



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Web and Apps

1) HTML - CSS

Emmanuel Benoist
Spring Term 2021

HyperText Markup Language and Cascading Style Sheets

- Introduction
- Forms and Input
Tables
- Headings and Styles Sheets
- CSS Basic formatting

Introduction

HTML is a Markup Language

- ▶ **Tags:**

- ▶ A command can be composed of two tags: `<h1>` and `</h1>`
- ▶ Or one single tag: ``

- ▶ **XHTML Syntax:**

- ▶ In XHTML (like in any XML), tags must be written in lower case, they must always terminate, arguments must be enclosed in "
- ▶ Single tags must be written like: `
` (self closing).

Structure of an HTML document

- ▶ **HTML Structure**

```
<html>  
<head> ... </head>  
<body> ...</body>  
</html>
```

- ▶ **Content of the head: Meta-information**

- ▶ Title, Author, Keywords, Abstract, Javascript and CSS files

- ▶ **Content of the body: Information to be displayed**

Optimize Header Content

- ▶ **Title: Appears in the window status bar**
- ▶ **Encoding to be used**
- ▶ **Program used to generate the file**
- ▶ **CSS and JavaScript to be integrated in the page.**

```
<head>  
  <meta charset="UTF-8"/>  
  <title>My Web Site</title>  
  <link href="style.css" type="text/css" rel="stylesheet">  
  <meta name="generator" content="MediaWiki 1.31.0-wmf.20"/>  
  <script> ...</script>  
  ...  
</head>
```

Content of the Body

- ▶ **Everything that needs to be displayed (almost everything)**
 - ▶ Headings, paragraphes, texts, images, tables,
- ▶ **The basic Document Object Model (DOM)**
 - ▶ Represents a tree, each tag (pair) is a node, texts are leaves.
 - ▶ This Tree can be manipulated in Javascript (deletion, insertion or modification of nodes),
 - ▶ Contains place holders (not displayed unless activated by javascript)

Forms and Input

Forms and Input

- ▶ **Input Fields have to be incorporated in a form tag**
 - ▶ A Form has: a method (GET or POST) an id (to be manipulated by javascript), an action (where to send the request).
- ▶ **Input fields**
 - ▶ Hidden: are not viewable (neither modifiable normally)
 - ▶ Text: To input text on one line
 - ▶ Password: to input text that can not be displayed (*****)
 - ▶ Radiobuttons: choose one button from a list (enabling one disable the others)
 - ▶ Checkbox: Can be checked or unchecked (independently).
- ▶ **Other Fields**
 - ▶ Selection Box (select one among many)
 - ▶ Textarea: Type any text (more than one line)

Example

```
<form method="GET" action="forms.php">
```

- ▶ There is here a hidden input, which is not displayed:

```
<input type="hidden" name="somethingsecret" value="yes"/>
```

- ▶ Input type text:

```
<input type="text" name="textfield1" value="1" size="9"/>
```

- ▶ Input type password:

```
<input type="password" name="pwdfield1" value="123" />
```

- ▶ Input type Radiobutton:

```
1:<input type="radio" name="radio1" value="1" checked="true"/>
```

```
2:<input type="radio" name="radio1" value="2" />
```

```
3:<input type="radio" name="radio1" value="3" />
```

<https://www.benoist.ch/WebApps/examples/html-css/forms.php>

Example (Cont.)

▶ Selection Box:

```
<select name="selectionfield1">  
  <option value="7">Tous</option>  
  <option value="1">Gare/Arr&#234;t</option>  
  <option value="2">Lieu,rue,num&#233;ro</option>  
  <option value="4">Tourisme</option>  
</select>
```

▶ Another Select (in a scrolling list)

```
<select name="select2" size="5" >  
  <option value="6">Auto</option>  
  <option value="3">Autor</option>  
  ...  
</select>
```

▶ A multi-select (all the values are transferred to the server)

```
<select multiple="1" name="multiselect" size="3">  
  <option value="Less_than_1_year.">Less than 1 year.</option>  
  <option value="1-5_years.">1-5 years.</option>
```

Example (Cont.)

- ▶ **Textarea**

```
<textarea name="textareafield">
```

This is the default value

```
</textarea>
```

- ▶ **Buttons (for Javascript) and submit (to send the request)**

```
<input type="button" value="test" name="btn1" />
```

```
<input type="submit" value="OK" name="send" />
```

- ▶ **Image (the coordinates of the click are sent)**

```
<input type="image" src="imgmap.gif" name="img">
```

Tables

Tables

- ▶ **A table contains rows and columns**
 - ▶ Two sort of rows: Headers `<th>` and normal rows `<tr>`
 - ▶ Data are stored in columns: `<td>`
- ▶ **Standard table**

```
<table>
<tr><th>Year</th><th>Warmest Month</th><th>Temp.</th> \
→ </tr>
<tr><td>2006</td><td>June</td><td>24</td></tr>
<tr><td>2007</td><td>August</td><td>27</td></tr>
<tr><td>2008</td><td>July</td><td>31</td></tr>
<tr><td>2009</td><td>June</td><td>29</td></tr>
</table>
```

https:

[//www.benoist.ch/WebApps/examples/html-css/tables.php](https://www.benoist.ch/WebApps/examples/html-css/tables.php)

Tables Cont.

