



Distributed Computing

Lesson 9: WWW, HTML, and URLs

Thomas Weise · 汤卫思

twiese@hfu.edu.cn · <http://www.it-weise.de>

Hefei University, South Campus 2
Faculty of Computer Science and Technology
Institute of Applied Optimization
230601 Shushan District, Hefei, Anhui, China
Econ. & Tech. Devel. Zone, Jinxiu Dadao 99

合肥学院 南艳湖校区/南2区
计算机科学与技术系
应用优化研究所
中国 安徽省 合肥市 蜀山区 230601
经济技术开发区 锦绣大道99号

- 1 World Wide Web
- 2 Web Pages and Sites
- 3 URLs & Co.



website

- Understand the concepts of static HTML and URLs
- Understand closely related standards such as CSS and JavaScript

89-03 First proposal for HTML/WWW by Tim Berners-Lee at CERN in Geneva for a network of documents ^[1] in a Hypertext ^[2] format

- 89-03 First proposal for HTML/WWW by Tim Berners-Lee at CERN in Geneva for a network of documents ^[1] in a Hypertext ^[2] format
- 91-12 First text-based prototype shown at the *Hypertext'91* conference ^[3]

- 89-03 First proposal for HTML/WWW by Tim Berners-Lee at CERN in Geneva for a network of documents ^[1] in a Hypertext ^[2] format
- 91-12 First text-based prototype shown at the *Hypertext'91* conference ^[3]
- 92-12 50 WWW server worldwide

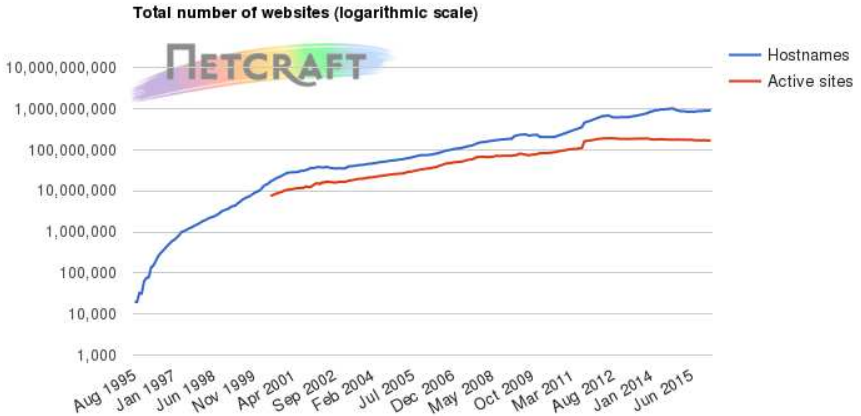
- 89-03 First proposal for HTML/WWW by Tim Berners-Lee at CERN in Geneva for a network of documents ^[1] in a Hypertext ^[2] format
- 91-12 First text-based prototype shown at the *Hypertext'91* conference ^[3]
- 92-12 50 WWW server worldwide
- 93-02 First graphical browser user interface (Mosaic) presented by Marc Andreessen ^[4]

- 89-03 First proposal for HTML/WWW by Tim Berners-Lee at CERN in Geneva for a network of documents ^[1] in a Hypertext ^[2] format
- 91-12 First text-based prototype shown at the *Hypertext'91* conference ^[3]
- 92-12 50 WWW server worldwide
- 93-02 First graphical browser user interface (Mosaic) presented by Marc Andreessen ^[4]
- 1994 Foundation of Netscape by M. Andreessen

- 89-03 First proposal for HTML/WWW by Tim Berners-Lee at CERN in Geneva for a network of documents ^[1] in a Hypertext ^[2] format
- 91-12 First text-based prototype shown at the *Hypertext'91* conference ^[3]
- 92-12 50 WWW server worldwide
- 93-02 First graphical browser user interface (Mosaic) presented by Marc Andreessen ^[4]
- 1994 Foundation of Netscape by M. Andreessen
- 1994 Foundation of the WWW Consortium (www.w3.org)

- 89-03 First proposal for HTML/WWW by Tim Berners-Lee at CERN in Geneva for a network of documents ^[1] in a Hypertext ^[2] format
- 91-12 First text-based prototype shown at the *Hypertext'91* conference ^[3]
- 92-12 50 WWW server worldwide
- 93-02 First graphical browser user interface (Mosaic) presented by Marc Andreessen ^[4]
- 1994 Foundation of Netscape by M. Andreessen
- 1994 Foundation of the WWW Consortium (www.w3.org)
- 1994 First international WWW conferenz in Geneva ^[5]

- 89-03 First proposal for HTML/WWW by Tim Berners-Lee at CERN in Geneva for a network of documents^[1] in a Hypertext^[2] format
- 91-12 First text-based prototype shown at the *Hypertext'91* conference^[3]
- 92-12 50 WWW server worldwide
- 93-02 First graphical browser user interface (Mosaic) presented by Marc Andreessen^[4]
- 1994 Foundation of Netscape by M. Andreessen
- 1994 Foundation of the WWW Consortium (www.w3.org)
- 1994 First international WWW conference in Geneva^[5]
- 1995 Netscape arrives at stock market: 1.5 billion USD



933 892 520 sites in Netcraft's February 2016 Web Server Survey ^[6]

① Resources

- ① **Resources:** a source of data: text, pictures, sounds, video, files. . .

