Vienna University of Technology

# Web Engineering

188.951 2VU SS15

## M2: (X)HTML, HTML5, CSS, CSS3, and JavaScript

**Business Informatics Group**

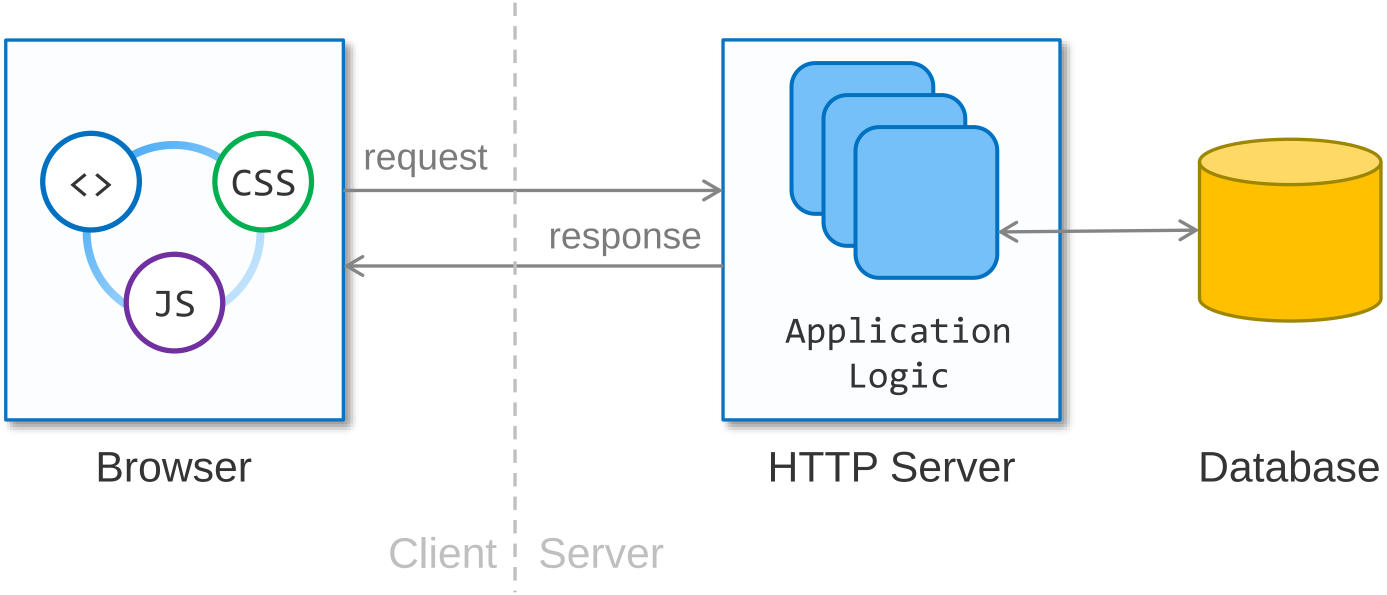
Institute of Software Technology and Interactive Systems
Vienna University of Technology

Favoritenstraße 9-11/188-3, 1040 Vienna, Austria
phone: +43 (1) 58801-18804 (secretary), fax: +43 (1) 58801-18896
office@big.tuwien.ac.at, www.big.tuwien.ac.at

# Basic Technologies

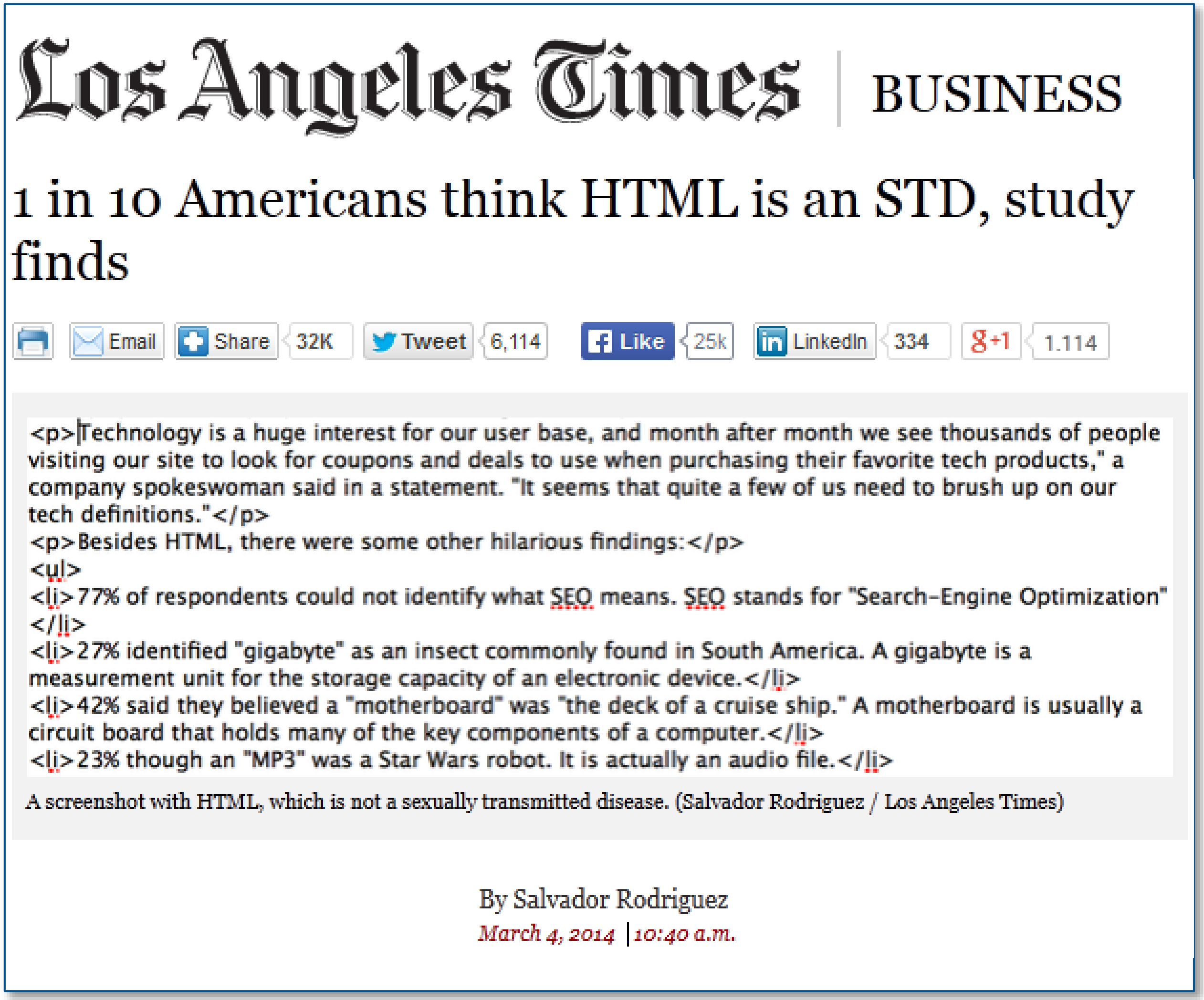
## Overview

- Client Side
  - (X)HTML for structure and content
  - CSS for layout and style
  - JavaScript for client-side functionality



# HTML

## What is HTML?



Los Angeles Times | BUSINESS

### 1 in 10 Americans think HTML is an STD, study finds

```
<p>Technology is a huge interest for our user base, and month after month we see thousands of people
visiting our site to look for coupons and deals to use when purchasing their favorite tech products," a
company spokeswoman said in a statement. "It seems that quite a few of us need to brush up on our
tech definitions."</p>
<p>Besides HTML, there were some other hilarious findings:</p>
<ul>
<li>77% of respondents could not identify what SEO means. SEO stands for "Search-Engine Optimization"
</li>
<li>27% identified "gigabyte" as an insect commonly found in South America. A gigabyte is a
measurement unit for the storage capacity of an electronic device.</li>
<li>42% said they believed a "motherboard" was "the deck of a cruise ship." A motherboard is usually a
circuit board that holds many of the key components of a computer.</li>
<li>23% though an "MP3" was a Star Wars robot. It is actually an audio file.</li>
```

A screenshot with HTML, which is not a sexually transmitted disease. (Salvador Rodriguez / Los Angeles Times)

By Salvador Rodriguez

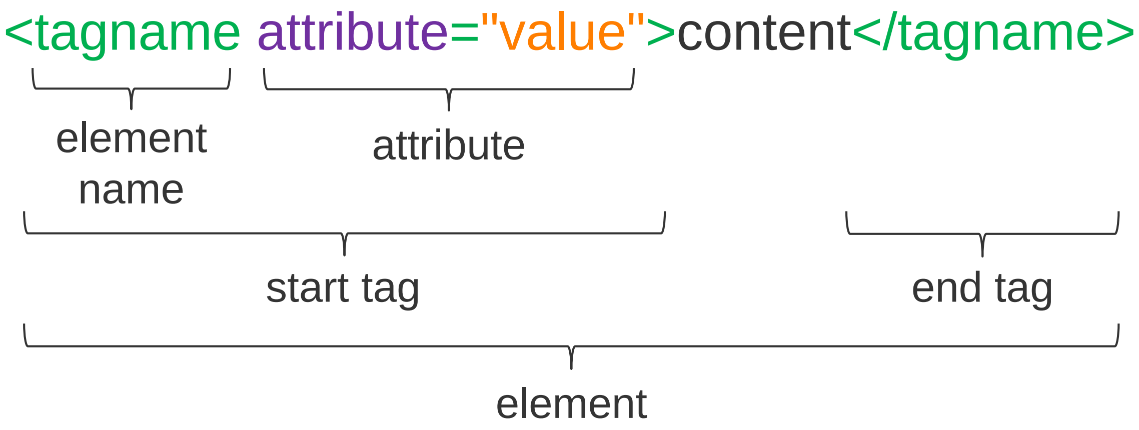
*March 4, 2014* | *10:40 a.m.*

http://www.latimes.com/business/technology/la-fi-tn-1-10-americans-html-std-study-finds-20140304,0,1188415.story#axzz2v6nXpHOy

# HTML

What is HTML?

- Hyper Text Markup Language
- Standardized by the W3C
- Describes structure and content of a document
- Human and non-human users
  - Web Browser parses the content and presents it to the end user
  - Crawler indexes the parsed content (machine-readability)

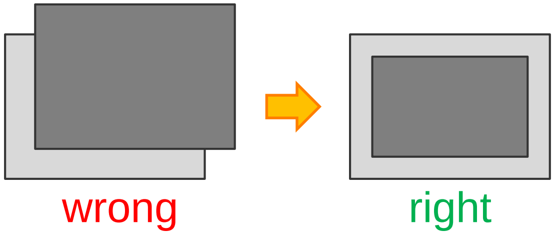


<tagname attribute="value">content</tagname>

# XHTML

## What is XHTML?

- Extensible HTML
- Uses XML markup language for HTML → Well-formedness
  - Single root element for docments
  - Elements must be closed
  - Proper nesting
  - …



```
<b><i>bold and italic</b></i><br>
```

→

```
<b><i>bold and italic</i></b><br/>
```

- Stricter Rules
  - Mandatory elements and attributes
    - XHTML DOCTYPE, namespace, <html>, <head>, <title>, <body>
  - Attributes
    - Names must be lower case, values must be quoted, minimization forbidden

# From HTML to XHTML (1/3)

## History

Erste Webseite
http://info.cern.ch

- 1989 Introduced by Tim Berners-Lee
- 1992 Early versions had too little features (formatting, …)
  - Browser vendors introduced new, incompatible features
    - → Browser war
- 1994 WC3 is founded
- 1997 HTML 4.0 introduced stylesheets, scripts, frames, …
  - Separation of doctypes (*strict*, *transitional*, *frameset*)
- 1999 HTML 4.01: replacing HTML 4.0 based on SGML
- 2000 XHTML 1.0: XML-based HTML 4.01 (including doctypes)
- 2001 XHTML 1.1: skipped deprecated elements, attributes
  - Only doctype *strict, modularization (later)*

# From HTML to XHTML (2/3)

History

2005

- Only 3.9 % of German web sites are valid (X)HTML
  - i.e. 3.9 % follow the W3C Recommendations
  - Source: ValiWatch 2005 [1)]

2007
ongoing

- HTML 5 (working draft)
  - focuses on the language vocabulary that is adequate for web documents, as well as for *web applications*
- XHTML 2.0 (working draft)
  - Discontinued by the end of 2009
  - → XHTML 5 (an XML-based serialization of HTML 5)

