
Markup Languages
General

Denis Helic

- ▶ Markup: additional information, how to interpret the text in a document
 - ▶ in all textprocessing applications
 - ▶ e.g. in L^AT_EX: `\section{title of section}`
 - ▶ or in HTML `<p>new paragraph, blabla...</p>`
 - ▶ or in RTF (StarOffice - “Hello World”):

```
{\rtf1\ansi\deff0{\fonttbl{\f0\froman\fprq2\fcharset0 Times;}}
{\colortbl\red0\green0\blue0;\red255\green255\blue255;\red128
\green128\blue128;}{\stylesheet{\s1\next1 Standard;}}
{\info{\comment StarWriter}{\vern5690}}\deftab720
[...]
\pard\plain \s1 Hello World
\par }
```
 - ▶ usually invisible to the user

- ▶ generalized/descriptive markups

```
<section id="sect1">  
  <p>this ia a paragraph</p>  
  <p>this ia another paragraph</p>  
  <p>this ia a third paragraph</p>  
</section>
```

- ▶ Structured as nested elements
- ▶ Element: markup (tags), attributes, content
 - ▶ Tags: start tag + end tag
 - ▶ Attributes: inside start tag
 - ▶ Content: Text + sub-elements

- ▶ generalized/descriptive markups
 - ▶ describe structure only
- ▶ separate content and presentation (Very important!)
 - ▶ consistency, easy to change
 - ▶ customizable (style sheets)

SGML - Standardized General Markup Language(1/7)

- ▶ SGML: quite old iso standard ISO8879 (1986)
- ▶ meta - markup language
 - ▶ allows other markup languages to be defined with it

SGML - Standardized General Markup Language(2/7)

- ▶ defines
 - ▶ basic syntax (how markup is distinguished from normal text)
 - ▶ elements
 - ▶ attributes
 - ▶ entities
- ▶ leaves interpretation open

SGML - Standardized General Markup Language(3/7)

- ▶ elements describe the structure of a document
- ▶ element is delimited with a start-tag and a corresponding end-tag
 - ▶ end-tags (not always :-)
 - ▶ nested

```
<p>
```

```
  this is a paragraph, that holds a quote
```

```
  <q>
```

```
    this is the quote
```

```
  </q>
```

```
  continue paragraph...
```

```
</p>
```

SGML - Standardized General Markup Language(4/7)

▶ attributes

- ▶ additional information like
`<note type="warning">`
 in case of emergency...
`</note>`

▶ entities

- ▶ SGML based on ASCII
- ▶ binary data coded as entities
`<!ENTITY figure1 SYSTEM "fig1.bmp" NDATA BMP>`
 [...]
`<figure entity="figure1">`

SGML - Standardized General Markup Language(5/7)

- ▶ document type definition (DTD)
- ▶ defines
 - ▶ which elements may be used
 - ▶ which elements are required
 - ▶ what are relationships between elements (e.g. nested elements)
 - ▶ attributes of elements
 - ▶ required attributes of elements

SGML - Standardized General Markup Language(6/7)

- ▶ different document types
- ▶ syntactical check against the basic SGML syntax
- ▶ syntactical check against the specific DTD syntax
 - ▶ *valid* document (DTD) vs. *wellformed* document (markup-syntax only)

SGML - Standardized General Markup Language(7/7)

- ▶ problems of SGML
 - ▶ very complex
 - ▶ omission of end-tags

Markup Languages
HTML

Denis Helic

HTML - HyperText Markup Language(1/3)

- ▶ 1990 Tim Berners-Lee (<http://www.w3.org/pub/WWW/People/Berners-Lee/>) used an SGML DTD
 - ▶ added images
 - ▶ added links
 - ▶ created HTML DTD

HTML - HyperText Markup Language(2/3)

- ▶ problems creating a standard
 - ▶ “Browser Wars” (Netscape vs. Microsoft)
- ▶ mixture of presentation and content
 - ▶ Adding presentation specific tags (font, center, ...), colors, ...
- ▶ Moved away from descriptive/generalized markup
- ▶ Web Consortium W3C tries to make standards: <http://www.w3.org/>

HTML - HyperText Markup Language(3/3)

- ▶ Today HTML 4.01
 - ▶ Strict DTD (without presentation elements)
`http://www.w3.org/TR/html401/strict.dtd`
 - ▶ Transitional DTD (with presentation elements as transition when stylesheets are better supported)
`http://www.w3.org/TR/html401/loose.dtd`
 - ▶ Frameset DTD (defines framesets)
`http://www.w3.org/TR/html401/frameset.dtd`

▶ Here just a short overview!


