



Web Engineering & Design 1

HTML

Advanced

Tobias Blaser



HSR

HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Content

- Tables
- Images, Figure
- Multimedia
- Active content
- Forms
- Accessibility

Learning goals

You are able to...

- ...explain in which case which HTML tag (list, table, img, figure, video, ...) should be used.
- ...validate, structure and markup table or matrix data.
- ...embed images and SVG in HTML documents and able to use figures.
- ...explain how multimedia should be embeded and you are able to embed multimedia in HTML documents (codecs, multiple sources, subtitles, fallback, ...).

You are able to...

- ...explain EME and discuss gains and problems of this technology.
- ...embed active content and know which tags should be used instead of deprecated plugin contents.
- ...choose the proper input/form tag for a certain case.
- ...create forms like contact forms or request forms and enforce validation for certain fields.
- ...provide autocomplete for form fields.
- ...use white space the correct way and understand how it is interpreted by the browser.

You are able to...

- ...list examples of accessibility restrictions.
- ...create well-defined and accessible web pages using WAI-ARIA.
- ...list Accessibility / Usability / SEO tipps.
- ...explain why tags like font, strike, center, u, s, ... have been removed in HTML5.

Tables (table)

```
&& true false  
true true false  
false false false
```

[examples/tables.html](#)

```
<table>  
    <tr>  
        <th>&&</th>  
        <th>true</th>  
        <th>false</th>  
    </tr>  
    <tr>  
        <th>true</th>  
        <td>true</td>  
        <td>false</td>  
    </tr>  
    <tr>  
        <th>false</th>  
        <td>false</td>  
        <td>false</td>  
    </tr>  
</table>
```

- Used to markup tabular (matrix) data
- **Must not** be used for layout!

logical combinations

	xor	and
	true	false
true	false	true
false	true	false

[examples/tables.html](#)

```
<table>
    <caption>logical combinations</caption>
    <colgroup>
        <col span="1" class="pane" />
        <col span="4" />
    </colgroup>
    <thead>
        <tr>
            <th></th><th colspan="2">xor</th><th colspan="2">and</th>
        </tr>
        <tr>
            <th></th>
            <th>true</th><th>false</th><th>true</th><th>false</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <th>true</th>
            <td>false</td><td>true</td><td>true</td><td rowspan="2">false</td>
        </tr>
        <tr>...</tr>
    </tbody>
</table>
```

jsfiddle.net/Lpo3Lmv6/

thead, tbody, tfoot

Table header, body and footer (optional)

tr

Table row (may not be empty!)

th / td

Table **header cell** (head, body) / **data cell** (body)

```
<th abbr="Table cell summary">...</th>
<th scope="colgroup" colspan="3">...</th>
```

```
<tr><th id="name">Name</th>...</tr>
...
<tr><td headers="name">Max</td>...</tr>
```

```
<!-- Spread cell over x columns/rows -->
<td colspan="2">...</td>
<td rowspan="3">...</td>
```

colgroup

```
<colgroup>
    <col span="1" class="pane" /><col span="4" class="content" />
</colgroup>
```

Images (img)



Logo HSR A3



[examples/images.html](#)
[jsfiddle.net/2eaethgm/1](#)

```




```

- **width / height:** Image source size
 - content information, not visual size!
- **srcset:** multiple images (mobile/desktop)
 - **320w, 640w:** Image size → requires "sizes"
- **alt:** Alternative text / short description

New picture tag ([picture](#))



[examples/picture.html](#)

```
<picture>
  <source type="image/webp" srcset="Titelbild_Abend-320.webp" />
  <source type="image/jpeg" srcset="Titelbild_Abend-320.jpg" />

  <!-- Fallback -->
  
</picture>

<picture>
  <source media="(min-width: 1024px)" srcset="Titelbild...jpg" />
  <source media="(min-width: 480px)"
    srcset="Titelbild_Abend-640.jpg, Titelbild_Abend.jpg 2x" />

  
</picture>
```

picture vs. img

- **type:** Browser will choose first supported format
- **media:** Media queries for responsive images (breakpoints)
→ see later
- For normal cases: **use img instead:**
blog.cloudfour.com/dont-use-picture-most-of-the-time/
 - *img:* Browser may **handle images smart**
 - *picture:* Browser needs to follow the rules (no smart enhancements)
- Bad browser support (only FF, GC, IE-Edge)