[1)] http://www.validome.org/lang/ge/html/valiwatch-web-2005

# From HTML to XHTML (2/3)

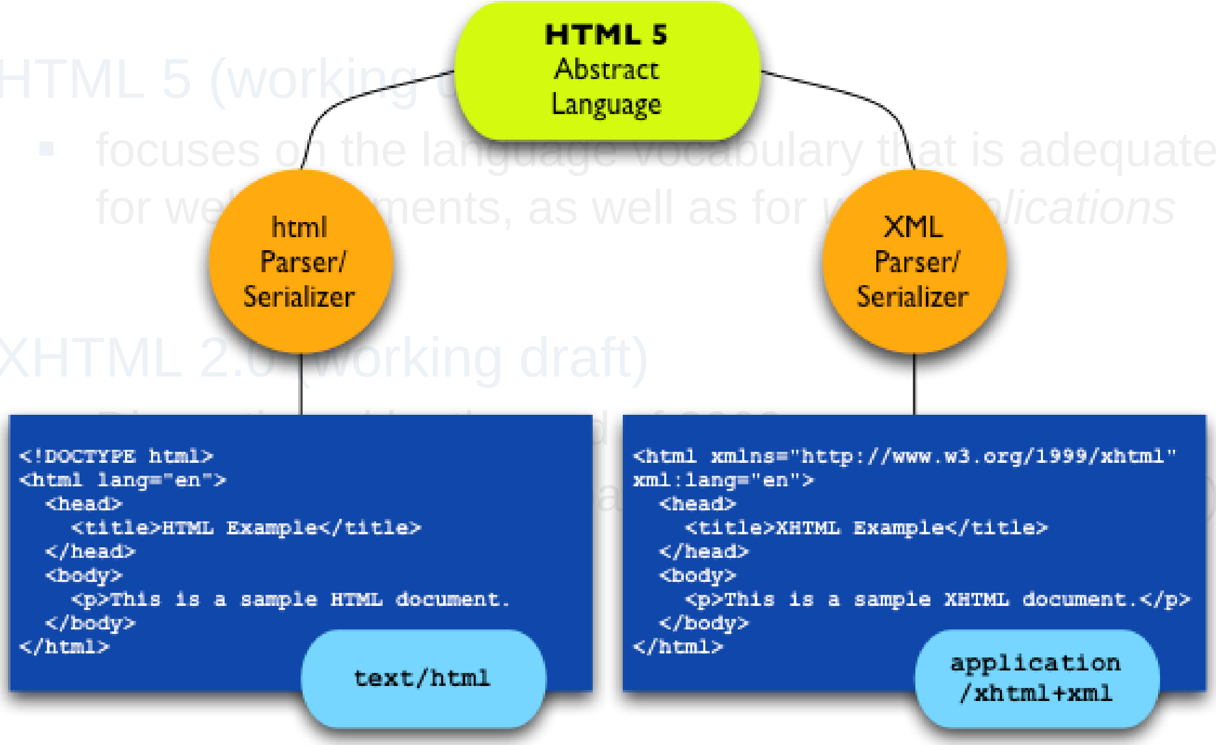
History

2005

- Only 3.9 % of German web sites are valid (X)HTML
  - i.e. 3.9 % follow the W3C Recommendations
  - Source: ValiWatch 2005 1)

2007 ongoing

- HTML 5 (working draft)
  - focuses on the language vocabulary that is adequate for web documents, as well as for web *applications*
- XHTML 2.0 (working draft)



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>HTML Example</title>
  </head>
  <body>
    <p>This is a sample HTML document.
  </body>
</html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en">
  <head>
    <title>XHTML Example</title>
  </head>
  <body>
    <p>This is a sample XHTML document.</p>
  </body>
</html>
```

http://www.w3.org/QA/2008/01/html5-is-html-and-xml.html

# XHTML

Typical page structure

- Structure

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
   http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta name="author" content="WE"/>
    <title xml:lang="en">Title</title>
  </head>
  <body>
    <h1>First order header</h1>
    <p>Paragraph content</p>
  </body>
</html>
```

XML declaration

Document type

Document element

Head with meta data

Body with content

- Note
  - namespace
  - xml:lang possible for any element

# XHTML

## Document Type

- XML declaration
  - Necessary for XHTML
  - Version of XML being used

```
<?xml version="1.0" encoding="UTF-8"?>
```

- Document Type
  - Dinstinguish Versions
  - Quirks mode
  - Check Validity
    - http://validator.w3.org

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
   http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

- Document element
  - Single root element

```
<html xmlns="http://www.w3.org/1999/xhtml">
...
</html>
```

# XHTML

## Document Type

- Head with Meta data
  - Title
  - Data from meta element
    - Author, Keywords, Date, …
  - Linking to other resources
    - CSS, JavaScript, …
  - Meta data with Dublin Core (DC)
    - Link to DC schema and terms
    - Use in meta element

```
<head>
    <meta name="author" content="WE"/>
    <title xml:lang="en">Title</title>
</head>
```

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

```
<link rel="schema.DC"
      href="http://purl.org/dc/elements/1.1/">
<meta name="DC.creator" content="WE">
```

- Body with content
  - Whatever you want ;-)

```
<body>
    <h1>First order header</h1>
    <p>Paragraph content</p>
</body>
```

# XHTML

## Modularization

- Categorization of elements into modules

| Module | Elements |
|---|---|
| Structure | body, head, html, title |
| Text | Heading (h1-h6), Blocks (address, div, p,…), Inline (br, span, em, …) |
| Hypertext | a |
| List | dl, dt, dd, ol, ul, li |
| Text extension | b, I, hr, sub, sup, … |
| Form | Form (form, fieldset), Formctrl (input, label, select, textarea, button), … |
| Table | table, caption, th, tr, td, col, colgroup, tbody, thead, tfoot |
| Image | img |
| … | … |

- Why?
  - Reduce complexity for mobile phones, etc.
  - Possibility to extend XHTML with other markup languages (Compound documents)
    - XHTML+SVG, XHTML+MathML

http://www.w3.org/TR/xhtml-modularization/

# (X)HTML

## Syntax vs Semantic

- Syntax

```
<tagname attribute="value">content</tagname>
```

- Semantic
  - Depends on the element
    - `<h1>` is a first order header != the thickest printed text
    - `<b>` prints text bold != `<em>` emphasizes the text
    - `<table>` represents tabular data != layout mechanism

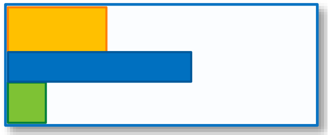
- Why use syntactically and semantically correct elements?
  - Browser compatibility, accessibility (later)
  - Easier processing for tools, e.g., XML tools, indexing for search engines
  - Improved usage of your content, e.g., readability
  - More efficient browsing (no interpretation of wrong XHTML necessary)

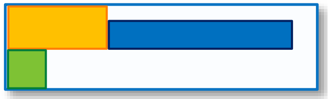
→ Web gets more usable and accessibly

# XHTML 1.1

## Links and Anchors

- Block vs Inline elements
  - Block elements take up full width and force a line break before and after
    - e.g., <h1>, <p>, <div>
  - Inline elements take up as much width as necessary
    - e.g., <span>, <a>





- Anchors and Links
  - Anchors define bookmarks within a document, which can be used by links
  - Links refer to (other) documents or elements within (other) documents

```
<a name="name1">Link text</a>
<a id="id1">Link text</a>
```

```
<a href="http://www.gibts.net/index.html">Link text</a>
<a href="index.html#name1">Link text</a>
<a href="#id1">Link text</a>
```

# XHTML 1.1

## Form Controls

- Buttons

```
<input type="submit" value="Submit" />
```

- Checkboxes

```
<input type="checkbox" name="…" value="…" />
```

- Radio Buttons

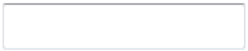
```
<input type="radio" name="…" value="…" />
```

- Menus

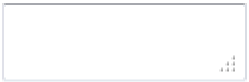
```
<select>
  <option value="EWA">EWA</option>
  …
</select>
```

- Text Input
  - Text Field
  - Text Area

```
<input type="text" />  <input type="password" />
```



```
<textarea type="text" rows="2" cols="50">
</textarea>
```



- File Select

```
<input type="file" />
```

- Hidden Controls

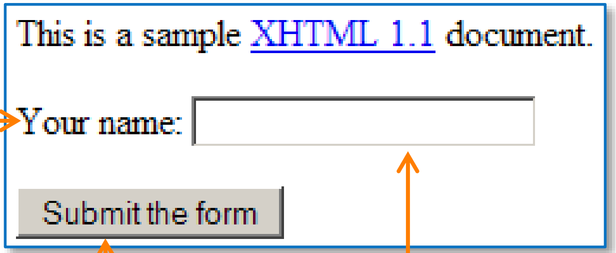
```
<input type="hidden" name="…" value="…"  />
```

  - Only for storing values between different sites
  - Not for sensitive data!

# XHTML 1.1

Forms Example

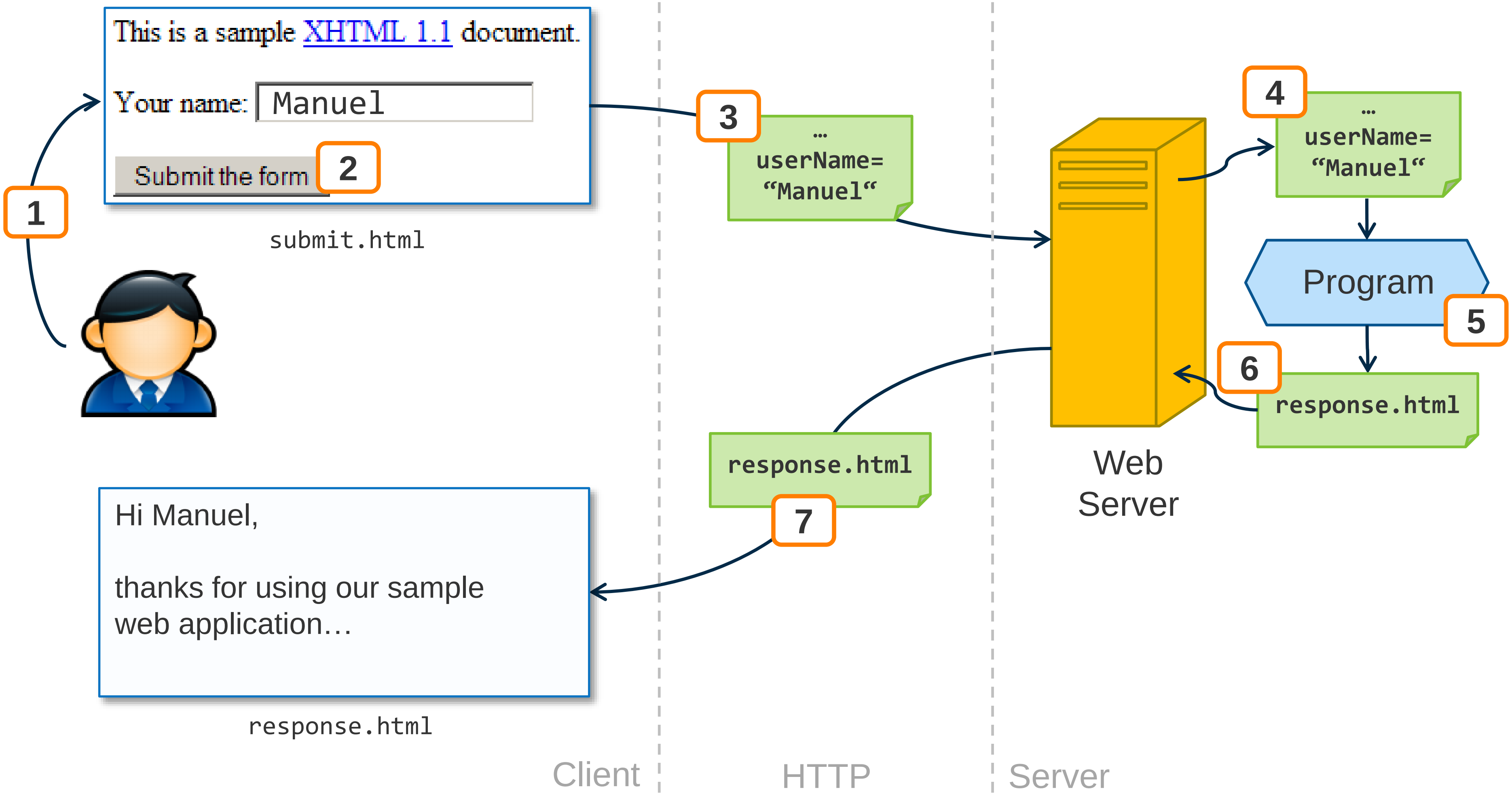
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
  <head>
    <title>A XHTML1.1 Document</title>
  </head>
  <body>
    <p>
      This is a sample <a href="...">XHTML 1.1</a> document.
    </p>
  <form action="process.jsp" method="post">
    <p>
      <label for="userName">Your name:</label>
      <input type="text" id="userName" name="userName" />
    </p>
    <p>
      <input type="submit" value="Submit the form" name="action" />
    </p>
  </form>
  </body>
</html>
```


This is a sample XHTML 1.1 document.

Your name:

# XHTML 1.1

## Forms Workflow



This is a sample XHTML 1.1 document.

Your name: Manuel

Submit the form

submit.html

Hi Manuel,

thanks for using our sample web application…

response.html

# XHTML 1.1

## Form: GET vs POST

- The method attribute of the FORM element specifies the HTTP method used to send the form to the processing agent
  - POST (Create), GET (Read), PUT (Update), DELETE (Delete), HEAD, OPTIONS, CONNECT

- **GET**
  - Requests data from a specified recourse (no side-effects)
  - Query String (name/value-pairs) of form data sent via URL to the server

```
/my/process.jsp?userName=Manuel&action=Submit+the+form
```

- **POST**
  - Submits data to be processed or changes state on server (side-effects)
  - Query String is sent in the HTTP message body