► One can merge cells

- `colspan` merges the cells on multiple columns
- `rowspan` merges the cells on multiple lines

```
<table>
<tr><th>&nbsp;</th><th colspan="2">Result</th></tr>
<tr><th>Year</th><th>Warmest Month</th><th>Temp.</th>↵
→</tr>
<tr><td rowspan="2">2006</td><td>June</td><td>27</td>↵
→</tr>
<tr><td>August</td><td>27</td></tr>
<tr><td>2007</td><td>July</td><td>31</td></tr>
<tr><td>2008</td><td>June</td><td>29</td></tr>
</table>
```

Table with CSS

► One can define classes in a table

```
<table class="colored">
<tr><th>&nbsp;</th><th colspan="2">Result</th></tr>
<tr><th>Year</th><th>Warmest Month</th><th>Temp.</th>↵
→</tr>
<tr><td rowspan="2">2006</td><td>June</td><td>27</td>↵
→</tr>
<tr><td>August</td><td>27</td></tr>
<tr><td>2007</td><td>July</td><td class="red">31</td></tr>
→tr>
<tr><td>2008</td><td>June</td><td>29</td></tr>
</table>
```


Headings and Styles Sheets

Headings

- ▶ **Headings** `h?` are used to display titles, subtitles, ...

```
<h1>Heading level 1</h1>
```

```
<h2>Heading level 2</h2>
```

```
<p>...</p>
```

```
<h3>heading level 3</h3>
```

```
<p>...</p>
```

```
<h2>Heading level2 class "second"</h2>
```

- ▶ **Style may be given in a style sheet.**

- ▶ We need to import the style file (in the HTML head for instance):

```
<link rel="stylesheet" href="style.css" type="text/css" />
```

- ▶ We can define properties for some tags

```
body { font-family: helvetica, arial, sans-serif; }
```

```
h1 {color:red}
```

```
h2 {color:blue}
```

<https://www.benoist.ch/WebApps/examples/html-css/>

CSS Basic formatting

Classes for Style Sheets

- ▶ **One want to select some h1 or some h2, without defining the same for h1.**

- ▶ We add a class to the tag

```
<h2 class="second">Heading level2 class "second"</h2>
```

```
<p> ... </p>
```

```
<h3 class="second">heading level 3</h3>
```

```
<p> ... </p>
```

```
<h3 class="third">heading level 3</h3>
```

- ▶ we can define some properties for the different classes:

```
h2.second {color:#F0C000}
```

```
h3.second {color:#907000}
```

```
h3.third {color:#302500}
```

What sort of properties can we define?

▶ Background properties

- ▶ Color
- ▶ image
- ▶ Repeat or not
- ▶ Its place and the way it is defined

```
<head> ...
```

```
<style type="text/css">
```

```
body { background: #00ff00 url('smiley.gif') no-repeat fixed ↘  
→center; }
```

```
.leftmenu_servicebutton_ACT {  
    background-image: url(bg_menubutton18h.gif);  
    color: #003366; background-color: #dedcbd  
}
```

```
</style>
```

```
</head>
```

Text formatting

- ▶ **Font color definition**

```
.green {color: #00ff00}  
.violet {color: #dda0dd}  
.blue {color: rgb(0,0,255)}
```

- ▶ **Text align**

```
h1.center {text-align: center}  
h1.left {text-align: left}  
h1.right {text-align: right}
```

Text Transformations

▶ Upper/lowercases

p.uppercase {text-transform: uppercase}

p.lowercase {text-transform: lowercase}

p.capitalize {text-transform: capitalize}

▶ Lines (over, through, or under)

.overline {text-decoration: overline}

.linethrough {text-decoration: line-through}

.underline {text-decoration: underline}

Font definition

▶ Font

- ▶ Font Family (we should give many, such that one is installed on the client).
- ▶ Font-Size
- ▶ Font-weight (how bold it is)
- ▶ ...

```
p.font { font: italic small-caps 900 12px arial }  
td { font-size: 11px; margin: 0px; color: #000000;  
      font-family: Verdana, Geneva, Arial, Helvetica, sans-serif}  
input { font-size: 11px; margin: 0px; color: #000000;  
        font-family: Verdana, Geneva, Arial, Helvetica, sans-  
        serif }
```


Paragraph surroundings

- ▶ **Margins**

```
p.margin {margin: 2cm 4cm 3cm 4cm}
```

- ▶ **Border**

```
p.border { border: medium double rgb(250,0,255) }
```

- ▶ **Padding**

```
td.test1 {padding: 1.5cm}
```

```
td.test2 {padding: 0.5cm 2.5cm}
```

Boxes configuration

► Padding and margins

```
h1 {  
  padding-right: 2px; margin-top: 0px; padding-left: 0px;  
  font-weight: bold; font-size: 17px; margin-bottom: 9px;  
  color: #003366; padding-top: 0px; background-color: #dcedff  
}  
h2 {  
  padding-right: 2px; margin-top: 4px; padding-left: 0px;  
  font-weight: bold; font-size: 14px; margin-bottom: 6px;  
  color: #7079aa; padding-top: 0px; background-color: #dcedff  
}  
h3 {  
  padding-right: 2px; margin-top: 12px; padding-left: 0px;  
  margin-bottom: 2px; padding-top: 0px;  
}
```

List Style

- ▶ We can use any image as a line start in a unordered list (ul)

```
<style type="text/css">
ul { list-style: square inside url('arrow.gif') }
</style>
<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>
```

Conclusion

- ▶ **This was just an introduction**

- ▶ HTML is much more than 5 tags
- ▶ CSS is not just formatting of text and colors
- ▶ More about CSS will be seen later

- ▶ **Principles**

- ▶ HTML defines the content, without being interested in layout
- ▶ CSS defines the layout
- ▶ HTML may work with different CSS (for different media for instance)