- ① Resources: a source of data: text, pictures, sounds, video, files. . .
- ② HTML

- ① Resources: a source of data: text, pictures, sounds, video, files. . .
- ② **HTML**: A document format allowing for
 - organizing and linking other HTML documents together as a directed graph

- ① Resources: a source of data: text, pictures, sounds, video, files. . .
- ② **HTML**: A document format allowing for
 - organizing and linking other HTML documents together as a directed graph
 - linking to arbitrary resources

- ① Resources: a source of data: text, pictures, sounds, video, files. . .
- ② HTML: A document format allowing for
 - organizing and linking other HTML documents together as a directed graph
 - linking to arbitrary resources
- ③ URIs

- ① Resources: a source of data: text, pictures, sounds, video, files. . .
- ② HTML: A document format allowing for
 - organizing and linking other HTML documents together as a directed graph
 - linking to arbitrary resources
- ③ URIs: A way to locate and identify resources

- ① Resources: a source of data: text, pictures, sounds, video, files. . .
- ② HTML: A document format allowing for
 - organizing and linking other HTML documents together as a directed graph
 - linking to arbitrary resources
- ③ URIs: A way to locate and identify resources
- ④ HTTP

- ① Resources: a source of data: text, pictures, sounds, video, files. . .
- ② HTML: A document format allowing for
 - organizing and linking other HTML documents together as a directed graph
 - linking to arbitrary resources
- ③ URIs: A way to locate and identify resources
- ④ **HTTP**: A protocol allowing for requesting resources from a (remote) server

- A web server hosts web pages

- A web server hosts web pages
- A web browser accesses these pages via a protocol

- A web server hosts web pages
- A web browser accesses these pages via a protocol
- A web page can be defined in HTML enriched with JavaScript and Style Sheets

- A web server hosts web pages
- A web browser accesses these pages via a protocol
- A web page can be defined in HTML enriched with JavaScript and Style Sheets
- We now look at these server-side data formats

- HyperText Markup Language ^[7, 8]: Language to create web pages to be displayed in a web browser

- HyperText Markup Language ^[7, 8]: Language to create web pages to be displayed in a web browser
- Originally based on SGML (ISO Standard, complex, 1986 ^[9])

- HyperText Markup Language ^[7, 8]: Language to create web pages to be displayed in a web browser
- Originally based on SGML (ISO Standard, complex, 1986 ^[9])
- Hierarchical structure of

- HyperText Markup Language ^[7, 8]: Language to create web pages to be displayed in a web browser
- Originally based on SGML (ISO Standard, complex, 1986 ^[9])
- Hierarchical structure of:
 - tags enclosed by opening and closing brackets (`<html>`)

- HyperText Markup Language ^[7, 8]: Language to create web pages to be displayed in a web browser
- Originally based on SGML (ISO Standard, complex, 1986 ^[9])
- Hierarchical structure of:
 - tags enclosed by opening and closing brackets (`<html>`)
 - that usually are paired like `<h1>contents</h1>`

- HyperText Markup Language ^[7, 8]: Language to create web pages to be displayed in a web browser
- Originally based on SGML (ISO Standard, complex, 1986 ^[9])
- Hierarchical structure of:
 - tags enclosed by opening and closing brackets (`<html>`)
 - that usually are paired like `<h1>contents</h1>`
 - but there also are stand-alone tags without contents such as `` or `
`

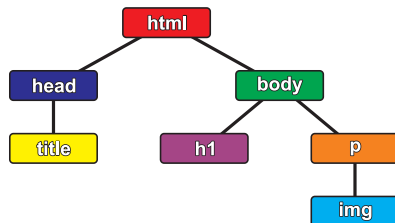
- HyperText Markup Language ^[7, 8]: Language to create web pages to be displayed in a web browser
- Originally based on SGML (ISO Standard, complex, 1986 ^[9])
- Hierarchical structure of:
 - tags enclosed by opening and closing brackets (`<html>`)
 - that usually are paired like `<h1>contents</h1>`
 - but there also are stand-alone tags without contents such as `` or `
`
 - some tags require attributes (e.g., ``)