```
POST /my/process.jsp HTTP/1.1
HOST: …
userName=Manuel&action=Submit+the+form
```

# (X)HTML5

## Intro

- Goal
  - Web Documents → Web Applications
  - Updating the HTML specification
  - Consider low-powered devices (e.g., smartphones)
  - Reduce the need for external plug-ins (e.g., Flash)
  - More markup to replace scripting

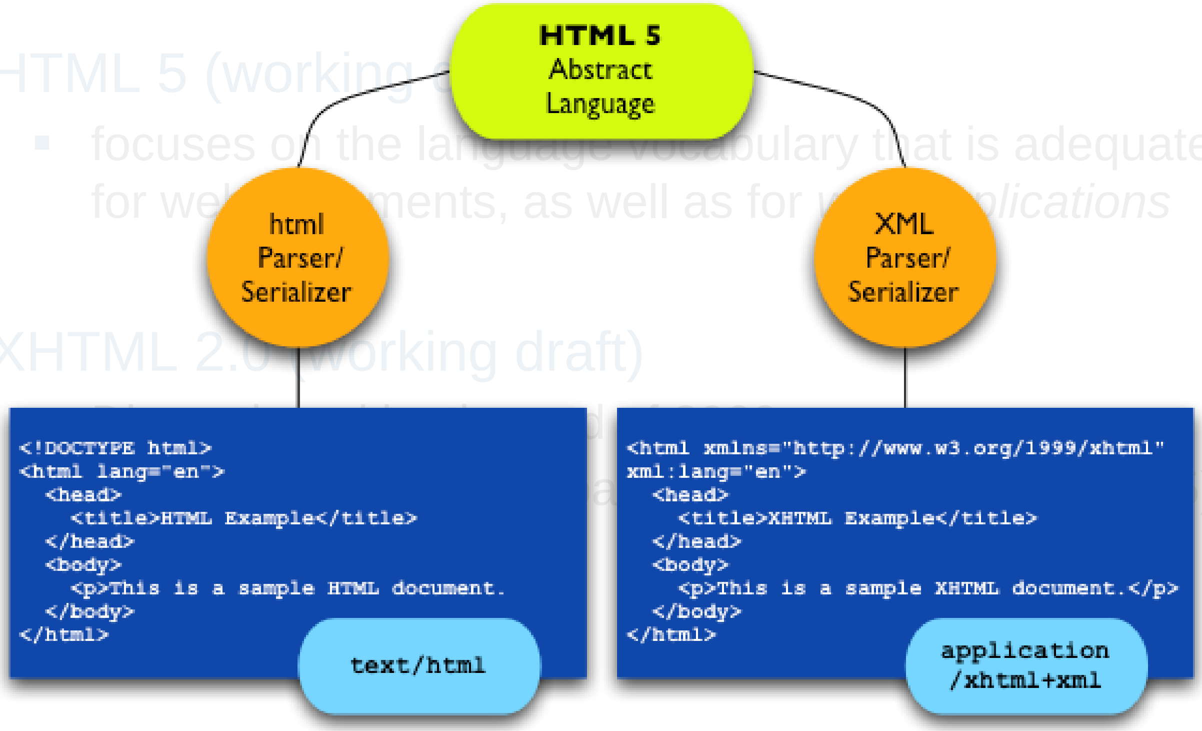
- Features
  - New markup elements
  - Form validation
  - Web storage
  - Offline support
  - APIs
  - Multimedia
  - …

# (X)HTML5

## History

2007 ongoing

- HTML 5 (working draft)
  - focuses on the language vocabulary that is adequate for web documents, as well as for web applications
- XHTML 2.0 (working draft)



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>HTML Example</title>
  </head>
  <body>
    <p>This is a sample HTML document.
  </body>
</html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en">
  <head>
    <title>XHTML Example</title>
  </head>
  <body>
    <p>This is a sample XHTML document.</p>
  </body>
</html>
```

2012

- WHATWG will develop HTML as "Living Standard"

2014

- W3C releases HTML5 recommendation (October 28)

https://html.spec.whatwg.org

# (X)HTML5

## What is new?

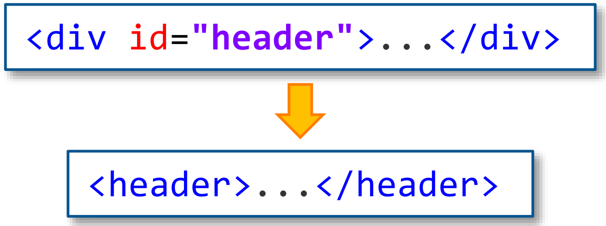
- Simpler DOCTYPE

```
<!DOCTYPE html>
```

- Simpler character encoding

```
<meta charset="UTF-8" />
```

- New elements
  - semantic elements
  - graphic elements
  - multimedia elements
  - form control



```
<div id="header">...</div>
```

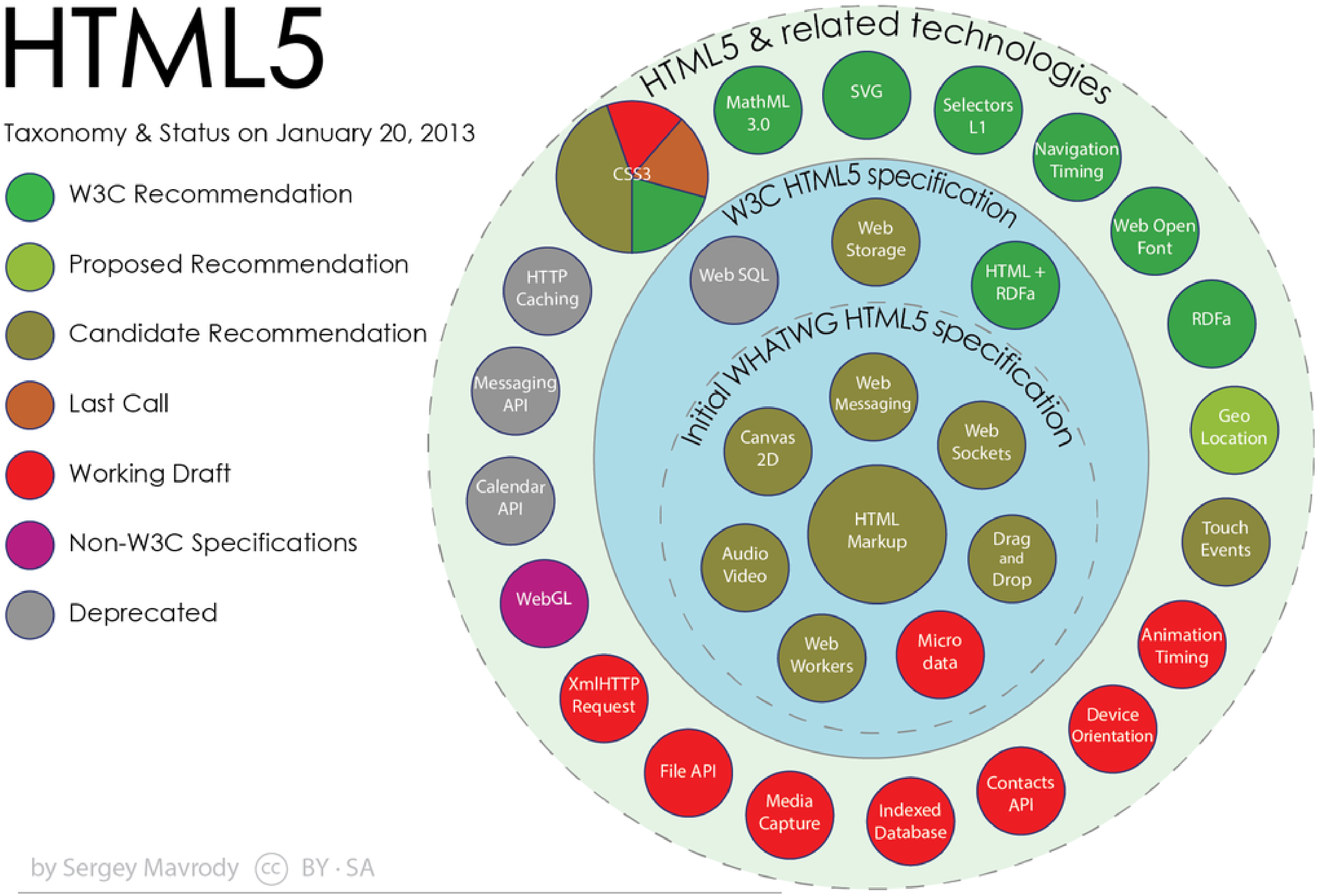
```
<header>...</header>
```

- New APIs
  - Geolocation
  - Local Storage
  - Drag and Drop
  - …

```
function getLocation() {
   if(navigator.geolocation) { // check if api is supported
      navigator.geolocation.getCurrentPosition(showPosition);
   } else { ... }
};

function showPosition(position) {
   alert("Latitute: " + position.coords.latitute + ", " +
         "Longitude: " + position.coords.longitude);
};
```

# (X)HTML5

## Related APIs



Sergey Mavrody "Sergey's HTML5 & CSS3 Quick Reference. 2nd Edition". Belisso Corp., 2012. ISBN 978-0-9833867-2-8

# (X)HTML5

## Semantic markup (excerpt)

`<header>`
defines header of document or section

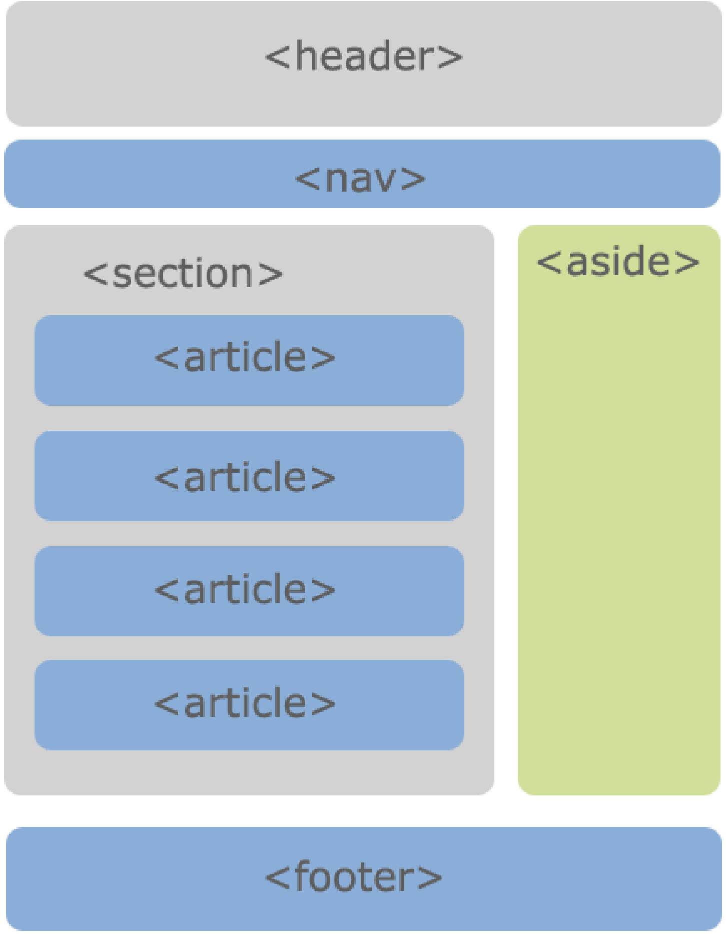
`<nav>`
defines navigation region of page or section

`<section>`
thematic grouping of content

`<article>`
specifies independent, self-contained content

`<aside>`
defines content aside from main content

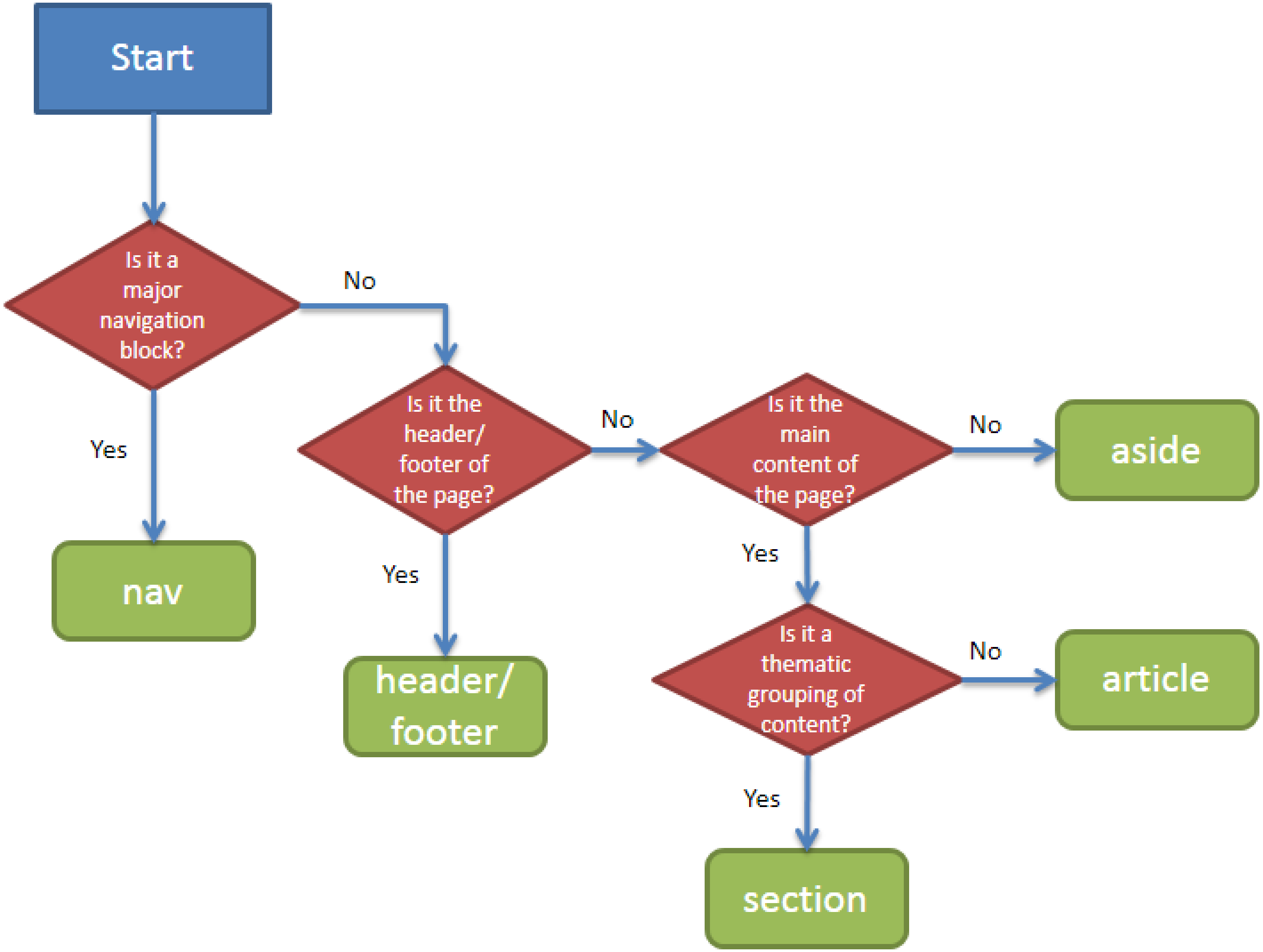
`<footer>`
defines footer of document or section



Many of these elements can be nested and it's not always straightforward which element should be used!