▶ An excellent tutorial:

`http://courses.iicm.edu/mmis/selfhtml80/`

▶ start-tag <tagname>

▶ end-tag </tagname>

▶ case insensitive

 attributes

```
<tagname attribute_key="value" >
```

```
<tagname attribute_key=value>
```

Example:

```
<h4 align=center>
```

```
    HTML - the language of the WWW
```


```
</h4>
```

- ▶ start with '&', ends with semicolon ';'
- ▶ unicode-number or name/abbreviation of character
- ▶ different HTML-versions (3.2/4.0)

char	description	name	unicode	from version
á	a with accent acute	<code>&aacute;</code>	<code>&#225;</code>	3.2
â	a with circumflex	<code>&acirc;</code>	<code>&#226;</code>	3.2
ä	a with Umlaut	<code>&auml;</code>	<code>&#228;</code>	3.2
α	alpha	<code>&alpha;</code>	<code>&#945;</code>	4.0
...				
∀	for all	<code>&forall;</code>	<code>&#8704;</code>	4.0
...				

- ▶ complete list:
<http://courses.iicm.edu/mmis/selfhtml80/html/referenz/zeichen.htm>

HTML - The very short overview(1/9)

 HTML-comments (comp. CSS-comments)

```
<!-- this is a comment -->
```

HTML - The very short overview(2/9)

▶ minimum HTML

```
<HTML>
  <HEAD>
    <TITLE>
      title of dokument
    </TITLE>
  </HEAD>
  <BODY>
    content of document
  </BODY>
</HTML>
```

▶ Example:

```
http://coronet.iicm.edu/mmis/examples/html_css/html/
minimum.html
```

HTML - The very short overview(3/9)

▶ tables (often (mis)used to layout the page)

▶ Tutorial:

`http://courses.iicm.edu/mmis/selfhtml80/html/tabellen/aufbau.htm`

▶ Example:

`http://coronet.iicm.edu/mmis/examples/html_css/html/table.html`

HTML - The very short overview(4/9)

▶ forms - enable user interaction

▶ send as mail (depends on browser settings!)

▶ send to (CGI (Common Gateway Interface))-application

▶ Tutorial:

`http://courses.iicm.edu/mmis/selfhtml80/html/formulare/definieren.htm`

▶ Example:

`http://coronet.iicm.edu/mmis/examples/html_css/html/form.html`

HTML - The very short overview(5/9)

 colors

 Tutorial:

`http://courses.iicm.edu/mmis/selfhtml80/html/allgemein/farben.htm`

HTML - The very short overview(6/9)

▶ embed (multimedia) objects

▶ Tutorial:

`http://courses.iicm.edu/mmis/selfhtml80/html/multimedia/objekte.htm`

▶ Example (incl. alternative):

```
<object data="film.avi" type="video/x-msvideo">  
    
</object>
```

HTML - The very short overview(7/9)

▶ java applets, flash animations

▶ ActiveX controls

▶ image maps

▶ very browser specific (Netscape does not like them!)

▶ Example:

```
http://coronet.iicm.edu/mmis/examples/html_css/html/  
applet.html
```

▶ Example:

```
http://coronet.iicm.edu/mmis/examples/html_css/html/  
object.html
```

HTML - The very short overview(8/9)

- ▶ frames allow multiple views, e.g. navigation part, content part, etc.
- ▶ Tutorial:
`http://courses.iicm.edu/mmis/selfhtml80/html/frames/definieren.htm`
- ▶ define with frameset and a number of frames

HTML - The very short overview(9/9)

▶ disadvantages of frames

- ▶ browser not capable (now rare!) (<noframes>)
- ▶ bookmarking
- ▶ search engines (<noframes>)

▶ advantages

- ▶ navigation (or other content) always on screen
- ▶ create art-work :-)

▶ Example:

`http://coronet.iicm.edu/mmis/examples/html_css/html/frames.html`

HTML - client-side scripting(1/2)

▶ JavaScript

▶ Tutorial:

`http://courses.iicm.edu/mmis/selfhtml80/javascript/index.htm`

▶ embedded in HTML-code

▶ active component in HTML-pages

▶ functionality

▶ design

HTML - client-side scripting(2/2)

- ▶ Examples:
`http://courses.iicm.edu/mmis/selfhtml80/javascript/beispiele/index.htm`
- ▶ offer `<noscript>` parts
- ▶ Document Object Model (DOM)
- ▶ browser dependent (big problem!)

- ▶ better control for formatting and layout of HTML-document
- ▶ better separation of content and presentation
- ▶ Cascading Style Sheets (CSS)
- ▶ check browser support:
<http://www.quirksmode.org/css/contents.html>

CSS - Cascading Style Sheets(1/4)

- ▶ Specifications (approved by Web Consortium)
 - ▶ CSS1 - supported by all browsers
 - ▶ CSS2 - almost completely supported
 - ▶ CSS3 - W3C works on it right now

CSS - Cascading Style Sheets(2/4)

- ▶ Style sheet is a set of instructions which tell browser how to draw a particular HTML element
- ▶ Instructions in CSS - Statements

Example of style statements

```
<style type="text/css">
  p.normal { font-size:10pt; color:black; }
  p.gross { font-size:12pt; color:black; }
  p.klein { font-size:8pt; color:black; }
  all.rot { color:red; }
  .blau { color:blue; }

  /* do not show menu on print: */
  @media print {
    .menu {display: none;}
  }
</style>
```

- ▶ Statement:
 - ▶ Selector (which element)
 - ▶ Declaration (which properties to apply for that element)

