

Windows-Automatisation 2

Markup Languages, Document Object Model

Business Programming 1

Business Programming 2



Basics,
Parsing

Commands,
APIs

Window-
Automatisation,
Web-Scripting

Security,
Debugging

Graphical User
Interfaces (GUI),
Sockets,
...

Markup Languages, 1



- Tag

- Enables one to use tags to enclose ("markup") plain text

- Opening tag (a.k.a. start tag)

- `<some_tag_name>`

- Closing tag (a.k.a. end tag)

- `</some_tag_name>`

- An opening tag may contain attributes (`attributeName="value"`)

- Makes it easy to parse text and denote which tag was used to mark it up

- Element

- The sequence "opening tag", text, "closing tag"



- **Document Type Definition (DTD)**
 - Defines the tags and their attributes, if any
 - Name (identifier) of the tag
 - Attributes for tags
 - "Content model"
 - Nesting of tags and the allowed sequence of tags
 - **Hierarchical structure !**
 - Allows to determine how many times an element may occur
 - "Instance" of a DTD
 - A document with text that got marked-up according to the rules defined in a DTD
 - A document that has been checked whether the DTD rules were applied correctly is named a "**valid**" document

Markup Languages, 3



- **HyperText Markup Language (HTML)**
 - A markup language for the WWW
 - HTML-Browser
 - Parses a document marked up according to HTML
 - Formats the text, depending on the used tags
- **Document Type Definition (DTD)**
 - Version 4.01: three variants defined
 - **SGML**-based, hence it is possible to
 - Use any case for the tags and attribute names
 - Some closing tags can be omitted, if the end tags could be determined by the rules set forth in the DTD
 - It is possible to define exclusions



- Extensible Markup Language (XML)
 - A simplified version of SGML
 - Allows the definition of DTDs for markup languages
 - Since 2002 an alternative got introduced in the form of "XML Schema":
<http://www.w3c.org>
 - Tag and attribute names must be written in exact case
 - End tags must always be given
 - Attribute values can now be enclosed within apostrophes/single quotes (') in addition to double quotes (")
 - It is possible to explicitly denote empty elements

`<some_tag_name/>`

Markup Languages, 5



- XML DTDs can be omitted
 - A matching DTD can be always inferred, if the document is "well formed":
 - All tags must be nested
 - Tags must not overlap
 - Start tags must have matching end tags
- Structure is always independent of the formatting!
 - Cascading Style Sheets (CSS)
 - Allows to define formatting (layout) rules for elements
 - It is possible to define specific formatting (layout) rules for elements with attributes that have specific values or depending on the sequence of the elements



HyperText Markup Language (HTML)



- Text, marked up in HTML

```
<html>
  <head>
    <title>This is my HTML file</title>
  </head>
  <body>
    <h1>Important Heading</h1>
    <p>This <span class="verb">is</span> the
      first paragraph.
    <h1>Another Important Heading</h1>
    <p id="xyz1">Another paragraph.
    <p id="9876">This <span class="verb">is</span> it.
  </body>
</html>
```

Web Browser Output:

Important Heading

This is the first paragraph.

Another Important Heading

Another paragraph.

This is it.



Linking a Cascading Style Sheet (CSS)



- Text, marked up in HTML

```
<html>
  <head>
    <title>This is my HTML file</title>
    <link rel="stylesheet" type="text/css" href="example2.css">
  </head>
  <body>
    <h1>Important Heading</h1>
    <p>This <span class="verb">is</span> the
      first paragraph.
    <h1>Another Important Heading</h1>
    <p id="xyz1">Another paragraph.
    <p id="9876">This <span class="verb">is</span> it.
  </body>
</html>
```

Web Browser Output:

The image shows a yellow background representing a web browser's output. It contains the following elements:

- A blue heading: **Important Heading**
- A paragraph: This **is** the first paragraph.
- A second blue heading: **Another Important Heading**
- A paragraph: This **is** it.
- A small text label in the bottom right corner: ANOTHER PARAGRAPH

Cascading Style Sheets (CSS)