# (X)HTML5

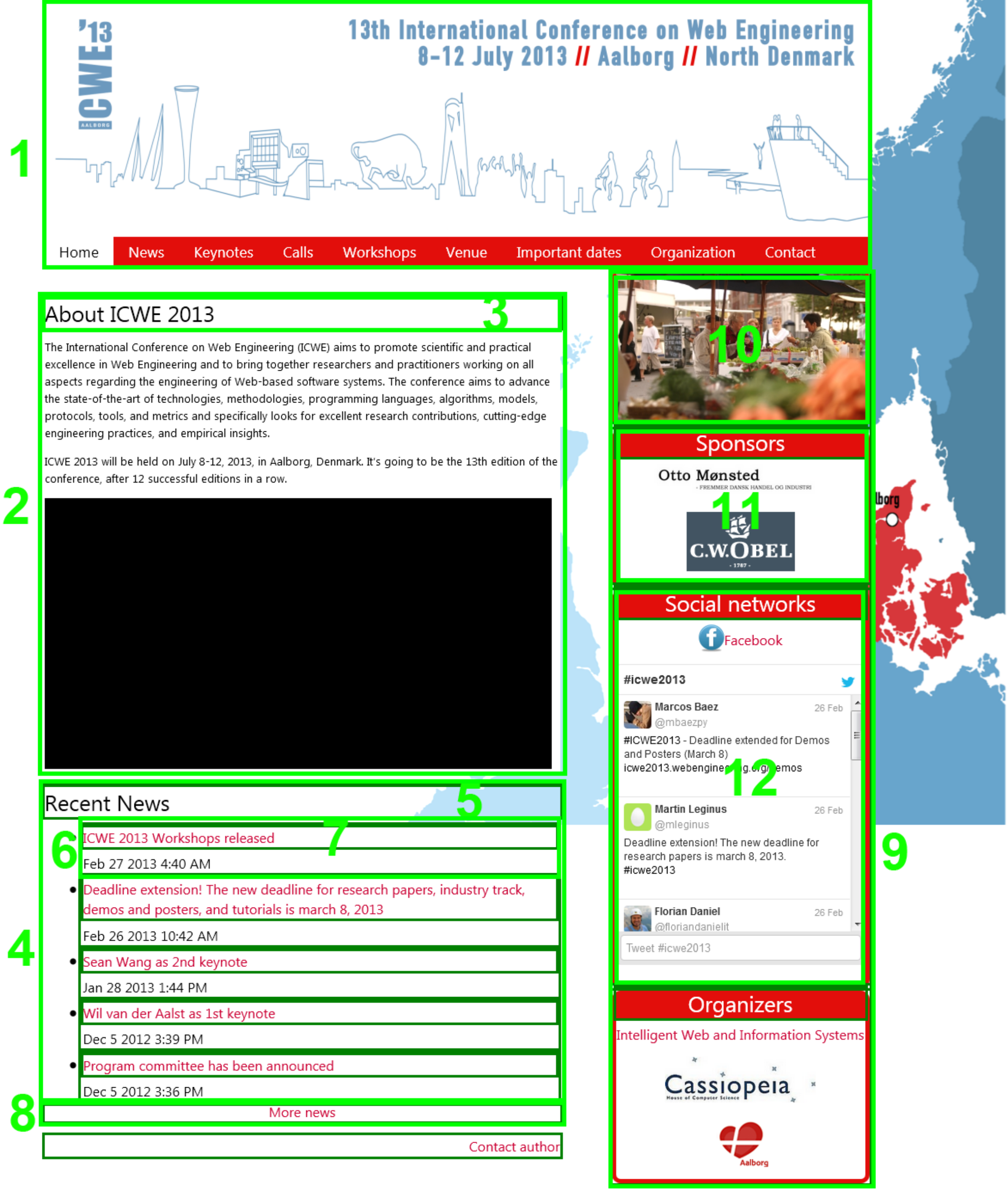
## Semantic markup (flow chart)


Start
Is it a major navigation block?
No
Yes
nav
Is it the header/ footer of the page?
No
Yes
header/ footer
Is it the main content of the page?
No
aside
Yes
Is it a thematic grouping of content?
No
article
Yes
section

This helps with the basics, but it doesn't consider multiple navs and nesting…

# (X)HTMl

Semantic m



13th International Conference on Web Engineering
8-12 July 2013 // Aalborg // North Denmark

Home | News | Keynotes | Calls | Workshops | Venue | Important dates | Organization | Contact

## About ICWE 2013

The International Conference on Web Engineering (ICWE) aims to promote scientific and practical excellence in Web Engineering and to bring together researchers and practitioners working on all aspects regarding the engineering of Web-based software systems. The conference aims to advance the state-of-the-art of technologies, methodologies, programming languages, algorithms, models, protocols, tools, and metrics and specifically looks for excellent research contributions, cutting-edge engineering practices, and empirical insights.

ICWE 2013 will be held on July 8-12, 2013, in Aalborg, Denmark. It's going to be the 13th edition of the conference, after 12 successful editions in a row.

## Recent News

- ICWE 2013 Workshops released
  Feb 27 2013 4:40 AM
- Deadline extension! The new deadline for research papers, industry track, demos and posters, and tutorials is march 8, 2013
  Feb 26 2013 10:42 AM
- Sean Wang as 2nd keynote
  Jan 28 2013 1:44 PM
- Wil van der Aalst as 1st keynote
  Dec 5 2012 3:39 PM
- Program committee has been announced
  Dec 5 2012 3:36 PM

More news

Contact author

## Sponsors

Otto Mønsted
· FREMMER DANSK HANDEL OG INDUSTRI

C.W.OBEL
· 1787 ·

## Social networks

Facebook

#icwe2013

Marcos Baez @mbaezpy 26 Feb
#ICWE2013 - Deadline extended for Demos and Posters (March 8) icwe2013.webengineering.org/demos

Martin Leginus @mleginus 26 Feb
Deadline extension! The new deadline for research papers is march 8, 2013. #icwe2013

Florian Daniel @floriandanielit 26 Feb

Tweet #icwe2013

## Organizers

Intelligent Web and Information Systems

Cassiopeia
House of Computer Science

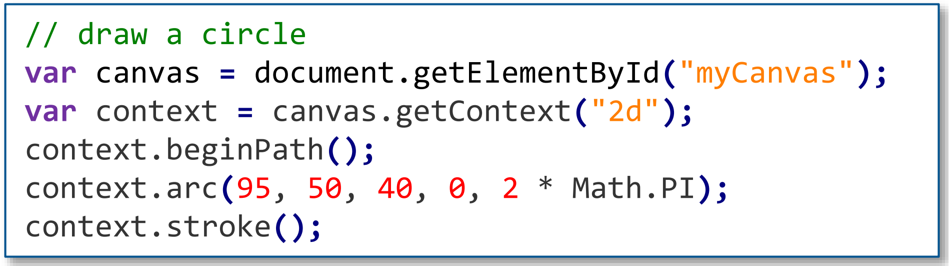
Aalborg

Aalborg

# (X)HTML5

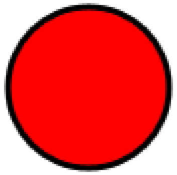
## Graphics

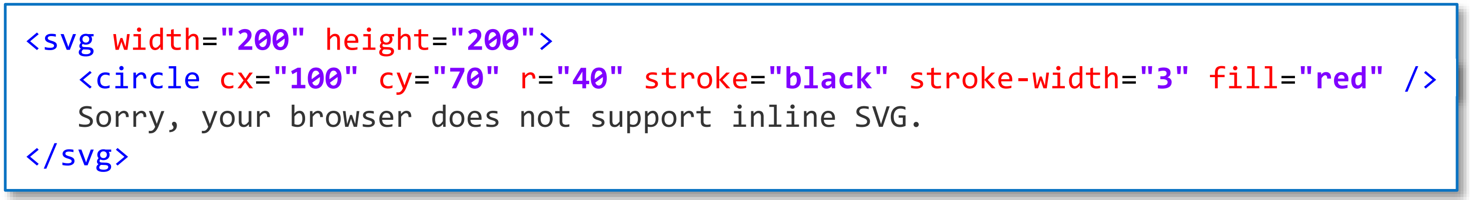
- Canvas (<canvas>)
  - Draw with JavaScript
  - Rendered pixel by pixel
  - Resolution dependent



```
// draw a circle
var canvas = document.getElementById("myCanvas");
var context = canvas.getContext("2d");
context.beginPath();
context.arc(95, 50, 40, 0, 2 * Math.PI);
context.stroke();
```

- Scalable Vector Graphics, SVG (<svg>)
  - Describe 2D graphics in XML
  - W3C recommendation
  - Available in DOM (attach JavaScript event handlers)
  - Rendered as SVG object remembered by browser
  - Resolution independant



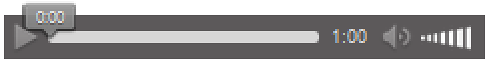


```
<svg width="200" height="200">
   <circle cx="100" cy="70" r="40" stroke="black" stroke-width="3" fill="red" />
   Sorry, your browser does not support inline SVG.
</svg>
```

# (X)HTML5

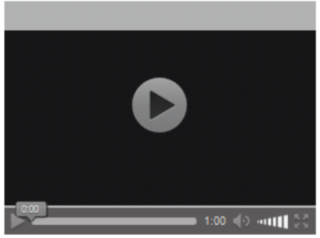
Multimedia

- Sound, music, videos, movies, animations, pictures, …
  - JavaScript to control and listen to events (pause, ended, …)
- Audio (<audio>)
  - Many formats exist (MIDI, MP3, WMA, AAC, WAV, RealAudio, …)
  - HTML5 supports MP3, WAV and Ogg



- Video (<video>)
  - Many formats exist (mpeg, avi, wmv, quicktime, flash, …)
  - HTML5 supports MP4, WebM, and Ogg

```
<video width="320" height="240" controls="controls" > // play, pause, volume
   <source src="movie.mp4" type="video/mp4" /> // first recognized format is played
   <source src="movie.ogg" type="video/ogg" />
   Sorry, your browser does not support the video tag.
</video>
```



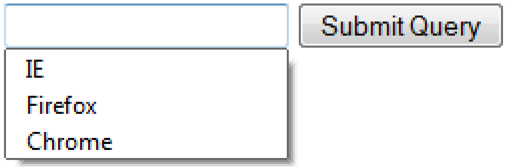
- Plug-ins (<embed> or <object>)
  - Java applets, PDF readers, Flash players, …

# (X)HTML5

## Form elements

- New form elements
  - `<datalist>` defines a list of pre-defined options
  - `<keygen>` specifies a key-pair generator
  - `<output>` represents the result of a calculation

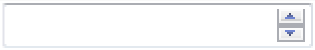
```
<input list="browsers" />
<datalist id="browsers">
   <option value="IE" />
   <option value="Firefox" />
   <option value="Chrome" />
</datalist>
```



- New form attributes
  - `autocomplete`: use previous values
  - `novalidate`: disable form validation

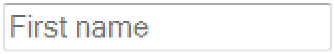
- New input types
  - color, date, number, time, url, …

```
<input type="number" min="1" max="5" />
```



- New input attributes (excerpt)
  - `pattern`: regexp for allowed values
  - `required`: field must not be empty
  - `placeholder`: suggest value for field

```
<input type="text"
       placeholder="First name" />
```



# (X)HTML5

## Web Storage

- Web Storage
  - Store data as key/value pairs on user side
  - Browser defines storage quota
- Local Storage (`window.localStorage`)
  - Store data in users browser
  - Vs Cookies: more secure, larger data, not transfered
  - No expiration date
- Session Storage (`window.sessionStorage`)
  - Store data in session
  - Data is destroyed when tab/browser is closed

```
// use web storage through JavaScript
var storage = permanent ? localStorage : sessionStorage;
if(!storage["name"]) {
   storage["name"] = prompt("Enter your name: ", "Name");
}
alert("Your name is " + storage["name"]);
```

# (X)HTML5

App Cache

- Application cache
  - Offline browsing – use application offline
  - Speed – load cached resources faster
  - Reduced server load – reduce transfer between server and client
  - Manifest file manages application cache

```
<!DOCTYPE html>
<html ... manifest="we.appcache">
...
</html>
```

- Cache is updated…
  - if user clears browser's cache
  - if manifest file is modified
  - programmatically

**Be careful what you cache!** Once a file is cached, the browser will continue to show the cached version, even if you change the file on the server.