- HyperText Markup Language ^[7, 8]: Language to create web pages to be displayed in a web browser
- Originally based on SGML (ISO Standard, complex, 1986 ^[9])
- Hierarchical structure of tags which give semantic information. . .
- . . .and layout information

- HyperText Markup Language ^[7, 8]: Language to create web pages to be displayed in a web browser
- Originally based on SGML (ISO Standard, complex, 1986 ^[9])
- Hierarchical structure of tags which give semantic information. . .
- . . . and layout information

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<html>  
  <head>  
    <title>Title of the website</title>  
  </head>  
  <body>  
    <h1>Headline</h1>  
    <p>The image   
      is nice.</p>  
  </body>  
</html>
```



Examples for HTML tags

Examples for HTML tags:

`<html>...</html>` tag pair enclosing the web page

Examples for HTML tags:

`<html>...</html>` tag pair enclosing the web page
`<head>...</head>` head of a web page: meta-information

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page
<code><body>...</body></code>	the body of a web page

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page
<code><body>...</body></code>	the body of a web page
<code><h1>...</h1></code>	head line of level 1

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page
<code><body>...</body></code>	the body of a web page
<code><h1>...</h1></code>	head line of level 1
<code><h2>...</h2></code>	head line of level 2

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page
<code><body>...</body></code>	the body of a web page
<code><h1>...</h1></code>	head line of level 1
<code><h2>...</h2></code>	head line of level 2
<code>...</code>	<code>...</code>

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page
<code><body>...</body></code>	the body of a web page
<code><h1>...</h1></code>	head line of level 1
<code><h2>...</h2></code>	head line of level 2
<code>...</code>	...
<code>...</code>	unordered list

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page
<code><body>...</body></code>	the body of a web page
<code><h1>...</h1></code>	head line of level 1
<code><h2>...</h2></code>	head line of level 2
...	...
<code>...</code>	unordered list
<code>...</code>	ordered list

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page
<code><body>...</body></code>	the body of a web page
<code><h1>...</h1></code>	head line of level 1
<code><h2>...</h2></code>	head line of level 2
...	...
<code>...</code>	unordered list
<code>...</code>	ordered list
<code>...</code>	list item

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page
<code><body>...</body></code>	the body of a web page
<code><h1>...</h1></code>	head line of level 1
<code><h2>...</h2></code>	head line of level 2
<code>...</code>	...
<code>...</code>	unordered list
<code>...</code>	ordered list
<code>...</code>	list item
<code><p>...</p></code>	text paragraph

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page
<code><body>...</body></code>	the body of a web page
<code><h1>...</h1></code>	head line of level 1
<code><h2>...</h2></code>	head line of level 2
<code>...</code>	<code>...</code>
<code>...</code>	unordered list
<code>...</code>	ordered list
<code>...</code>	list item
<code><p>...</p></code>	text paragraph
<code>
</code>	newline

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page
<code><body>...</body></code>	the body of a web page
<code><h1>...</h1></code>	head line of level 1
<code><h2>...</h2></code>	head line of level 2
<code>...</code>	...
<code>...</code>	unordered list
<code>...</code>	ordered list
<code>...</code>	list item
<code><p>...</p></code>	text paragraph
<code>
</code>	newline
<code>...</code>	hyperlink to another page

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page
<code><body>...</body></code>	the body of a web page
<code><h1>...</h1></code>	head line of level 1
<code><h2>...</h2></code>	head line of level 2
<code>...</code>	...
<code>...</code>	unordered list
<code>...</code>	ordered list
<code>...</code>	list item
<code><p>...</p></code>	text paragraph
<code>
</code>	newline
<code>...</code>	hyperlink to another page
<code></code>	embedded image

Examples for HTML tags:

<code><html>...</html></code>	tag pair enclosing the web page
<code><head>...</head></code>	head of a web page: meta-information
<code><title>...</title></code>	title of the web page
<code><body>...</body></code>	the body of a web page
<code><h1>...</h1></code>	head line of level 1
<code><h2>...</h2></code>	head line of level 2
<code>...</code>	...
<code>...</code>	unordered list
<code>...</code>	ordered list
<code>...</code>	list item
<code><p>...</p></code>	text paragraph
<code>
</code>	newline
<code>...</code>	hyperlink to another page
<code></code>	embedded image
<code>...</code>	...

