
<title>Lists</title>
</head>
<body>
<h1>A list with 3 elements</h1>
<ul>
<li>first</li>
<li>second</li>
<li><a href="2.html">third</a></li>
</ul>
</body>
</html>
File 2.html

<html>
<head>
<title>X and 0</title>
</head>
<body>
<table border="1">
<tr>
<td>x</td>
<td>O</td>
</tr>
<tr>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>x</td>
<td>O</td>
</tr>
<tr>
<td>O</td>
<td>x</td>
<td><a href="3.html">x</a></td>
</tr>
</table>
</body>
</html>
<html>
<head>
<title>Links with images</title>
</head>
<body>
<h3>Back to the first page</h3>
<table border="1">
<tr>
<td>L1 - C1</td>
<td>L1 - C2</td>
<td>L1 - C3</td>
</tr>
<tr>
<td>L2 - C1</td>
<td>L2 - C2</td>
<td>L2 - C3</td>
</tr>
<tr>
<td>L3 - C1</td>
<td>L3 - C2</td>
<td><a href="1.html"><img src="Desert.jpg" width="50px" height="50px"></a></td>
</tr>
</table>
</body>
</html>