# (X)HTML5

## App Cache

- Cache Manifest Example

```
CACHE MANIFEST
# we.appcache, 2015-03-09
# implicit 'CACHE:' section
# cached after first download
/theme.css
/logo.png
/main.js

NETWORK:
# never cache these files
login.jsp

FALLBACK:
# fallback in case of no internet
# serve offline.html if path at /html/
# can not be accessed
/html/
/offline.html
```

# XTHML, (X)HTML5

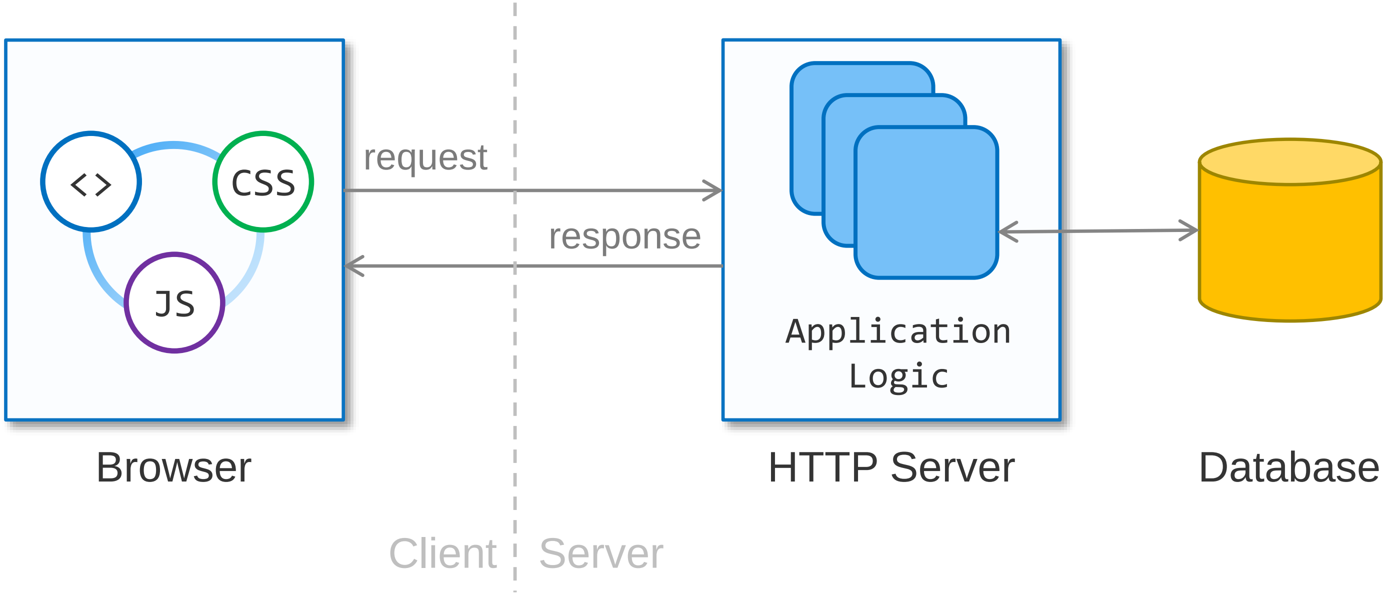
## Resources

- Standards / Validation
  - http://www.w3.org/TR/xhtml11/
  - http://www.w3.org/TR/html5/
  - https://html.spec.whatwg.org/
  - http://validator.w3.org/
  - https://html5.validator.nu/
- Tutorials
  - http://www.w3schools.com/htmldom
  - http://www.w3schools.com/html/html5_intro.asp
  - http://selfhtml.org/html
- Check browser support for HTML5 with "Can I use ___?"
  - http://caniuse.com/

# Basic Technologies

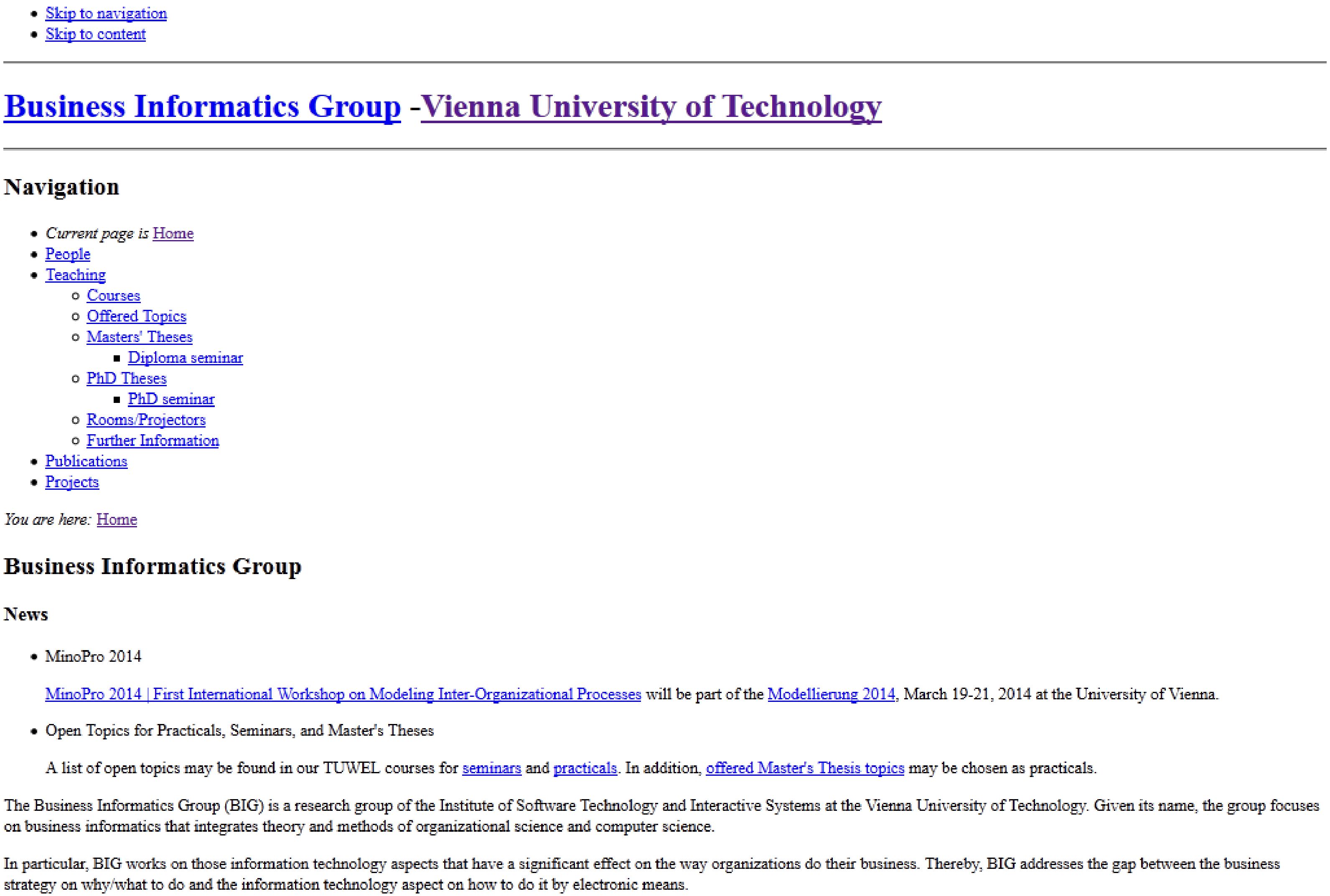
Overview

- Client Side
  - (X)HTML for structure and content ✓
  - CSS for layout and style
  - JavaScript for client-side functionality



# CSS

## Motivation



- Skip to navigation
- Skip to content

# Business Informatics Group -Vienna University of Technology

## Navigation

- *Current page is* Home
- People
- Teaching
  - Courses
  - Offered Topics
  - Masters' Theses
    - Diploma seminar
  - PhD Theses
    - PhD seminar
  - Rooms/Projectors
  - Further Information
- Publications
- Projects

*You are here:* Home

## Business Informatics Group

### News

- MinoPro 2014

  MinoPro 2014 | First International Workshop on Modeling Inter-Organizational Processes will be part of the Modellierung 2014, March 19-21, 2014 at the University of Vienna.

- Open Topics for Practicals, Seminars, and Master's Theses

  A list of open topics may be found in our TUWEL courses for seminars and practicals. In addition, offered Master's Thesis topics may be chosen as practicals.

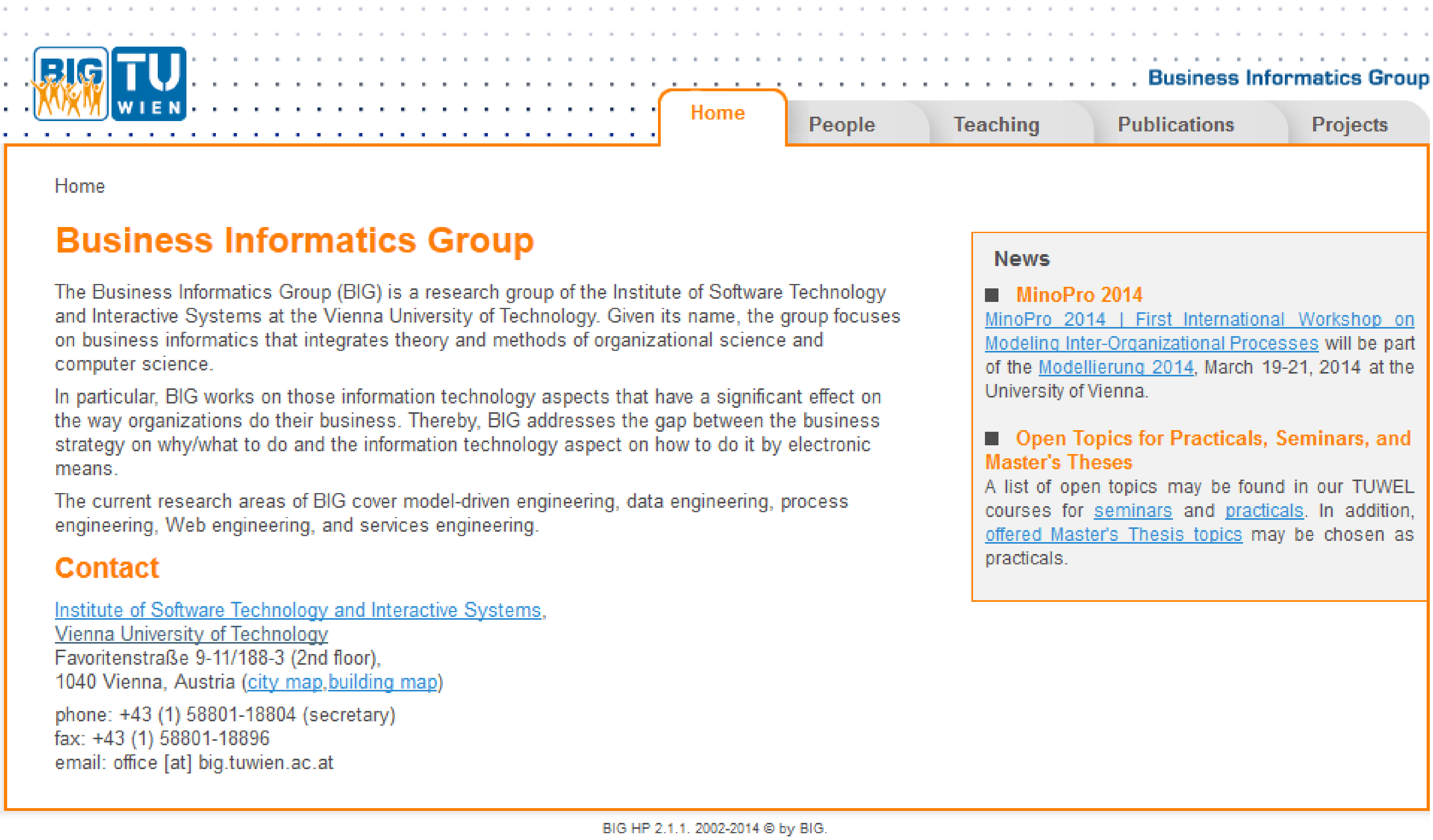
The Business Informatics Group (BIG) is a research group of the Institute of Software Technology and Interactive Systems at the Vienna University of Technology. Given its name, the group focuses on business informatics that integrates theory and methods of organizational science and computer science.