Image formats

Type		Size / quality	Trans-parency	Anim-a-tions	Usage	Browser Support
jpg	Raster	++	-	-	photographs	✓
png	Raster	+	✓	-	clip arts, logos	✓
gif	Raster	+	✓	✓	clip arts, logos, animations	✓
webp	Raster	+++	✓	✓	everything	GC, AB
svg	Vector		✓	✓	clip arts, logos, animations	✓

GC: Google Chrome, AB: Android browser

🖼 Embedded images (Data-urls)

```

```

SVG - Scalable vector graphics



```
<?xml version="1.0" encoding="UTF-8"?>
<svg version="1.2" width="100mm" height="100mm"
  viewBox="0 0 10000 10000" xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink" xml:space="preserve">
  <g>
    <path fill="rgb(0,96,160)" stroke="none" d="M 2460,3920
      L 999,3920 999,999 3920,999 3920,3920 2460,3920 z"/>
    <path fill="rgb(180,180,180)" stroke="none" d="M 7614,3902
      L 6175,4156 5668,1279 8545,772 9052,3648 7614,3902 z"/>
    <path fill="rgb(0,96,160)" stroke="none" d="M 2460,8999
      L 999,8999 999,6078 3920,6078 3920,8999 2460,8999 z"/>
    <path fill="rgb(0,96,160)" stroke="none" d="M 7539,8999
      L 6078,8999 6078,6078 8999,6078 8999,8999 7539,8999 z"/>
  </g>
</svg>
```

jsfiddle.net/qguhrqu6/

- Logos
- Animations
- Linked areas for images → instead of **image maps**
- Cartoons
- Interactive images
- Clip-arts
- Banners

Embed SVG

	native	object	iframe	img	CSS
	<pre><svg width... > <g> <path.../> </g> </svg></pre>	<pre><object data="logo.svg" type="image /svg+xml"> </object></pre>	<pre><iframe src="logo.svg"> </iframe></pre>	<pre></pre>	<pre><div style=" background: url('logo.svg'); "> </div></pre>
display	✓	✓	✓	✓	✓
animation	✓	✓	✓	✓	✓
script	✓	✓	✓	-	-
interactivity	✓	✓	✓	-	-
linking	✓	✓	✓	-	-

```
<!-- Data url -->

<!-- data:image/svg+xml;utf8 working, but not valid :-(

  --> use base64 encoded svg instead: -->

```

Figure



The University of Rapperswil

to string & to value functions:

```
function(item) { return item.toString(); }
function(item) { return item.toValue(); }
```

[examples/figures.html](#)

```
<figure id="university">
    
    <figcaption>The University of Rapperswil</figcaption>
</figure>
<figure id="toStringToValueFunctions">
    <figcaption>to string & to value functions:</figcaption>
    <code>function(item) { return item.toString(); }</code>
    <code>function(item) { return item.toValue(); }</code>
</figure>
```

jsfiddle.net/2eaethgm/1/

Multimedia

Audio ([audio](#))



This file is licensed under the [Creative Commons Attribution-Share Alike 4.0 International](#) license. Created by **ZappaMidiPF** shared on <https://commons.wikimedia.org/wiki/>

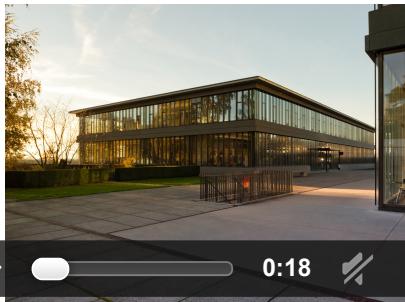
[examples/audio.html](#)

```
<audio src=".//Piano_acustico_acorde_Am.ogg" controls></audio>

<audio controls autoplay loop muted preload="metadata">
    <source src=".//Piano_acustico_acorde_Am.ogg"
        type="audio/ogg" />
    <source src=".//Piano_acustico_acorde_Am.opus"
        type="audio/opus" />

    <a href=".//Piano_acustico_acorde_Am.ogg">
        Download example audio</a>
</audio>
```