- example2.css

Tag	H1	<pre>{ color: blue; text-align: center; font-family: Arial,sans-serif; font-size: 200%; }</pre>
Tag	body	<pre>{ background-color: yellow; font-family: Times, Avantgarde; font-size: small; }</pre>
"class" Attribut	.verb	<pre>{ background-color: white; color: red; font-weight: 900; }</pre>
"id" Attribut	#xyz1	<pre>{ font-variant: small-caps; text-align: right; }</pre>
"id" Attribut	#9876	<pre>{ font-size: large; }</pre>



Linking HTML with a CSS, 2



```
<html>
  <head>
    <title>This is my HTML file</title>
    <link rel="stylesheet" type="text/css" href="example2.css">
  </head>
  <body>
    <h1>Important Heading</h1>
    <p>This <span class="verb">is</span> the
      first paragraph.
    <h1>Another Important Heading</h1>
    <p id="xyz1">Another paragraph.
    <p id="9876">This <span class="verb">is</span> it.
  </body>
</html>
```

```
H1      { color: blue;
          text-align: center;
          font-family: Arial,sans-serif;
          font-size: 200%; }
body    { background-color: yellow;
          font-family: Times, Avantgarde;
          font-size: small; }
.verb   { background-color: white;
          color: red;
          font-weight: 900; }
#xyz1   { font-variant: small-caps;
          text-align: right; }
#9876   { font-size: large; }
```

Web Browser Output:

Important Heading

This **is** the first paragraph.

Another Important Heading

Another paragraph.

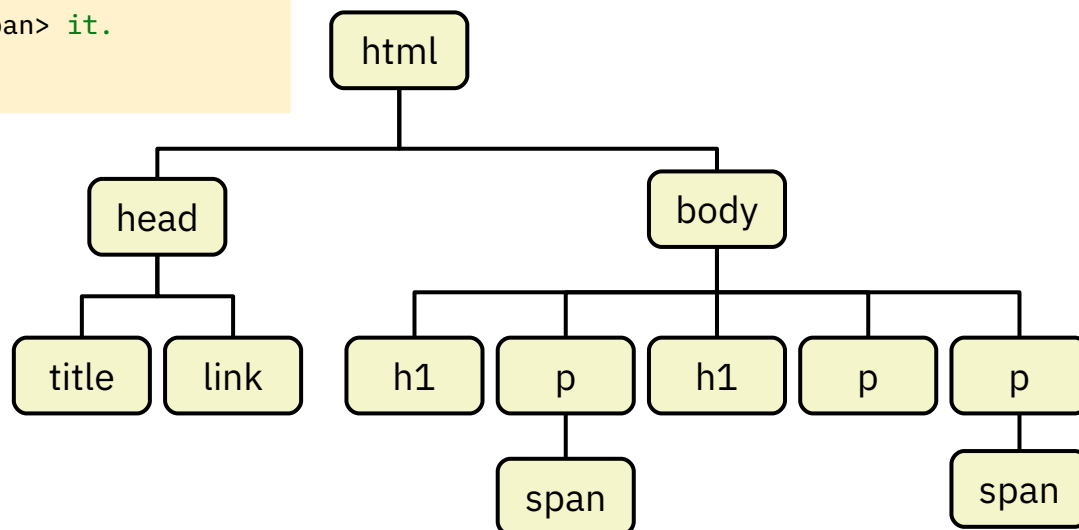
This **is** it.

ANOTHER PARAGRAPH

Document Object Model (DOM) – Parse Tree



```
<html>
  <head>
    <title>This is my HTML file</title>
    <link rel="stylesheet" type="text/css" href="example2.css">
  </head>
  <body>
    <h1>Important Heading</h1>
    <p>This <span class="verb">is</span> the
      first paragraph.
    <h1>Another Important Heading</h1>
    <p id="xyz1">Another paragraph.
    <p id="9876">This <span class="verb">is</span> it.
  </body>
</html>
```



Example: Extracting Link from HTML Text, 1



File Edit View History Bookmarks Tools Help

bach WU Applications

https://bach.wu.ac.at/z/start 90% Search

WU Start > bach > start

→ WU Applications

Home API

Ihr Alltag bekommt ein großes Update.