```
<html>
  <head>
    <title>Main Page</title>
  </head>

  <body>
    <h1>Main Page</h1>
    <h2>Ordered List</h2>
    <ol>
      <li>item 1</li>
      <li>item 2</li>
      <li>item 3</li>
      <li>item 4</li>
    </ol>
  </body>
</html>
```

HTML

Main Page

Ordered List

1. item 1
2. item 2
3. item 3
4. item 3

**as shown
in Browser**

```
<html>
  <head>
    <title>Main Page</title>
  </head>

  <body>
    <h1>Main Page</h1>
    <h2>Unordered List</h2>
    <ul>
      <li>item 1</li>
      <li>item 2</li>
      <li>item 3</li>
      <li>item 4</li>
    </ul>
  </body>
</html>
```

HTML

Main Page

Unordered List

- item 1
- item 2
- item 3
- item 3

**as shown
in Browser**

```
<html>
<head>
  <title>Main Page</title>
</head>

<body>
  <h1>Main Page</h1>

  <h2>Several Paragraphs of Text</h2>

  <p>The Nature Publishing Group released the Nature Publishing
Index (NPI) Asia-Pacific and Nature Publishing Index China in
January, 2013 and the University of Science and Technology of
China (USTC) continues to take the lead among Chinese universities
in the rankings.</p>

  <p>University of Science and Technology of China was ranked no.3
in Chinese Mainland along with Fudan University and 220-225th in
the world in the 2012-13 Times Higher Education (THE) World
University Rankings, announced Oct 3, 2012. Peking University was
ranked 46th and Tsinghua University 52nd.</p>

</body>
</html>
```

HTML

Main Page

Several Paragraphs of Text

The Nature Publishing Group released the Nature Publishing Index (NPI) Asia-Pacific and Nature Publishing Index China in January, 2013 and the University of Science and Technology of China (USTC) continues to take the lead among Chinese universities in the rankings.

University of Science and Technology of China was ranked no.3 in Chinese Mainland along with Fudan University and 220-225th in the world in the 2012-13 Times Higher Education (THE) World University Rankings, announced Oct 3, 2012. Peking University was ranked 46th and Tsinghua University 52nd.

**as shown
in Browser**

```
<html>
<head>
  <title>Main Page</title>
</head>

<body>
  <h1>Main Page</h1>

  <h2>Image</h2>

</body>
</html>
```

HTML

Main Page

Image



**as shown
in Browser**

```
<html>
<head>
  <title>Main Page</title>
</head>

<body>
  <h1>Main Page</h1>

  <h2>Table</h2>
  <table border="1">
    <tr>
      <th>A</th>    <th>B</th>    <th>C</th>    <th>D</th>
    </tr>

    <tr>
      <td>E</td>    <td rowspan="2">F</td>    <td>G</td>    <td>H</td>
    </tr>

    <tr>
      <td>I</td>    <td colspan="2">J</td>
    </tr>

    <tr>
      <td colspan="3" rowspan="2">K</td>    <td>L</td>
    </tr>

    <tr>
      <td>M</td>
    </tr>

    <tr>
      <td>N</td>    <td>O</td>    <td>P</td>    <td>Q</td>
    </tr>
  </table>

</body>
</html>
```

HTML

Main Page Table

A	B	C	D	
E	F	G	H	
I		J		
K			L	
			M	
N	O	P	Q	

as shown
in Browser

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Main Page</title>
</head>
<body>
<h1>Main Page</h1>
<h2>Ordered List</h2>
<ol>
<li>item 1</li>
<li>item 2</li>
<li>item 3</li>
<li>item 3</li>
</ol>
<h2>Unordered List</h2>
<ul>
<li>item 1</li>
<li>item 2</li>
<li>item 3</li>
<li>item 3</li>
</ul>
<h2>Several Paragraphs of Text</h2>
<p>The Nature Publishing Group released the Nature Publishing Index (NPI) Asia-Pacific and Nature Publishing Index China in January, 2013 and the University of Science and Technology of China (USTC) continues to take the lead among Chinese universities in the rankings </p>
<p>University of Science and Technology of China was ranked no.3 in Chinese Mainland along with Fudan University and 220-225th in the world in the 2012-13 Times Higher Education (THE) World University Rankings, announced Oct 3, 2012. Peking University was ranked 46th and Tsinghua University 52nd </p>

<table border="1">
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
<tr>
<td>B</td>
<td colspan="2">F</td>
<td>D</td>
<td>B</td>
</tr>
<tr>
<td>I</td>
<td colspan="2">J</td>
<td></td>
<td></td>
</tr>
<tr>
<td colspan="3">K</td>
<td colspan="2">L</td>
</tr>
<tr>
<td colspan="5">M</td>
</tr>
<tr>
<td>N</td>
<td>O</td>
<td>P</td>
<td colspan="2">Q</td>
</tr>
</table>
</body>
</html>
```

HTML

Main Page

Ordered List

- item 1
- item 2
- item 3
- item 3

Unordered List

- item 1
- item 2
- item 3
- item 3

Several Paragraphs of Text

The Nature Publishing Group released the Nature Publishing Index (NPI) Asia-Pacific and Nature Publishing Index China in January, 2013 and the University of Science and Technology of China (USTC) continues to take the lead among Chinese universities in the rankings.