Video ([video](#))



Google Street View <https://www.google.ch/>

[examples/video.html](#)

```
<video src=".example.mp4" controls poster=".Titelbild_Abend.jpg">
    <a href=".example.mp4">Download example video</a>
</video>

<video poster=".Titelbild_Abend.jpg" controls autoplay loop muted>
    <source type="video/mp4" src=".example.mp4" />
    <source type="video/ogg" src=".example.webm" />

    <a href=".example.mp4">Download example video</a>
</video>
```

jsfiddle.net/fww243Lx/4/ → jsfiddle.net/fww243Lx/5/

Video subtitles (track / WebVTT)



```
// example.en.vtt
WEBVTT

00:00.000 --> 00:09.000
<v HSR>On the left: Castle Rapperswil

00:10.000 --> 00:16.000
<v HSR>On the right: Rapperswil Station
```

examples/video.html
jsfiddle.net/fww243Lx/5/ → jsfiddle.net/fww243Lx/7/

```
<video controls autoplay loop src="./example.mp4">
  <track src="./example.en.vtt" kind="subtitles" srclang="en"
        label="English" default />
  <track src="./example.de.vtt" kind="subtitles" srclang="de"
        label="Deutsch" />

  <track src="./example.desc.en.vtt" kind="descriptions"
        srclang="en" label="English description" />
</video>
```

- Subtitles / Multilingual support
- Captions for visually impaired people

Video codecs

Video	Audio	Container	Desktop	Mobile
Theora/VP9	Vorbis/Opus	Ogg/WebM	FF/GC	FF
VP8	Vorbis	WebM	FF/GC/~IE9+	FF/AB
H.264	MP3/AAC	MP4	✓	✓

FF: Firefox, GC: Google Chrome, AS: Apple Safari, IE: Internet Explorer, AB: Android browser

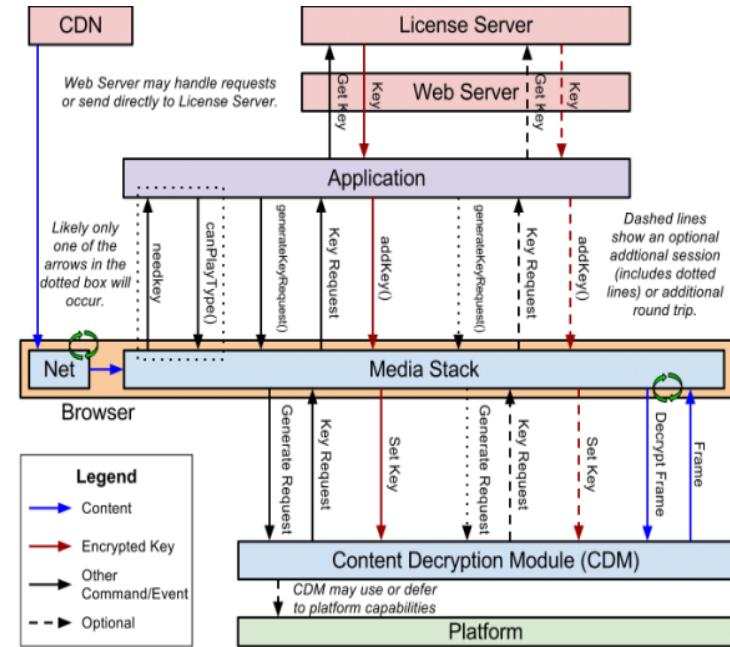
Audio codecs

Audio	Container	Desktop	Mobile
Vorbis/Opus	WebM/Ogg	FF/GC	FF
PCM	Wave	✓ (IE-Edge+)	FF/GC/AS/AB
MP3	MP3	✓	✓
AAC	AAC/MP3	✓	✓

MDN: developer.mozilla.org/en-US/docs/Web/HTML/Supported_media_formats

Encrypted Media Extensions (EME)