- ▶ embedding style into HTML pages

- ▶ special HTML element <style>

- ▶ Example:

- `http://coronet.iicm.edu/mmis/examples/html_css/embed/min_inline.html`

- ▶ as an attribute of an HTML element

- ▶ Example:

- `http://coronet.iicm.edu/mmis/examples/html_css/embed/min_tag_inline.html`

▶ external document <link>

▶ Example:

```
http://coronet.iicm.edu/mmis/examples/html_css/embed/  
min_extern.html
```

▶ Example:

```
http://coronet.iicm.edu/mmis/examples/html_css/embed/  
simple.css
```

▶ external document: import

▶ Example:

```
http://coronet.iicm.edu/mmis/examples/html_css/embed/  
min_import.html
```

▶ Tutorial:

```
http://courses.iicm.edu/mmis/selfhtml80/css/formate/  
einbinden.htm
```

type selectors

```
<style type="text/css">  
  p { font-size:10pt; color:black; }  
</style>
```

class selectors

```
<style type="text/css">
  p.normal { font-size:10pt; color:black; }
  p.gross { font-size:12pt; color:black; }
  p.klein { font-size:8pt; color:black; }
  all.rot { color:red; }
  .blau { color:blue; }
</style>
```

example

```
<p class="menu">Navigations Menu: etc.</p>
```

```
<p class="normal">Normaler Textabsatz mit Schrift 10  
Punkt schwarz</p>
```

```
<p class="gross">Textabsatz mit Schrift 12  
Punkt schwarz</p>
```

```
<p class="klein">Textabsatz mit Schrift 8  
Punkt schwarz</p>
```

 nested type selectors

```
<style type="text/css">
  h1 { color:red; }
  h1 i { color:blue; font-weight:normal; }
</style>
```

independent selectors

```
<style type="text/css">
    #fettkursiv { font-weight:bold; font-style:italic; }
</style>
[...]
```

`<p id="fettkursiv">`
Extra-Formatierung
`</p>`

`<p>`
Das ist `<em id="fettkursiv">formatierter` Text.
`</p>`

attribute dependent selectors (CSS2)

```
<style type="text/css">
  p { font-weight:bold; font-family:Tahoma,sans-serif;
      font-size:14pt; }
  p[align] { color:red; }
  p[align=center] { color:blue; text-align:left; }
  td[abbr~=Berlin] { background-color:#FFFF00 }
  *[lang|=en] { background-color:#FF0000; color:#FFFFFF; }
</style>
```

pseudo selectors

```
<style type="text/css">
  a:link { color:#FF0000; font-weight:bold }
  a:visited { color:#990000; }
  a:active { color:#0000FF; font-style:italic }
</style>
```

▶ format text with ``:

```
<p>
```

```
  normal paragraph with
```

```
  <span style="font-style:italic; color:red;">
```

```
    red italic text
```

```
  </span>
```

```
</p>
```

▶ format text with <div>

```
<div class="blue">
```

```
    normal paragraph with blue (probably!) text
```

```
</div>
```

▶ Example:

```
http://coronet.iicm.edu/mmis/examples/html_css/  
selectors/mmis.html
```

▶ Example:

```
http://coronet.iicm.edu/mmis/examples/html_css/  
selectors/selectors.css
```

▶ Tutorial:

```
http://courses.iicm.edu/mmis/selfhtml80/css/formate/  
zentrale.htm
```

CSS - Alternative styles and media(1/2)

- ▶ Provide alternative styles

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

```
<link rel = "alternate stylesheet" type = "text/css"  
      title = "Alternate Page Style" href = "alternate.css">
```

- ▶ Example:

```
http://coronet.iicm.edu/mmis/examples/html_css/  
selectors/mmis.html
```

- ▶ Example:

```
http://coronet.iicm.edu/mmis/examples/html_css/  
selectors/alternate.css
```

CSS - Alternative styles and media(2/2)

- ▶ Provide alternative output media

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

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

- ▶ Example:

```
http://coronet.iicm.edu/mmis/examples/html_css/  
selectors/mmis.html
```

- ▶ Example:

```
http://coronet.iicm.edu/mmis/examples/html_css/  
selectors/print.css
```

- ▶ Tutorial:

```
http://courses.iicm.edu/mmis/selfhtml80/css/formate/  
einbinden.htm#link_media
```

CSS - Declarations/Apply Properties(1/3)

- ▶ Which properties can I influence with CSS?
- ▶ Fonts
 - ▶ size
 - ▶ family
 - ▶ line-height
 - ▶ alignment (left, right, center)
 - ▶ color
 - ▶ decoration (none, underline, line-through, ...)

CSS - Declarations/Apply Properties(2/3)

- ▶ page layout
 - ▶ page size
 - ▶ left/right pages
 - ▶ page breaks
- ▶ colors, borders, etc.

CSS - Declarations/Apply Properties(3/3)

▶ Example:

`http://coronet.iicm.edu/mmis/examples/html_css/site/
frames.html`

▶ Tutorial:

`http://courses.iicm.edu/mmis/selfhtml80/css/eigenschaften/
index.htm`