University of Science and Technology of China was ranked no.3 in Chinese Mainland along with Fudan University and 220-225th in the world in the 2012-13 Times Higher Education (THE) World University Rankings, announced Oct 3, 2012. Peking University was ranked 46th and Tsinghua University 52nd.

Image



Table

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

as shown in Browser

- HTML itself is static

- HTML itself is static:
 - You can click links to get to other pages

- HTML itself is static:
 - You can click links to get to other pages
 - fill out and submit forms

- HTML itself is static:
 - You can click links to get to other pages
 - fill out and submit forms
 - but there is no other dynamic action going on

- HTML itself is static:
 - You can click links to get to other pages
 - fill out and submit forms
 - but there is no other dynamic action going on
 - HTML is **NOT** a programming language!

- HTML itself is static
- JavaScript ^[10]: Script-based programming language licensed by Netscape

- HTML itself is static
- JavaScript ^[10]: Script-based programming language licensed by Netscape
- Enables performing of activities on the client side (browser)

- HTML itself is static
- JavaScript ^[10]: Script-based programming language licensed by Netscape
- Enables performing of activities on the client side (browser)
- Not really similar to Java, name choice for marketing reasons

- HTML itself is static
- JavaScript ^[10]: Script-based programming language licensed by Netscape
- Enables performing of activities on the client side (browser)
- Not really similar to Java, name choice for marketing reasons
- Goal: “Dynamic HTML” (DHTML) ^[11]

- HTML itself is static
- JavaScript ^[10]: Script-based programming language licensed by Netscape
- Enables performing of activities on the client side (browser)
- Not really similar to Java, name choice for marketing reasons
- Goal: “Dynamic HTML” (DHTML) ^[11]:
 - web page can change while it is viewed in the browser

- HTML itself is static
- JavaScript ^[10]: Script-based programming language licensed by Netscape
- Enables performing of activities on the client side (browser)
- Not really similar to Java, name choice for marketing reasons
- Goal: “Dynamic HTML” (DHTML) ^[11]:
 - web page can change while it is viewed in the browser
 - these changes depend on events or user interaction

- HTML itself is static
- JavaScript ^[10]: Script-based programming language licensed by Netscape
- Enables performing of activities on the client side (browser)
- Not really similar to Java, name choice for marketing reasons
- Goal: “Dynamic HTML” (DHTML) ^[11]
- AJAX (Asynchronous JavaScript and XML) ^[12]

- HTML itself is static
- JavaScript ^[10]: Script-based programming language licensed by Netscape
- Enables performing of activities on the client side (browser)
- Not really similar to Java, name choice for marketing reasons
- Goal: “Dynamic HTML” (DHTML) ^[11]
- AJAX (Asynchronous JavaScript and XML) ^[12]:
 - web page in browser keeps communicating with web/application server

- HTML itself is static
- JavaScript ^[10]: Script-based programming language licensed by Netscape
- Enables performing of activities on the client side (browser)
- Not really similar to Java, name choice for marketing reasons
- Goal: “Dynamic HTML” (DHTML) ^[11]
- AJAX (Asynchronous JavaScript and XML) ^[12]:
 - web page in browser keeps communicating with web/application server
 - user interacts with web page, web page sends request to server, gets answer, is updated
 - user gets impression that an application is running in his browser

- HTML itself is static
- JavaScript ^[10]: Script-based programming language licensed by Netscape
- Enables performing of activities on the client side (browser)
- Not really similar to Java, name choice for marketing reasons
- Goal: “Dynamic HTML” (DHTML) ^[11]
- AJAX (Asynchronous JavaScript and XML) ^[12]:
 - web page in browser keeps communicating with web/application server
 - user interacts with web page, web page sends request to server, gets answer, is updated
 - user gets impression that an application is running in his browser

```
<html>
<head>
  <title>Example 1 for JavaScript</title>
</head>
<body>
  <script type="text/javascript">
    var now      = new Date();
    var day      = now.getDate();
    var month    = now.getMonth() + 1;
    var year     = now.getFullYear();
    var hours    = now.getHours();
    var minutes  = now.getMinutes();

    if (year < 2000) year = year + 1900;

    document.write("<h2>Hi!</h2><b>Today is the " +
      year + "-"");

    if (month < 10) document.write("0");

    document.write(month + "-"");

    if (day < 10) document.write("0");

    document.write(day + ", It is now " + hours + ":"");

    if (minutes < 10) document.write("0");

    document.write(minutes + " o'clock</b>");
  </script>
</body>
</html>
```

HTML

Hi!