- Plugin Interface to play DRM protected media
- CDM: Content Decryption Module
 - playback of encrypted media
 - Vendor specific plugins
 - Closed source
- HTMLMediaElement extension (optional)
- Future: Danger of expansion to other elements



Source: golem.de/news/encrypted-media-extensions-drm-fuer-html5-1202-89953.html

Active content

Frames (`iframe`)



```
<iframe src="./index.html" name="HTML Slides overview"></iframe>
```

- **sandbox:** Security restrictions (e.g. "allow-scripts")
- **seamless:** Not supported yet
- **srcdoc:** Additional information about the source

Object & Embed (object, embed)



[examples/active-content.html](#)

```
<object data="./Selfhtml-active-content.pdf" type="application/pdf">
  <param name="src" value="./Selfhtml-active-content.pdf" />

  <!-- Fallback -->
  <a href="./Selfhtml-active-content.pdf">PDF herunterladen</a>
</object>
```

```
<embed width="600" height="400" data="animation.swf"
  type="application/x-shockwave-flash" src="./animation.swf">
```

object and *embed* are used to embed:



→ Use **video, SVG or Canvas** instead!

object behaviour

Browser supported content



Like *img*

Web content



Like *iframe*

Other content



Like *embed*

→ Use **object** instead of **embed**

Forms (form)

Male Female

Name

Phone

Email

Contact me by email

Customer number

Type of message

Urgency of message

Message

[examples/forms.html](#)

jsfiddle.net/ghdesgt4/4/ → jsfiddle.net/ghdesgt4/5/

```
<form action=".//sendMessage" method="post">
    <label>Male
        <input type="radio" name="gender" value="male" checked />
    </label>
    <label>Female
        <input type="radio" name="gender" value="female" />
    </label>
```

Name	<input type="text"/>
Phone	<input type="text"/>
Email	<input type="text"/>
Contact me by email	<input checked="" type="checkbox"/>
Customer number	<input type="text" value="CH1234"/>
Type of message	<input type="text" value="Feedback"/> ▼

[examples/forms.html](#)

```
<label for="name">Name</label>
<input type="text" name="name" id="name" />

<!-- phone field (numbers like (+41)12345678) -->
<label for="phone">Phone</label>
<input type="tel" name="phone-number" id="phone" />

<!-- email address field -->
<label for="mail">Email</label>
<input type="email" name="e-mail" id="mail" />
```

```
<!-- contact me checkbox -->
<label for="reply">Contact me by email</label>
<input type="checkbox" name="reply" id="reply" />

<!-- custom text input + placeholder -->
<label for="customer-number">Customer number</label>
<input type="text" ... placeholder="CH1234" />
```

Type of message **Feedback ▾**

Urgency of message 

Message

Send request **Reset form**

[examples/forms.html](#)

```
<label for="messageType">Type of message</label>
<select name="message-type" id="messageType" size="1">
    <option value="feedback" selected>Feedback</option>
    <option value="question">Question</option>
</select>

<!-- urgency slider min:1, max:5 --&gt;
&lt;label for="messageUrgency"&gt;Urgency of message&lt;/label&gt;
&lt;input type="range" ... min="1" max="5" step="1" value="3" /&gt;</pre>
```

```
<!-- multiline text field -->
<label for="message">Message</label>
<textarea name="message" id="message"></textarea>

<!-- send and reset buttons -->
<button type="submit">Send request</button>
<button type="reset">Reset form</button>
</form>
```

Form properties

- **action:** Target address
- **method:** POST (Request body) or GET (URL params)
- **accept-charset, enctype:** Data character encoding
- **autocomplete:** *on/off*
- **novalidate:** disables validation

Field properties

- **name:** Data label
- **placeholder:** Text visible in empty field
- **value, checked, selected:** Predefined element value
- **size, multiple, datalist**
- **readonly:** disable editing → use *output* for not-input content
(e.g. a calculated price)
- **min, max, step:** Properties for number and range input
- **multiple:** Select multiple options
- **autofocus:** Place cursor in this field
- **form:** Link fields outside form
- **autocomplete="off":** Disable autocompletion