In particular, BIG works on those information technology aspects that have a significant effect on the way organizations do their business. Thereby, BIG addresses the gap between the business strategy on why/what to do and the information technology aspect on how to do it by electronic means.

Our website without CSS – Default Browser Rendering

# CSS

## Motivation



Home

# Business Informatics Group

The Business Informatics Group (BIG) is a research group of the Institute of Software Technology and Interactive Systems at the Vienna University of Technology. Given its name, the group focuses on business informatics that integrates theory and methods of organizational science and computer science.

In particular, BIG works on those information technology aspects that have a significant effect on the way organizations do their business. Thereby, BIG addresses the gap between the business strategy on why/what to do and the information technology aspect on how to do it by electronic means.

The current research areas of BIG cover model-driven engineering, data engineering, process engineering, Web engineering, and services engineering.

## Contact

Institute of Software Technology and Interactive Systems,
Vienna University of Technology
Favoritenstraße 9-11/188-3 (2nd floor),
1040 Vienna, Austria (city map,building map)

phone: +43 (1) 58801-18804 (secretary)
fax: +43 (1) 58801-18896
email: office [at] big.tuwien.ac.at

### News

- **MinoPro 2014**
  MinoPro 2014 | First International Workshop on Modeling Inter-Organizational Processes will be part of the Modellierung 2014, March 19-21, 2014 at the University of Vienna.

- **Open Topics for Practicals, Seminars, and Master's Theses**
  A list of open topics may be found in our TUWEL courses for seminars and practicals. In addition, offered Master's Thesis topics may be chosen as practicals.

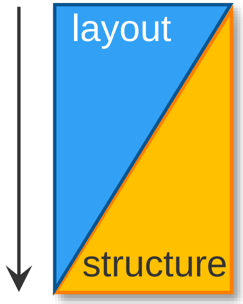
BIG HP 2.1.1. 2002-2014 © by BIG.

Our website with CSS

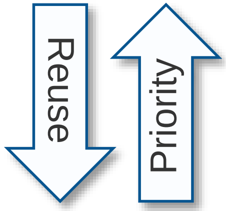
# CSS

## What is CSS?

- Cascading Style Sheets
- Describes the style and layout of a document (XML, HTML, XHTML,…)
- Recommended by the W3C to separate content and design
  - Initial problem when style and content were mixed
  - Layout got removed gradually with every new standard



- Levels: CSS1 ⊆ CSS2 / CSS 2.1 ⊆ CSS3
- Integration into HTML
  - Inline – using the style attribute in elements
  - Internal – using the <style> element in <head>
  - External – linking to an external CSS file in <head>



  - External way preferred to separate structure/content and style/layout

```
element-selector {
   property:value;
}
```

# CSS 2.1

## Selectors

- Type Selector
  - Select a group of elements via their name

```
<h1>...</h1>

h1 { font-size: 12pt; }
```

- ID Selector
  - Select a single unique element via id ('#')

```
<p id="first">…</p>

#first { color: red; }
```

- Class Selector
  - Select a group of elements via class ('.')

```
<p class="small">…</p>
<h1 class="small">…<h1/>

.small { font-size: 5pt; }
```

- Additional Selectors
  - Descendants: Separate using white-space
  - Children: Separate using '>'
  - Siblings: Separate using '+'
  - Attribute: Specify attribute via '[att=val]'

```
body p { ... }
body > p { ... }
p + div { ... }
h1[title] { ... }
h1[title="a"] { ... }
```

- Selectors can be grouped by separating them via comma ','
- Specificity determines which style is applied when multiple rules apply

# CSS 2.1

Some Properties

- Formatting Text/Fonts
  - Font family, style, size, and weight
    - Use font fallback
  - Color
  - Line Height
  - Text Alignment

```
font-family: Arial, sans-serif;
font-style: italic;
font-size: 1.2em;
font-weight: bold;
color: #00ff00;
line-height: 120%;
text-alignment: center;
```

- Background
  - Color
  - Image, Repeat, Attachment, Position

```
background-color: rgb(250,20,16);
background-image: url("bg.jpg");
background-repeat: repeat-x;
background-position: right top;
```

- Lists
  - Item marker or Image

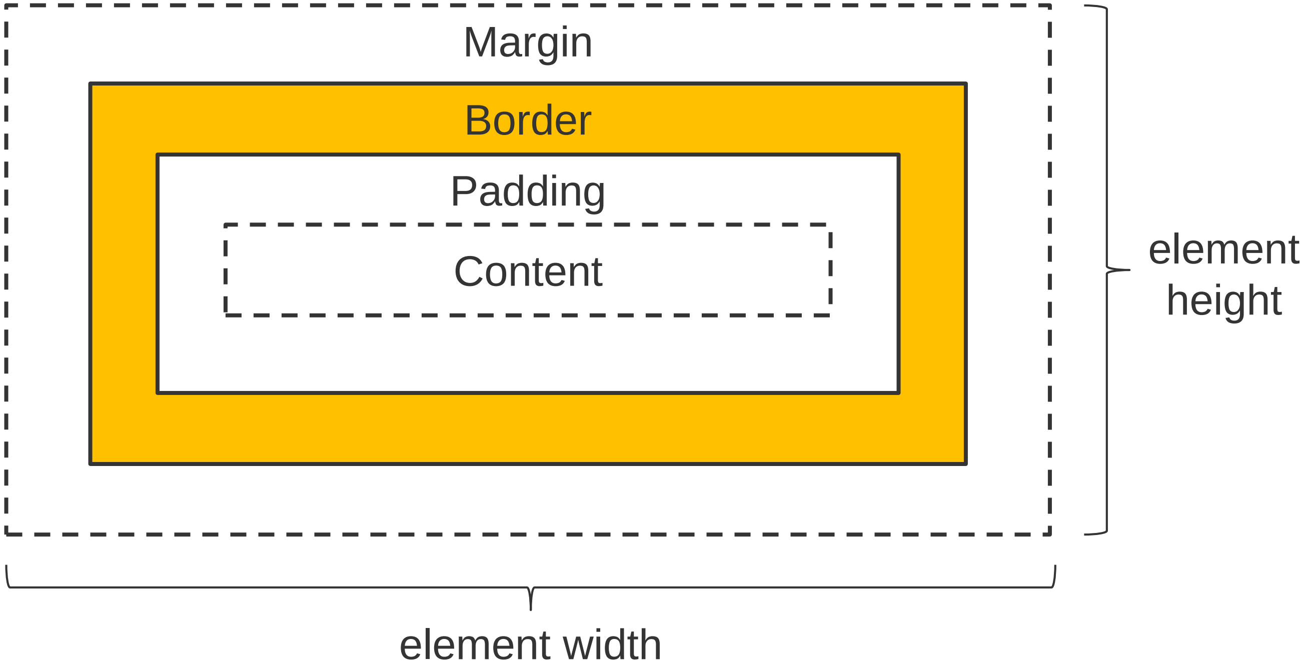
```
list-style-type: circle;
list-style-image: url('logo.gif');
```

- …

# CSS 2.1

Box Model

- CSS Box Model
  - Content width and height
  - Margin, Padding, and Border can be set for left, right, top, bottom
  - IE8 and earlier versions of IE calculated the element width differently!
    - Included Padding and Border in the content width property
    - Much work for developers to handle all browsers



# CSS 2.1

Layout

- Positioning of Elements
  - Standard page flow is static
    - horizontal, one element after another (inline vs block)
  - Coordinates can position elements differently (top, bottom, left, right)
    - Fixed: Element removed from flow
    - Relative: Position relative to position in flow (original space still taken)
    - Absolute: Position relative to first non-static parent or html
    - z-index defines which elements should be placed in front

```
position: absolute;
left: 10px;
top: 10px;
```

- Floating of Elements
  - Push element left or right
  - Following elements float around
  - Use clear to turn floating off

```
float: left;
float: right;
clear: left;
clear: both;
```

- Display and visibility of Elements
  - Make element behave as block or inline element
  - Hide elements (hidden still takes space!)

```
display: inline;
```

```
visibility: hidden;
```

# CSS 2.1

## Size and Proportion Values

- Absolute values
  - For fixed sized rendering (printed pages, images)
  - Inches (in), Centimeters (cm), Millimeters (mm), Points (pt), Picas (pc)
- Absolute/Relative values
  - Pixel (px): Relative to screen resolution, but absolute for output device
- Relative values
  - For screen rendering and easy accessible content (change base font size)
  - em (relative to font square), ex (relative to letter 'x'), % (relative to parent)
- Calculation of sizes depends on browser (default font size) and OS (pixes size, default scaling of a system font)

→ W3C recommends em size unit

# CSS 2.1

Advanced Topics

- Pseudo-classes / Pseudo-elements
  - Use information present outside the document tree
  - Pseudo-classes
    - :first-child, :link, :hover, :active, :focus, :lang
  - Pseudo-elements
    - :first-line, :first-letter, :before, :after

```
a:hover { color: #ff0099; }
```

```
p:first-letter { font-size: 20px; }
```

- Media Types
  - CSS depending on media type
  - all, print, braille, handhelt, screen, tv, …

```
@media screen {
    p { … }
}
```

- Vendor Prefixes
  - Proprietary browser extensions (no specification!)
    - May work differently or not at all in other browsers!
    - Sometimes used to solve browser issues

```
-webkit-transition: …;
-moz-transition: …;
-ms-transitional: …;
-o-transition: …;
```

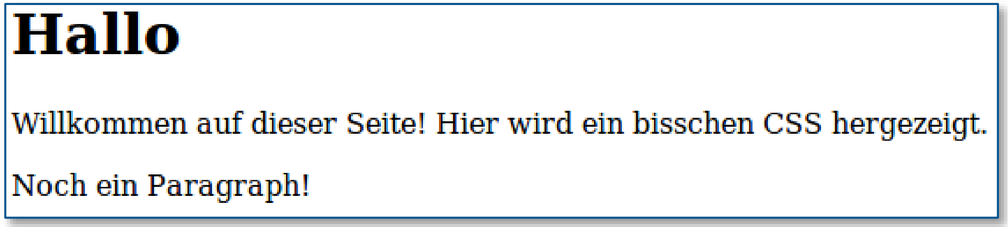
- Reset CSS
  - Should remove inconsistent default styling of browsers!
  - Beware of specificity

# CSS 2.1

## Example 1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de">
  <head>
    <title>CSS example</title>
    <link rel="stylesheet" type="text/css" href="screen.css" />
  </head>
  <body>
    <h1>Hallo</h1>
    <p>
    Willkommen auf <span class="class1">dieser Seite</span>!
    Hier wird ein bisschen <span class="class1">CSS</span> hergezeigt.
    </p>
    <p id="secondPar">
    Noch ein <span class="class1">Paragraph</span>!
    </p>
  </body>
</html>
```

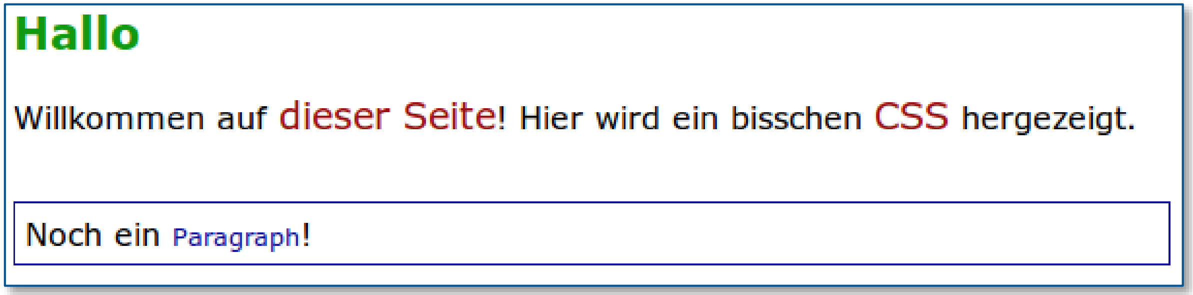
```
html {
  font-family: Verdana, sans-serif;
}

h1 {
  font-size: 1.4em;
  color: #119911;
}

.class1 {
  font-size: 1.2em;
  color: #991111;
}

#secondPar {
  border: 1px solid #111199;
  padding: 0.3em 0.3em 0.3em 0.3em;
  margin-top: 2em;
}