Today is the 2013-03-27. It is now 15:29 o'clock

**as shown
in Browser**


```
<html>
<head>
  <title>Example 2 for JavaScript</title>
</head>
<body>
  <h1>Calculator</h1>
  <form name="Calculator" action="">
    <table border="0">
      <tr><td>Formula:</td>
      <td><input type="text" size="20"
        name="Formula" /></td></tr>

      <tr><td>Result:</td>
      <td><input type="text" size="20"
        name="Result" disabled="disabled"/>
      </td></tr>

      <tr><td colspan="2">
        <input type="button" value="Calculate!"
          onclick="document.Calculator.Result.value =
            eval(document.Calculator.Formula.value)" />
      </td></tr></table>
    </form>
  </body>
</html>
```

HTML

Calculator

Formula:

Result:

as shown
in Browser

- Cascading Style Sheets (CSS) ^[13]: Standard complementing HTML

- Cascading Style Sheets (CSS) ^[13]: Standard complementing HTML
- HTML elements define structure (e.g., table) and semantics (e.g., paragraph, emphasize)

- Cascading Style Sheets (CSS) ^[13]: Standard complementing HTML
- HTML elements define structure (e.g., table) and semantics (e.g., paragraph, emphasize)
- CSS defines the presentation for HTML elements

- Cascading Style Sheets (CSS) ^[13]: Standard complementing HTML
- HTML elements define structure (e.g., table) and semantics (e.g., paragraph, emphasize)
- CSS defines the presentation for HTML elements:
 - margins, fonts, sizes, colors, bold/italic faces, ...

- Cascading Style Sheets (CSS) ^[13]: Standard complementing HTML
- HTML elements define structure (e.g., table) and semantics (e.g., paragraph, emphasize)
- CSS defines the presentation for HTML elements:
 - margins, fonts, sizes, colors, bold/italic faces, ...
- Different versions, sometimes incorrectly implemented

- Cascading Style Sheets (CSS) ^[13]: Standard complementing HTML
- HTML elements define structure (e.g., table) and semantics (e.g., paragraph, emphasize)
- CSS defines the presentation for HTML elements:
 - margins, fonts, sizes, colors, bold/italic faces, ...
- Different versions, sometimes incorrectly implemented
- Goals

- Cascading Style Sheets (CSS) ^[13]: Standard complementing HTML
- HTML elements define structure (e.g., table) and semantics (e.g., paragraph, emphasize)
- CSS defines the presentation for HTML elements:
 - margins, fonts, sizes, colors, bold/italic faces, ...
- Different versions, sometimes incorrectly implemented
- Goals:
 - completely separate presentation from contents

- Cascading Style Sheets (CSS) ^[13]: Standard complementing HTML
- HTML elements define structure (e.g., table) and semantics (e.g., paragraph, emphasize)
- CSS defines the presentation for HTML elements:
 - margins, fonts, sizes, colors, bold/italic faces, ...
- Different versions, sometimes incorrectly implemented
- Goals:
 - completely separate presentation from contents
 - different formatting for different output media: visual, language, Braille, ...

```
<html>
<head>
  <title>CSS Example</title>
  <style type="text/css">

    h1 { color:red;
         font-size:48px;}
    body {
      font-family:'Comic Sans MS',Comic,sans-serif}
  </style>
</head>
<body>
  <h1>Main Page</h1>

  <p>This is a text.</p>

  <p>This is another one.</p>
</body>
</html>
```

HTML

Main Page

This is a text.

This is another one.

**as shown
in Browser**

- OK, we can now define a web page

- OK, we can now define a web page and put it on the web server.

- OK, we can now define a web page and put it on the web server. But how does a browser *find* the web page?

- How does a browser *find* the web page?

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI)

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - `scheme` states how to get a resource

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - `scheme` states how to get a resource (could identify a protocol)

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - scheme states how to get a resource
 - the rest is the name of the resource

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - scheme states how to get a resource
 - the rest is the name of the resource
 - Examples

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - scheme states how to get a resource
 - the rest is the name of the resource
 - Examples:
 - `http://en.wikipedia.org/wiki/Uniform_resource_identifier#Syntax`

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - scheme states how to get a resource
 - the rest is the name of the resource
 - Examples:
 - `http://en.wikipedia.org/wiki/Uniform_resource_identifier#Syntax`
 - `urn:isbn:0-486-27557-4`