<input type="?" />

Text input tags

- text
- search
- password
- tel
- url
- email

Selection input tags

- radio
- checkbox

Number input tags

- number
- range

Time input tags

- date
- datetime
- datetime-local
- week
- month
- time

Other input tags

- hidden
- file
- color

A screenshot of an iPhone X displaying a mobile form. At the top, there are fields for 'Phone' and 'Email'. Below these are checkboxes for 'Contact me by email' and 'Customer number' (with value 'CH1234'). A dropdown menu for 'Type of message' is open, showing 'Feedback' as the selected option. Below the dropdown is a text input field for 'Message'. At the bottom of the screen are buttons for 'Send request' and 'Reset form', along with navigation icons.

A screenshot of an iPhone X displaying a mobile form, identical to the one above but with a different keyboard layout. The text input field for 'Email' is at the top. Below it are checkboxes for 'Contact me by email' and 'Customer number' (with value 'CH1234'). A dropdown menu for 'Type of message' is open, showing 'Feedback' as the selected option. Below the dropdown is a text input field for 'Message'. At the bottom of the screen are buttons for 'Send request' and 'Reset form', along with navigation icons.

A screenshot of an iPhone X displaying a mobile form. At the top, there are fields for 'Phone' and 'Email'. Below these are checkboxes for 'Contact me by email' and 'Customer number' (with value 'CH1234'). A dropdown menu for 'Type of message' is open, showing 'Feedback' as the selected option. Below the dropdown is a text input field for 'Message'. At the bottom of the screen are buttons for 'Send request' and 'Reset form', along with navigation icons. A date picker overlay titled 'Select day' is displayed, showing a grid of months and years. The current selection is 'February 09 2016'.

A screenshot of an iPhone X displaying a mobile form. At the top, there are fields for 'Phone' and 'Email'. Below these are checkboxes for 'Contact me by email' and 'Customer number' (with value 'CH1234'). A dropdown menu for 'Type of message' is open, showing 'Feedback' as the selected option. Below the dropdown is a text input field for 'Message'. At the bottom of the screen are buttons for 'Send request' and 'Reset form', along with navigation icons. A date picker overlay titled 'Select' is displayed, showing a grid of months and years. The current selection is 'February 09 2016'.

A screenshot of a desktop browser window showing a form with various input types. At the top, there are radio buttons for 'Male' and 'Female'. Below them are text input fields for 'Name', 'Phone', and 'Email'. There is a checkbox for 'Contact me by email' and a text input field for 'Customer number' (with value 'CH1234'). A dropdown menu for 'Birthdate' is open, showing '2014' as the selected year. A text input field for 'Preferred color' has a black bar placeholder. A dropdown menu for 'Type of message' is open, showing 'Feedback' as the selected option. Below the form are buttons for 'Send request' and 'Reset Form'. A color picker dialog box is overlaid on the bottom right, titled 'Choose a color'. It shows a color wheel with a red-to-pink gradient, and the hex code '#000000' is displayed in the 'Color name:' field. Buttons for 'Cancel' and 'OK' are at the bottom right of the dialog.

- Input/field dependent keyboard/picker → + usability
- Test form elements in your browser: nativeformelements.com

Date	<input type="text" value="mm/dd/yyyy, --:-- --"/>
Age	<input type="text"/>
Image	<input type="file" value="Choose Files"/> No file chosen
Color	<input type="color" value="#ff0000"/>
Password	<input type="password" value="....."/>

Date

February 2016 ▾

Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	1	2	3	4	5

examples/forms.html
jsfiddle.net/jyeqLLw6/ → jsfiddle.net/jyeqLLw6/

```
<form enctype="multipart/form-data" method="post" ... >
  <label for="date">Date</label>
  <input type="datetime-local" name="date" id="date" />

  <label for="age">Age</label>
  <input type="number" name="age" id="age" min="0" max="120" />

  <label for="images">Image</label>
  <input type="file" name="images" ... accept="image/*" multiple />

  <label for="colors">Color</label>
  <input type="color" name="colors" id="colors" value="#ff0000" />

  <label for="password">Password</label>
  <input type="password" name="password" id="password" ... />

  <input type="hidden" name="hidden-values" value="checksum" />
</form>
```