Alle Applikationen

WU

- WU Directory
- Online Vorlesungsverzeichnis
- Control Panel
- Raumreservierung Rooms
- Short URLs

Studierende

- LV- und Prüfungsanmeldung
- Online Datenvorerfassung
- Master Online Bewerbungsverfahren
- SB-Anerkennung (SBANK)
- Internetdienste für Studierende (ISTUD)

Mitarbeiter/innen Forschung

- Forschungsdokumentation (PURE)

bach.wu.ac.at/d/directory/

Display HTML text:
Ctrl-U

bach WU Directory

view-source:https://bach.wu.ac.at/z/start

view-source:https://bach.wu.ac.at/z/start

Gmail YouTube Maps Jitsi Meet

```
307
308 <div class="b3k_splitview_col2" id="b3k_appmenu">
309 <h1>Alle Applikationen</h1>
310 <h2>WU</h2>
311 <ul>
312 <li><a href="http://bach.wu.ac.at/d/directory/">WU Directory</a></li>
313 <li><a href="http://bach.wu.ac.at/start/vvz">Online Vorlesungsverzeichnis</a></li>
314 <li><a href="https://controlpanel.wu.ac.at/">Control Panel</a></li>
315 <li><a href="http://rooms.wu.ac.at/">Raumreservierung Rooms</a></li>
316 <li><a href="http://short.wu.ac.at/">Short URLs</a></li>
317 </ul>
```

Example: Extracting Link from HTML Text, 2 (MS Internet Explorer: Windows Only)



- Extracting WU's Directory URL (as of 2023-01-15)
 - Get WU BACH's home page from the URL: *https://bach.wu.ac.at/z/start*
 - Parse the received data for the WU directory URL and display it

```
ie=.OLEObject~new("InternetExplorer.Application")
ie~visible=.true           -- make IE visible
ie~navigate("https://bach.wu.ac.at/z/start")
do counter i until ie~busy=.false | i>1000 -- wait for page to be loaded (maximum one second)
    call sysSleep 0.001    -- sleep 1/1000 second
end
doc=ie~document           -- get loaded document
source=doc~body~innerHTML -- get body element and get its HTML text
ie~quit                   -- close Internet explorer
    -- parse the HTML text to extract a specific URL
parse var source '<h2>WU</h2>' . '<a href="' url '>WU Directory<'
say "Current URL of BACH-WU Directory:" url
```

Output:

```
Current URL of BACH-WU Directory: http://bach.wu.ac.at/d/directory/
```



Example: Extracting Link from HTML Text, 3 (Command "curl": All Operating Systems)



- Extracting WU's Directory URL (as of 2023-01-15)
 - Get WU BACH's home page from the URL: *https://bach.wu.ac.at/z/start*
 - Parse the received data for the WU directory URL and display it

```
command="curl --silent https://bach.wu.ac.at/z/start"
outArr=.array~new          -- array for stdout
ADDRESS SYSTEM command WITH OUTPUT USING (outArr)
source=outArr~makeString  -- turn array into string
parse var source '<h2>WU</h2>' . '<a href="' url '>WU Directory<'
say "Current URL of BACH-WU Directory:" url
```

Output:

```
Current URL of BACH-WU Directory: http://bach.wu.ac.at/d/directory/
```

- **HTML/XML Files**
 - Markup language
 - Tags defined in **SGML** or XML **DTD**
 - XML allows in addition **XSD** (XML Schema Definition)
 - **CSS** (cascading style sheets) for formatting elements
- **DOM** (document object model, w3.org)
 - Hierarchical (parse tree)
 - APIs for querying, inserting, changing and deleting of elements in the parse tree
 - Combine programming statements with events
 - **DHTML**
 - Microsoft's implementation of DOM used only in its Internet explorer (**IE**)
 - Incomplete and proprietary extensions

- World Wide Web Consortium (W3C)
 - <https://www.w3.org/> (2022-05-22)
 - <https://www.w3.org/Style/CSS/> (2022-05-22)
 - <https://dom.spec.whatwg.org/> (2022-05-22)
 - <https://www.w3.org/MarkUp/> (2022-05-22)
- SelfHTML
 - Excellent, easy to understand resources about HTML, CSS, XML, DOM, ...!
 - Unfortunately not in English, yet the examples are understandable
 - Tutorial
 - <https://www.selfhtml.org/> (2022-05-22)
- "curl"
 - <https://en.wikipedia.org/wiki/CURL> (2023-01-15)
 - <https://curl.se/> (2023-01-15)