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - scheme states how to get a resource
 - the rest is the name of the resource
 - Examples:
 - `http://en.wikipedia.org/wiki/Uniform_resource_identifier#Syntax`
 - `urn:isbn:0-486-27557-4`
 - Dynamic scenarios: `http:host/path/object?param1=val1¶m2=val2`

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - scheme states how to get a resource
 - the rest is the name of the resource
 - Examples:
 - `http://en.wikipedia.org/wiki/Uniform_resource_identifier#Syntax`
 - `urn:isbn:0-486-27557-4`
 - Dynamic scenarios: `http:host/path/object?param1=val1¶m2=val2`
- Uniform Resource Locator (URL)

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - scheme states how to get a resource
 - the rest is the name of the resource
 - Examples:
 - `http://en.wikipedia.org/wiki/Uniform_resource_identifier#Syntax`
 - `urn:isbn:0-486-27557-4`
 - Dynamic scenarios: `http:host/path/object?param1=val1¶m2=val2`
- Uniform Resource Locator (URL):
 - URI based on location rather than any other attributes

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - scheme states how to get a resource
 - the rest is the name of the resource
 - Examples:
 - `http://en.wikipedia.org/wiki/Uniform_resource_identifier#Syntax`
 - `urn:isbn:0-486-27557-4`
 - Dynamic scenarios: `http:~:host/path/object?param1=val1¶m2=val2`
- Uniform Resource Locator (URL):
 - URI based on location rather than any other attributes

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - scheme states how to get a resource
 - the rest is the name of the resource
 - Examples:
 - `http://en.wikipedia.org/wiki/Uniform_resource_identifier#Syntax`
 - `urn:isbn:0-486-27557-4`
 - Dynamic scenarios: `http:host/path/object?param1=val1¶m2=val2`
- Uniform Resource Locator (URL):
 - URI based on location rather than any other attributes
 - the `host` part of an URL can be mapped to an IP-address via DNS

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - scheme states how to get a resource
 - the rest is the name of the resource
 - Examples:
 - `http://en.wikipedia.org/wiki/Uniform_resource_identifier#Syntax`
 - `urn:isbn:0-486-27557-4`
 - Dynamic scenarios: `http:host/path/object?param1=val1¶m2=val2`
- Uniform Resource Locator (URL):
 - URI based on location rather than any other attributes
 - the `host` part of an URL can be mapped to an IP-address via DNS
 - The `path/object#component` part is a relative identifier that the server running on `host` should understand

- How does a browser *find* the web page?
- Uniform Resource Identifier (URI):
 - a string of characters used to identify a name or a web resource
 - most common structure: `scheme:host/path/object#component`
 - scheme states how to get a resource
 - the rest is the name of the resource
 - Examples:
 - `http://en.wikipedia.org/wiki/Uniform_resource_identifier#Syntax`
 - `urn:isbn:0-486-27557-4`
 - Dynamic scenarios: `http:host/path/object?param1=val1¶m2=val2`
- Uniform Resource Locator (URL):
 - URI based on location rather than any other attributes
 - the `host` part of an URL can be mapped to an IP-address via DNS
 - `path/object#component` is an identifier that the host should understand (in the simple case, `path/object` is a path to a file relative to a specified root folder on host)

- URI vs. URL

- URI vs. URL:
 - URIs are not necessarily location-based ($\text{URLs} \subset \text{URIs}$)

- URI vs. URL:
 - URIs are not necessarily location-based ($\text{URLs} \subset \text{URIs}$)
 - Most URIs based on location in practice and use often interchangeable

- URI vs. URL:
 - URIs are not necessarily location-based ($\text{URLs} \subset \text{URIs}$)
 - Most URIs based on location in practice and use often interchangeable
 - Exception: XML namespaces (see a later lesson)

- URI vs. URL:
 - URIs are not necessarily location-based ($\text{URLs} \subset \text{URIs}$)
 - Most URIs based on location in practice and use often interchangeable
 - Exception: XML namespaces (see a later lesson)
 - Other exception URNs for identifying documents (my dissertation thesis has `urn:nbn:de:hebis:34-2009051127217`)

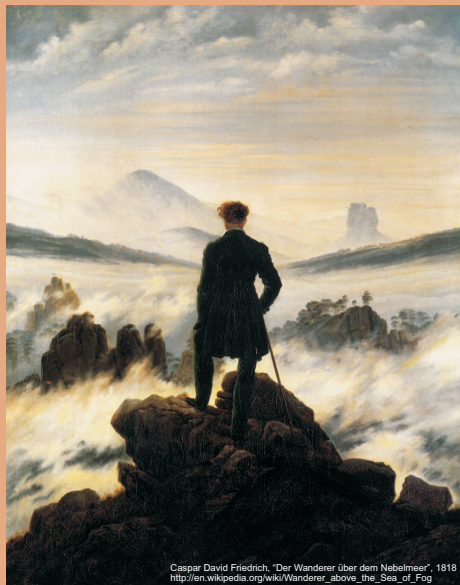
- HTML allows us to create structured text documents with graphics, tables, lists and other components.
- CSS allows us to style these documents.
- JavaScript allows us to include code which will run in the web browser.
- URLs allow us to locate resources in the WWW and we can use them together with hyperlinks to link different documents and resources together.