Fieldsets

Options

- Receive newsletter
- Contact me by email

```
<fieldset id="optionsFieldSet">
    <legend>Options</legend>
    <!-- ... -->
</fieldset>
```

[examples/forms.html](#)

Autocompletion

Corporation

[examples/advancedForms.html](#)

```
<datalist id="vendors">
    <option value="Microsoft">Microsoft</option>
    <option value="Apple">Apple</option>
    <option value="Sun Microsystems">Sun Microsystems</option>
</datalist>
<label for="vendor">Corporation</label>
<input type="text" name="vendor" id="vendor" list="vendors" />
```

Validation (form validation)

■ Type enforced validation

■ *type="tel"*

Phone by email

Please match the requested format.

```
<input type="tel" ... />
```

■ *type="email"*

Email by email

Please enter an email address.

```
<input type="email" ... />
```

■ Validation attributes

■ *required*

Name

Phone by email

Please fill out this field.

```
<input type="text" ... required />
```

■ *pattern*

Customer number by email

Type of message Please match the requested format.

```
<input type="text" ... pattern="([A-Z][A-Z])?[0-9]*" />
```

[examples/forms.html](#) | [jsfiddle.net/ghdesgt4/5/](#) [jsfiddle.net/jyeqLLw6](#)

- Form will not be sendable until **all required fields** match
- Disable validation:

```
<form novalidate>...</form>
```

- Error messages are **browser dependent**
- *email, text, tel, date, ...* → enforce **specific formats**
- *patter* → enforce **custom format**
- Field relations are not possible yet without JavaScript
- Custom validation → JavaScript
- Form may be bypassed. Check every form on receiver side!

White space

«*Characters creating one empty space*»

- " " Different blank spaces
- \t, \r, \n Tabulator, Return and Line Feed

Appearance

- White spaces and breaks will **collide**
- Except `<pre>` and `<code>`
- Enforce break: `
`
- Non breaking space: `&nbsp`
- White space appearance is **controllable by CSS**

Accessibility

Accessibility Restriction Examples

- **User Disability**
 - sight, hearing, cognitive speed, color blindness, ...
- **Environment**
 - noise, disturbing light, display distance, ...
- **Temporary** → network speed, interceptions, ...
- **Device**
 - no speakers, small screen, no haptic feedback, ...
- **Equipment**
 - gloves, earplugs, sunglasses, missing glasses, ...

WAI-ARIA

Web Accessibility Initiative - Accessible Rich Internet Applications

- Use a **semantical HTML5 structure**
- Use **landmark *roles***

```
<main role="main">
    <article>
        ...
</main>
```