#secondPar .class1 {
  font-size: 0.8em;
  color: #111199;
}
```



Hallo

Willkommen auf dieser Seite! Hier wird ein bisschen CSS hergezeigt.

Noch ein Paragraph!



Hallo

Willkommen auf dieser Seite! Hier wird ein bisschen CSS hergezeigt.

Noch ein Paragraph!

# CSS3

Intro

- Fully backwards compatible to CSS2
- Modules
  - Selectors
  - Box Model
  - Background and Borders
  - Image Values and Replaced Content
  - Text Effects
  - 2D/3D Transformations
  - Animations
  - Multiple Column Layout
  - User Interface
  - …

"There will never be a CSS4!"
*- Tab Atkins Jr, member of CSS Working Group*

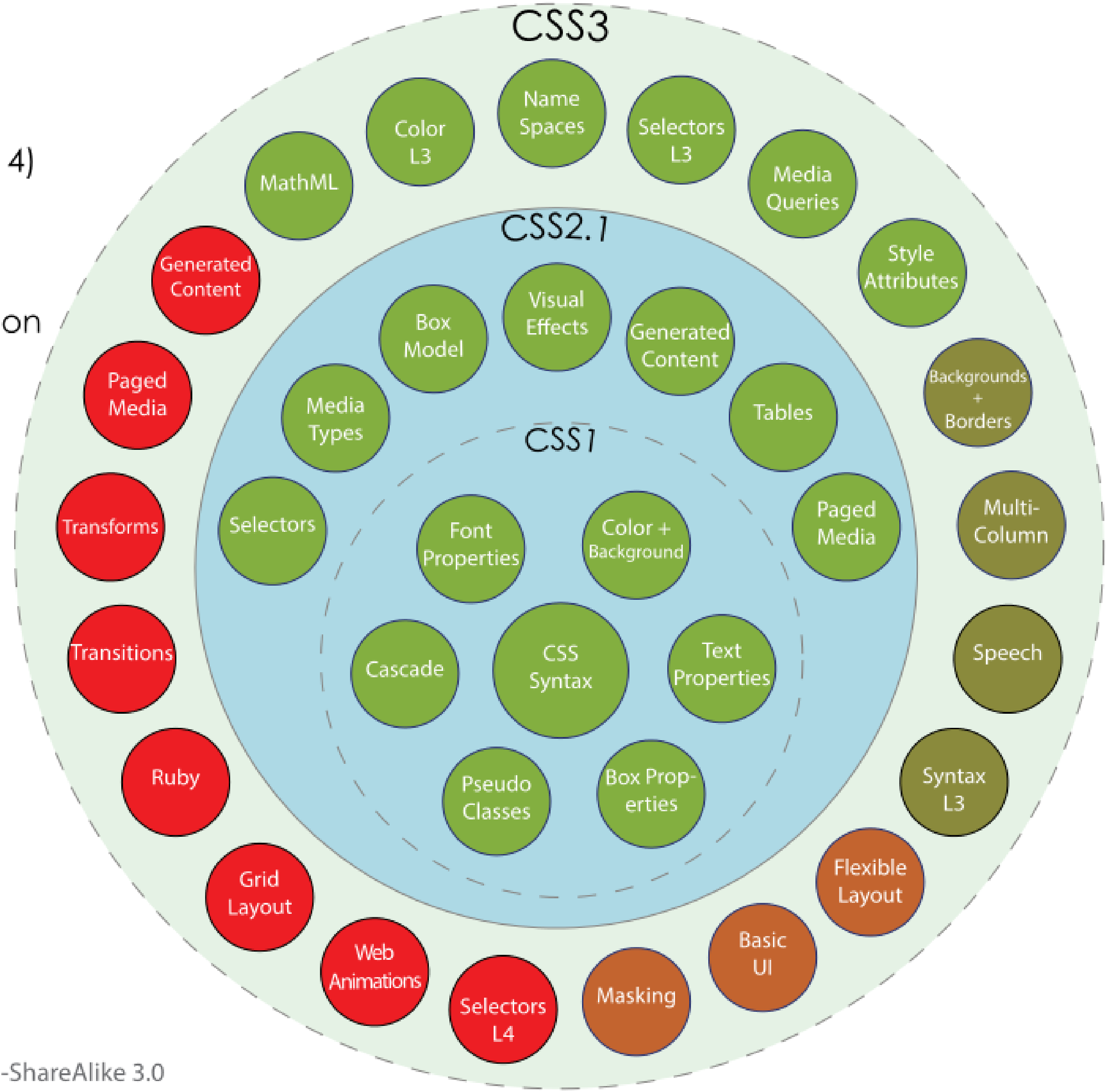
# CSS3

Modules

## CSS3

Taxonomy & Status (October 2014)

- 🟢 W3C Recommendation
- 🟤 Candidate Recommendation
- 🟠 Last Call
- 🔴 Working Draft
- ⚪ Obsolete or inactive



By Sergey Mavrody 2011-14 | CC Attribution-ShareAlike 3.0

# CSS3

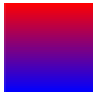
## Examples

- Borders
  - Rounded corners
  - Shadow

```
border: 2px solid #A1A1A1;
border-radius: 25px;
```

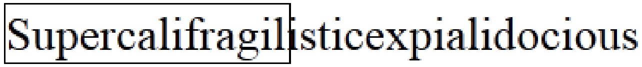
```
border: 1px solid black;
box-shadow: 3px 3px 3px #FF9900;
```

- Gradient
  - Linear gradient
  - Radial gradient

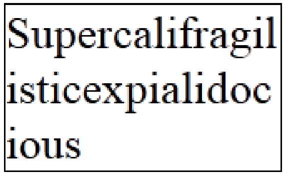


```
background: -webkit-linear-gradient(red, blue);
background: -o-linear-gradient(red, blue);
background: -moz-linear-gradient(red, blue);
background: linear-gradient(red, blue);
```

- Text
  - Shadow
  - Word Wrapping



Supercalifragilisticexpialidocious



Supercalifragil
isticexpialidoc
ious

```
border: 1px solid black;
width: 6em;
word-wrap: break-word;
```

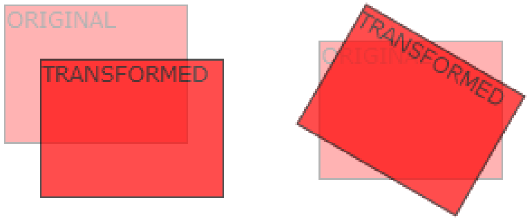
- Selectors
  - More detailed selection

```
p:nth-child(2) { ... }
p:only-child { ... }
input:required { ... }
```

# CSS3

## Examples

- 2D Transformations
  - Rotate, translate, scale, skew, matrix



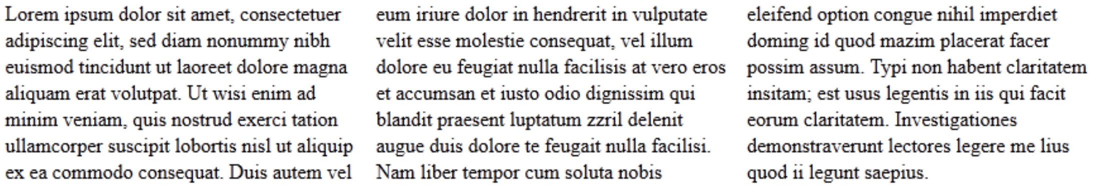
- Transitions
  - Gradual change

```
transition: width 2s;
```

- Multiple columns
  - count, gap, rule

```
<p>Lorem ipsum dolor
sit amet, consectetuer
adipiscing elit...</p>
```

```
-webkit-column-count: 3;
-moz-column-count: 3;
column-count: 3;
```



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.

- Web fonts
  - Individual fonts
  - Formats: TTF, OTF, WOFF, …
  - @font-face rule

```
@font-face {
   font-family: myFamily;
   src: url(my_font.ttf);
}
p { font-family: myFamily; }
```

# CSS3

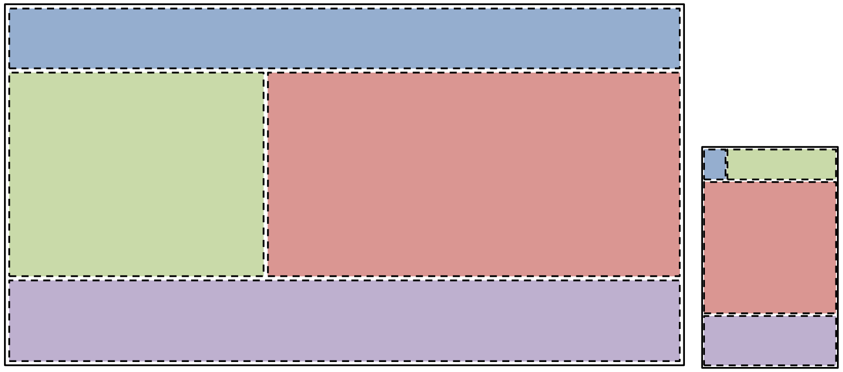
Media Queries

- Media Queries
  - Previously only media types (screen, print, braille, handheld …)
  - @media rule
  - Additional features
    - color
    - aspect-ratio
    - max-width
    - orientation
    - resolution
    - scan
    - …
  - Build complex queries using logical operators (not, and, only)

```
@media only screen and (max-width: 500px) { … }
@media tv and (min-width: 700px) and (orientation: landscape) { … }
<!-- comma acts as 'or' -->
@media (min-width: 700px), handheld and (orientation: landscape) { … }
```

# CSS3

## Media Queries



```css
#header {
  height: 200px;
}
#navigation {
  width: 300px;
}
#content {
  width: 900px;
}
#footer {
  height: 200px;
}
```

```css
@media screen and (max-width: 768px) {
  #header {
    width: 80px;
    height: 120px;
    display: inline;
  }
  #navigation {
    width: 560px;
    height: 120px;
  }
  #content {
    width: 640px;
    height: 760px;
  }
  #footer {
    height: 80px;
  }
}
@media screen and (max-width: 1024px) {
   ...
```

# Further Literature

Resources

- Validation
  - http://jigsaw.w3.org/css-validator/

- Resources
  - http://de.selfhtml.org/css (german)
  - http://www.css4you.de/ (german)
  - http://www.w3schools.com/css/ (english)
  - http://www.w3schools.com/css/css3_intro.asp (english)

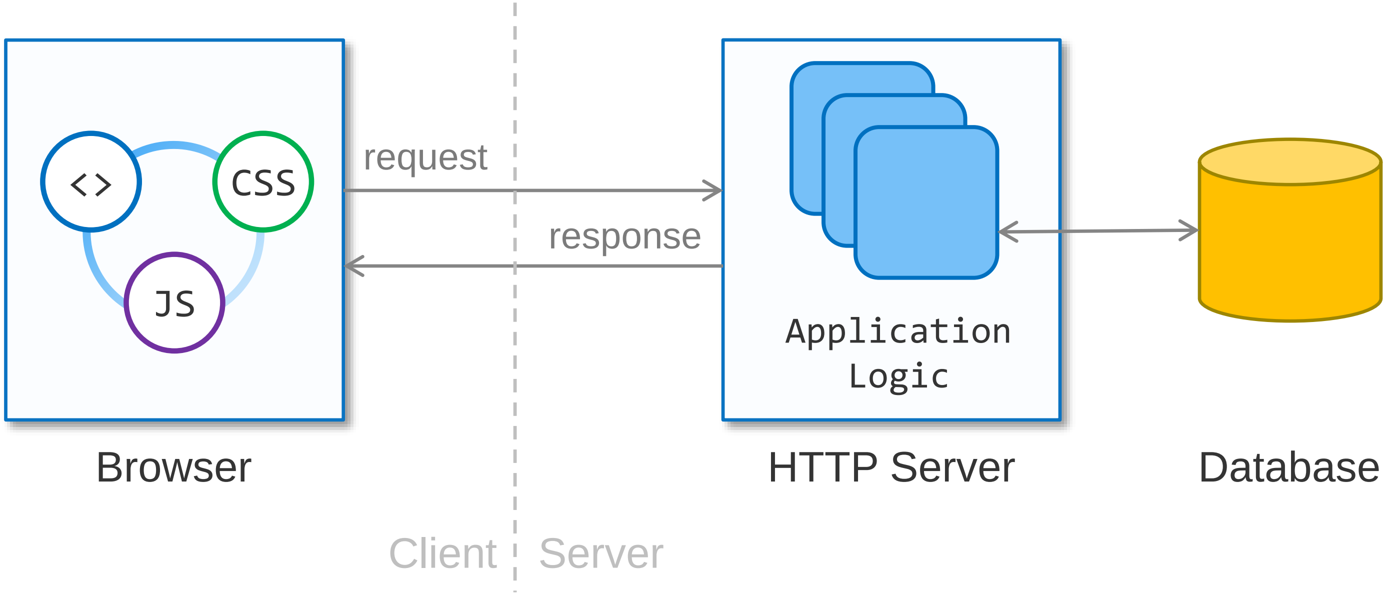
- Check browser support for CSS3 with "Can I use ___?"
  - http://caniuse.com/

- Literature
  - R. Andrew, D.Shafer, *CSS*, dpunkt.verlag, 2006, ISBN: 3-89864-423-5.

# Basic Technologies

Overview

- Client Side
  - (X)HTML for structure and content ✓
  - CSS for layout and style ✓
  - JavaScript for client-side functionality



# JavaScript

## What is JavaScript?

- JavaScript is an interpreted scripting language
  - Originally developed by Netscape in 1995
  - Standardized as ECMAScript language in the ECMA-262
    - Other ECMAScript implementations: JScript, ActionScript, QtScript, …
  - Integration into HTML
    - Internal in <head> – using the <script> element
    - Internal in <body> – using the <script> element
    - External – linking to an external JavaScript file in <head>

```
<script type="text/javascript">
  statement;
  statement;
  ...
</script>
<script type="text/javascript" src="myScript.js" />
```

  - JavaScript is executed as soon as it is read!

# JavaScript

## Basics

- Variables have dynamic types
  - Changing data changes type
  - String, Number, Boolean, Array
  - undefined if there is no value
  - null empties the variable

```
var data;                // data undefined
data = true;             // Boolean object
data = '7';              // String object
data = Number(data);     // Number object
data = new Array();      // Array object
data[0] = "A";              // 0-based
data = new Array("A");      // condensed
data = ["A"];               // literal
data = null;             // data is empty
```

- Objects have no class!
  - (Almost) everything is an object!
  - Objects are associative arrays
    - key = property
    - Added when value assigned

```
var student = new Object();
student.nr = 'e120...';
student.name = 'A Student';
student.age = 19;
student.hasSteop = false;

student = { nr : 'e120...',
            name : 'A Student',
            age : 19,
            hasSteop : false };
```

- Events
  - List of pre-defined events for objects
  - Mouse, keyboard, forms, loading, ...