谢谢

Thank you

Thomas Weise [汤卫思]
tweise@hfu.edu.cn
<http://www.it-weise.de>

Hefei University, South Campus 2
Institute of Applied Optimization
Shushan District, Hefei, Anhui,
China



Caspar David Friedrich, "Der Wanderer über dem Nebelmeer", 1818
http://en.wikipedia.org/wiki/Wanderer_above_the_Sea_of_Fog



1. Timothy John Berners-Lee. Information management: A proposal. Technical report, Geneva, Switzerland: European Organization for Nuclear Research (Organisation Européenne pour la Recherche Nucléaire, CERN), 1989. URL <http://www.w3.org/History/1989/proposal.html>.
2. John B. Smith and S. F. Weiss. Hypertext. *Communications of the ACM (CACM)*, 31(7):816–819, 1988. doi: 10.1145/48511.48512.
3. *Proceedings of ACM Hypertext 91 Conference (Hypertext'91)*, San Antonio, TX, USA, December 15–18, 1991. New York, NY, USA: ACM Press. ISBN 0897914619.
4. Marc Andreessen. Ncsa mosaic technical summary. Technical report, Urbana-Champaign, IL, USA: University of Illinois at Urbana-Champaign, National Center for Supercomputing Applications (NCSA), Software Development Group, February 20–May 8, 1993. URL www.livinginternet.com/w/references/mosaic.2.1.zip.
5. Mario Ruggier, editor. *First International Conference on the World-Wide Web (WWW'94)*, Geneva, Switzerland: European Organization for Nuclear Research (Organisation Européenne pour la Recherche Nucléaire, CERN), May 25–27, 1994. URL <http://www94.web.cern.ch/www94/PrelimProcs.html>.
6. UK: Netcraft Ltd. Bath. March 2013 web server survey, March 2013. URL <http://news.netcraft.com/archives/2013/03/01/march-2013-web-server-survey.html>.
7. Dave Raggett, Arnaud Le Hors, and Ciel J. H. Jacobs. *HTML 4.01 Specification*. W3C Recommendation. MIT/CSAIL (USA), ERCIM (France), Keio University (Japan): World Wide Web Consortium (W3C), December 24, 1999. URL <http://www.w3.org/TR/1999/REC-html401-19991224>.
8. Murray Altheim and Shane McCarron. *XHTML™ 1.1 – Module-based XHTML – Second Edition*. W3C Recommendation. MIT/CSAIL (USA), ERCIM (France), Keio University (Japan): World Wide Web Consortium (W3C), November 23, 2010. URL <http://www.w3.org/TR/2010/REC-xhtml11-20101123>.
9. *ISO 8879:1986 Information processing – Text and office systems – Standard Generalized Markup Language (SGML)*. Geneva, Switzerland: International Organization for Standardization (ISO), October 1, 1986.
10. *ISO/IEC 16262:2011(E): Information Technology – Programming Languages, their Environments and System Software Interfaces – ECMAScript Language Specification*. Geneva, Switzerland: International Organization for Standardization (ISO), June 15, 2011. URL [http://standards.iso.org/ittf/PubliclyAvailableStandards/c055755_ISO_IEC_16262_2011\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c055755_ISO_IEC_16262_2011(E).zip).
11. Patrick Carey. *New Perspectives on HTML, XHTML, and Dynamic HTML: Comprehensive*. New Perspective Series. Stamford, CT, USA: Cengage Learning (Wadsworth Publishing Company), 4th, illustrated edition, 2009. ISBN 1-4239-2543-2 and 978-1-4239-2543-9. URL <http://books.google.de/books?id=BeEzti-oB0IC>.

12. Jesse James Garrett. Ajax: A new approach to web applications, February 18, 2005. URL <http://web.archive.org/web/20080702075113/http://www.adaptivepath.com/ideas/essays/archives/000385.php>.
13. Bert Bos, Tantek Çelik, Ian Hickson, and Håkon Wium Lie. *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*. W3C Recommendation. MIT/CSAIL (USA), ERCIM (France), Keio University (Japan): World Wide Web Consortium (W3C), June 7, 2011. URL <http://www.w3.org/TR/2011/REC-CSS2-20110607>.