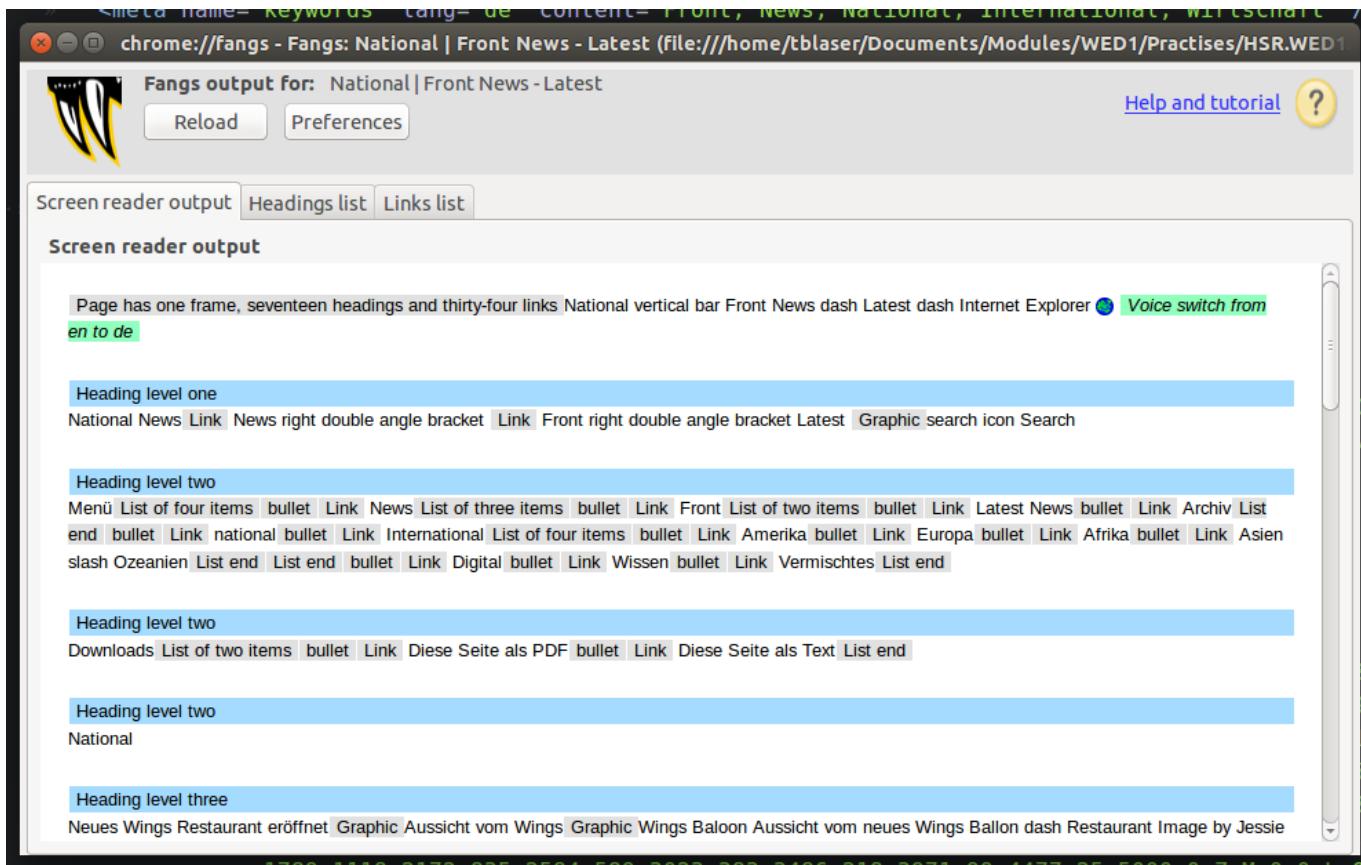
```
<aside role="complimentary">
    <h2>Related links</h2>
    ...
</aside>
```

```
<footer role="contentinfo">
    <small>Rights reserved by...</small>
    ...
</footer>
```

Roles (**role**)

- *application*: Standalone web app / multimedia
- *banner*: Site specific content (page title, logo)
- *complementary*: Additional content, may be related to main content
- *contentinfo*: Author, Copyright notes, small marks, ...
- *form*
- *main*: Main content or content linking to main content
- *navigation*
- *search*: Search function
- SVG: role="img" or role="application"

- *aria-label*: Label form fields without label
- Which tags need an ARIA-attribute:
w3.org/TR/html-aria/#docconformance



The screenshot shows the Fangs Screen Reader Emulator interface within a Firefox browser window. The title bar reads "chrome://fangs - Fangs: National | Front News - Latest (file:///home/tblaser/Documents/Modules/WED1/Practises/HSR.WED1)". The main area is titled "Screen reader output" and displays a hierarchical tree of elements. The tree is organized by heading levels:

- Heading level one**: National News Link News right double angle bracket Link Front right double angle bracket Latest Graphic search icon Search
- Heading level two**: Menü List of four items bullet Link News List of three items bullet Link Front List of two items bullet Link Latest News bullet Link Archiv List end bullet Link national bullet Link International List of four items bullet Link Amerika bullet Link Europa bullet Link Afrika bullet Link Asien slash Ozeanien List end List end bullet Link Digital bullet Link Wissen bullet Link Vermischtes List end
- Heading level two**: Downloads List of two items bullet Link Diese Seite als PDF bullet Link Diese Seite als Text List end
- Heading level two**: National
- Heading level three**: Neues Wings Restaurant eröffnet Graphic Aussicht vom Wings Graphic Wings Balloon Aussicht vom neues Wings Ballon dash Restaurant Image by Jessie