```
<img src="..." alt="..."
     onClick="alert('Hello');" />
```

# JavaScript

## Basics

- Functions ('function') are first-class
  - Functions are objects too!
  - Must be explicitely called
  - Might have arguments (no types!)
  - Might have a return value
  - Local variables use 'var' when declared

```
function myFunction(a, b) {
   var localVariable = a + b;
   return 5;
   // nothing executed after return
}

// call replaced by return value
x = myFunction('A ', 'Student');
```

- "Methods"
  - = function definition assigned as properties (use 'new' and 'this')

```
function student(nr, name, age, hasSteop) {
   this.nr = nr;
   this.name = name;
   this.age = age;
   this.hasSteop = hasSteop;

   this.finishSteop = function() {
     this.hasSteop = true;
   }
}

var aStudent = new student('e120…', 'A Student', 19, false);
aStudent.finishSteop();
```

# JavaScript

Basics

- All objects have a prototype
  - All prototype are objects
  - Any object can be a prototype
  - Objects inherit properties and methods from prototype
  - Object.prototype is top of prototype chain (its prototype is null)

```
var aStudent = { nr : 'e120...', ... };    // aStudent  -> Object.prototype
var students = [ aStudent, ... ];          // students  -> Array.prototype    -> Object.prototpye
var aoStudent = new student('e120…', ...); // aoStudent -> Student.prototype  -> Object.prototype
function print() { ... };                  // print     -> Function.prototype -> Object.prototype
```

- Existing prototypes can be extended at any time
  - Beware of monkey patching!

```
function student(nr, name, age, hasSteop) { ... } // as before
student.prototype.university = 'TU Wien';         // add property
student.prototype.summary = function() {          // add "method"
   return this.nr + ' ' + this.name;
};

String.prototype.distance = function() { ... };   // monkey patching
```

# JavaScript

## Document Object Model

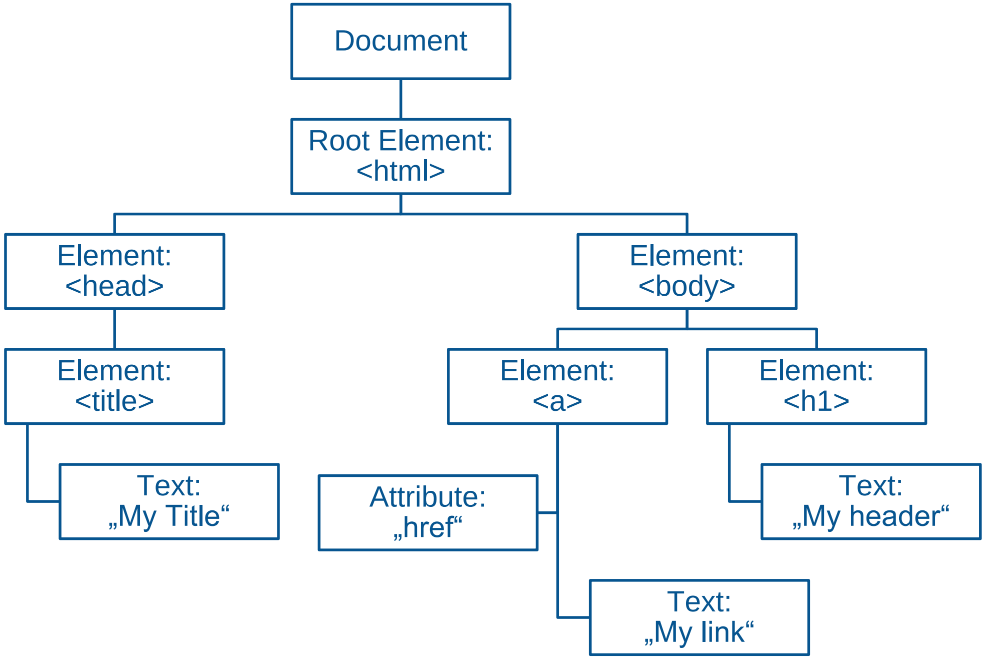
- Browser Object Model (BOM)
  - Allows access to browser objects
  - Not standardized! (But very similar in all modern browsers)
- Window is the global object
  - All global objects, functions, and variables are members of window

| Object | Property and Methods |
|---|---|
| window | Other global objects, open(), close(), moveTo(), resizeTo() |
| screen | width, height, colorDepth, pixelDepth, … |
| location | hostname, pathname, port, protocol, assign(), … |
| history | back(), forward() |
| navigator | userAgent, platform, systemLanguage, … |
| document | body, forms, write(), close(), getElementById(), … |
| *Popup Boxes* | alert(), confirm(), prompt() |
| *Timing* | setInterval(), setTimeout() |
| ... | ... |

# JavaScript

Document Object Model

- Document Object Model (DOM)
  - Tree structure for interacting with (X)HTML and XML documents
    - HTML elements as objects with properties, methods and events
  - Standardized by the W3C
    - Platform- and language-independent
    - Levels: Level 1 ⊆ Level 2 ⊆ Level 3
    - Browsers still implement it slightly differently!



# JavaScript

## Document Object Model

- Retrieving Elements
  - ID, tag name, class name
  - Document property

```
title = document.getElementById("title");
links = document.getElementsByTagName("a");
greens = document.getElementsByClassName("green");
imgs = document.images;
```

- Change Elements
  - Content (innerHTML)
  - Element attributes
  - Element style

```
title.innerHTML = "newTitle";
links[0].href = "http://...";
greens[0].style.color = "red";
```

- Manipulating DOM Nodes
  - Create, append, remove, ...

```
var header = document.createElement("h2");
var text = document.createTextNode("SubTitle");
header.appendChild(text);
document.removeChild(title);
document.replaceChild(title, header);
```

- Register Events
  - Separate behavior and structure
  - Cleaner than to add directly in the HTML element

```
header.onclick = function() { alert('Hello!'); }
```

# JavaScript

Best practices

- Avoid browser-specific extensions wherever possible
  - Stay close to the standard
  - Test functionality before you use it
- Don't pollute global objects or global "namespaces"

```
at.ac.tuwien.we.Grading = function() { ... }
```

- Separate behavior from markup
  - Avoid inline references to JavaScript

```
<input name="name" type="text"
       onchange="validateName()" />
```

```
<input name="name" type="text" />
```

```
window.onload = function() {
   document.getElementByName("name")
           .onchange = validateName;
}
```

- Use already known best practices
  - Use encapsulation
  - Follow meaningful naming conventions
  - Apply design patterns
  - Test your code
  - …

# jQuery

JavaScript library

- JavaScript library jQuery
  - Abstract from browser differences
  - More concise code ("write less, do more")
  - Add commonly used functions, components, patterns, effects, …
    - HTML/DOM manipulation using CSS selectors
    - CSS manipulation
    - HTML event methods
    - Effects and animations
    - ...
  - Provide additional plugins for many other tasks
  - Integration in HTML
    - Download and link to JavaScript-File
    - Use Content Delivery Network (CDN)
      - Refer to file hosted by Google or Microsoft
      - Can re-use cache if multiple visited sites link to same CDN

http://jquery.com/

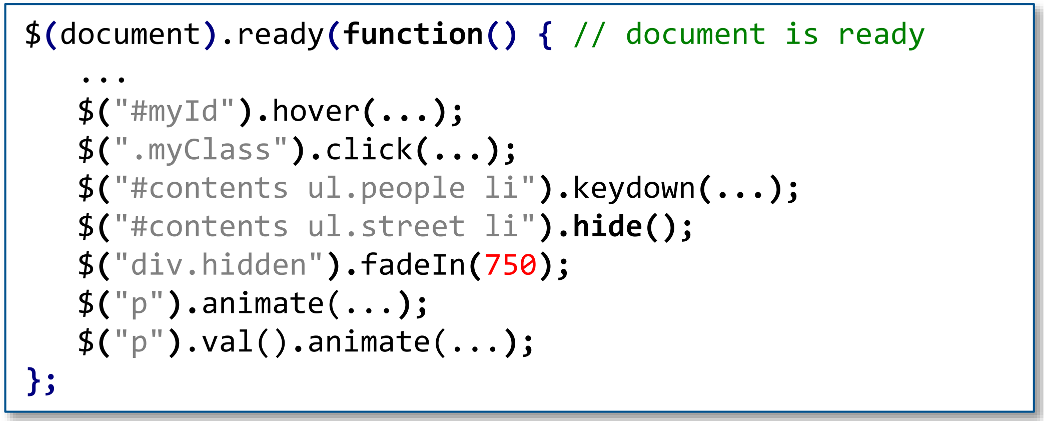
# jQuery

JavaScript library

- JavaScript library jQuery

$(selector).action()

  - Re-using CSS selectors
  - Several actions and events/effects supported
  - Good practice: Wait until document is fully loaded



```
$(document).ready(function() { // document is ready
   ...
   $("#myId").hover(...);
   $(".myClass").click(...);
   $("#contents ul.people li").keydown(...);
   $("#contents ul.street li").hide();
   $("div.hidden").fadeIn(750);
   $("p").animate(...);
   $("p").val().animate(...);
};
```

http://jquery.com/

# Java libraries

Excerpt

- Other well-known JavaScript libraries
  - Modernizr
    - Detects HTML5 and CSS3 features in different browsers
    - http://modernizr.com/
  - Node.js
    - Build fast, scalable applications ("real-time web applications")
    - http://nodejs.org/
  - AngularJS
    - Develop single-page applications using MVC architecture
    - https://angularjs.org/
  - Prototype
    - Common base library to support OOP and Ajax
    - http://prototypejs.org/
  - MooTools
    - Tons of JavaScript utility objects
    - http://mootools.net/

# Further Literature

Resources

- https://developer.mozilla.org/de/learn/javascript
  - Tutorials for beginners and advanced
  - German

- http://www.w3schools.com/js/default.asp
  - Tutorials for beginners
  - English

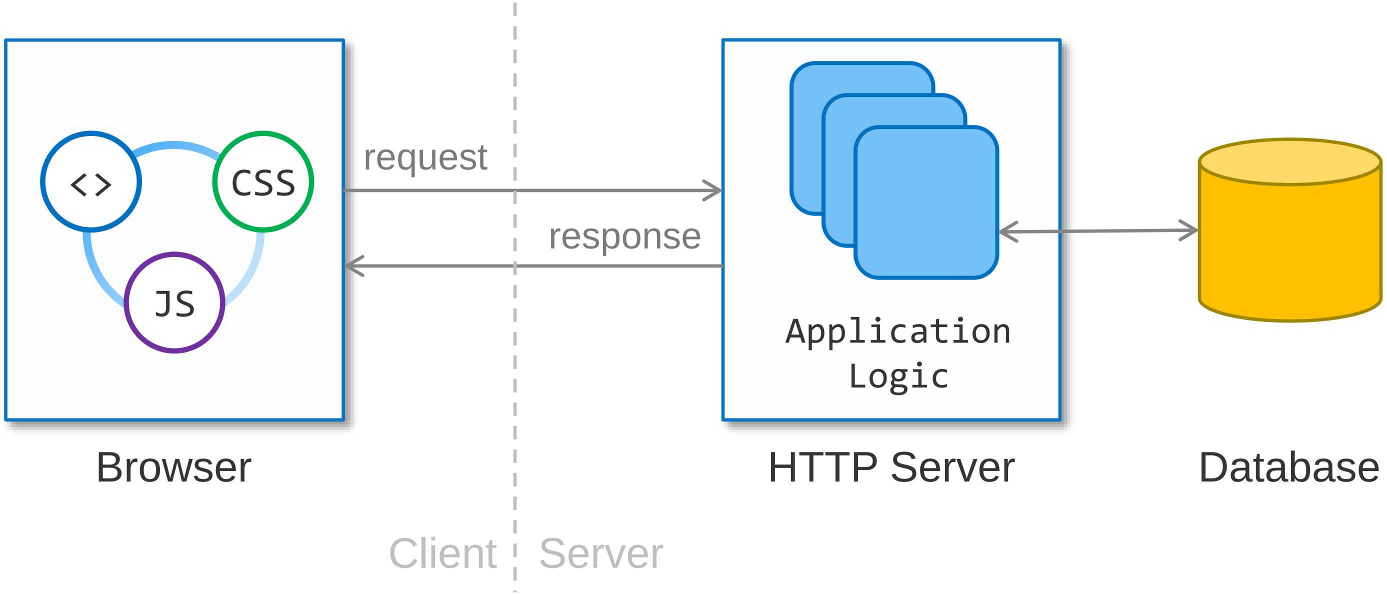
- http://learn.jquery.com/
  - jQuery documentation

- Many more available on the web
  - Google for „javascript tutorial“

# Basic Technologies

Overview

- Client Side
  - (X)HTML for structure and content ✓
  - CSS for layout and style ✓
  - JavaScript for client-side functionality ✓



# Thank you for your attention

Questions?

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
 <head>
   <title>Thank you for your attention</title>
 </head>
 <body>
    <h1>Thank you for your attention!</h1>
    <p>
      I want to thank you for your attention and hope that you
      enjoyed the talk.
    </p>
    <p>
      Are there any questions?
    </p>
 </body>
</html>
```