Fangs Screen Reader Emulator (Firefox)
addons.mozilla.org/en-US/firefox/addon/fangs-screen-reader-emulator/?src=search

Accessibility / Usability / SEO tipps

- Valid html code, no element misuse
- Speaking title, **Good content, content descriptions**
 - *alt, figcaption, video subtitles, form labels ...*
- plain text/html/svg> instead of PDF/image/active content
- Site search placed easy findable in header
- Flat content structure → **less clicks**
- Easy accessible navigation structures
- **Accessible on all devices**
 - keyboard, mouse, cursor (TV), touch, printers, ...
- Low network data usage

HTML5: Removed

Removed tags

- acronym ■ applet ■ dir ■ frame ■ frameset
- noframes ■ isindex ■ basefont ■ center ■ big
- font ■ strike ■ s ■ tt ■ u

Removed attributes

- align ■ alink ■ background
- bgcolor ■ border ■ cellpadding
- cellspacing ■ clear ■ compact
- marginheight ■ marginwidth ■ nohref
- noshade ■ valign ■ ...



Web Engineering & Design 1

HTML

Advanced

Tobias Blaser



HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Self Check

1. In which case should you use a table and in which not?
2. Tables: What do *scope*, *abbr*, *colspan* and *rowspan*?
3. Images: How and for what do you use *srcset*?
4. Img tag: What is specified by the width and height attribute?
5. Why should you use *img* instead of *picture*? When may you use *picture*?

6. 3 Advantages of *SVG*

7. In which case do you embed *SVG* native, as *object*,
iframe or *img*?

8. How do you embed images by data-urls?

9. When should you use *figure*?

10. How do you use multiple video formats?
11. What is *EME* and where are the problems?
12. Why should you use *object* instead of *embed*?
13. What's the difference between the following lines?

```
<label>Label<input type="text" ... /></label>
```

```
<label>Label</label><input type="text" ... />
```

14. Why should you use button instead of input?

```
<button type="submit">
```

```
<input type="submit">
```

15. Why should you use different *input* types and not *text* for all?

16. Does *input-password* protect the password?

17. Why should the receiver of a form always validate the data?

18. What happens to multiple white spaces, tabs and line breaks?

19. Why should you add WAI-ARIA attributes?
20. Why should you not use the *strike*-tag?

Solutions

1. Use for: matrix data, not use for: layouts, forms, ...
2. scope: declares the title scope (row, col, colgroup),
abbr: Help for blind people, row-/colspan: Spread cell over
multiple cols/rows
3. Usage see [S5](#). Used for multiple images for different
screen resolutions and densities.
4. The size of the source image (NOT the displayed size!)
5. Img: Browser is able to do smart enhancements. Use for
different image formats not supported by all browsers

6. Resolution independent, often smaller file size,
versionable, human editable, animations possible,
styleable

7. ``

8. Native: For small images (prevents HTTP request),
object: if interactivity is needed, iframe: use object
instead, img: normal case including animations
9. Everytime you need a container around one or more
img/code blocks/videos/... or if you would like to group
such components and label by a caption

10. See [S6/1](#)
11. A DRM plugin interface extension for the video element. The DRM systems will need a vendor specific plugin. It's closed source (not accessible for impaired people).
12. Object will react as img, iframe or embed, depending on the content type.
13. The second variant needs a for-relation to link the label and the input.

14. Input will send the value of the button. The button is an interactive element only.
15. Devices may enhance the usability of different input types (e.g. date chooser on mobiles).
16. No. The password is not visible, but it's sent plain text. There is now protection.
17. Form validation may be bypassed by sending a foreign request to the server. Form validation is a usability feature only.
18. They will collide

19. To help not HTML5-compatible screen readers to understand the structure of a page.
20. Visual appearance should not be marked by HTML. This is the reason, tags like *strike* are not allowed anymore in HTML5.



Web Engineering & Design 1

HTML

Advanced

Tobias Blaser



HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz