

November 29, 2006

Understandable Content and Controls

Greg Shin ()
LG Electronics
<http://gregshin.pe.kr/blog/>

Table of Contents

1. Make it understandable
2. Table
3. Semantic Markups & CSS
4. Forms

1. Make It Understandable

The third guideline of KWCAAG 1.0 covers understanding problems.

Guidelines

Checklists



1. Content must be **perceivable**.



2. Interface components in the content must be **operable**.



3. Content and controls must be **understandable**.

Accessible Table

Logical organization

Accessible Form

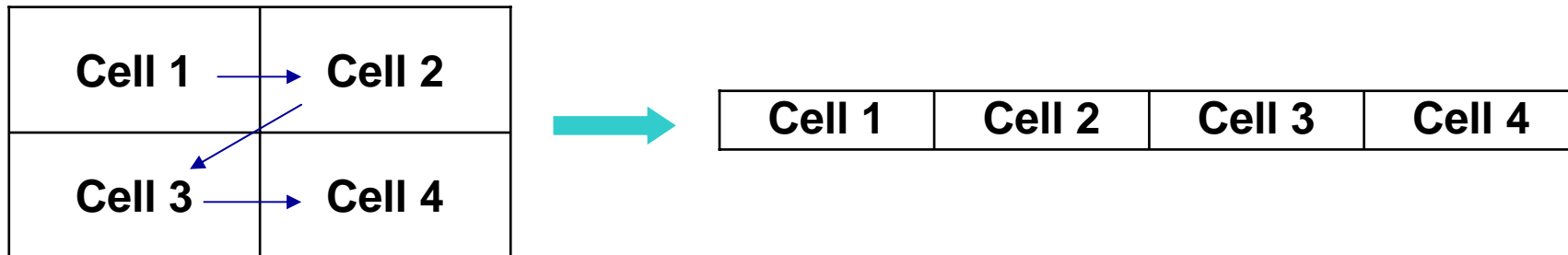


4. Content should be robust enough to work with current and future user agents (including assistive technologies).

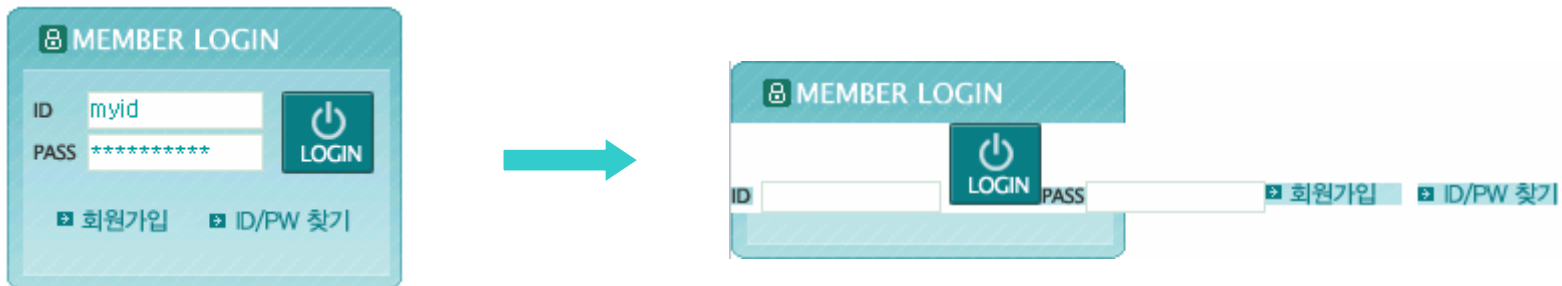
2. Table – Layout Table

Content must be understandable when linearized.

Linearization



Bad Example



Solutions

1. Interim Solution: Add `tabindex` properties.
2. Better Solution: Use Stylesheet for proper positioning.

2. Table – Data Table – Identifying a Table

Add appropriate table identification information.

- `<caption>` for table titles
- `summary` for table purpose

Vegetable Prices in Seoul and Busan (Won)

	Beans	Carrots	Tomatoes
Seoul			
Wholesale	100	120	100
Retail	200	300	250
Busan			
Wholesale	110	90	80
Retail	210	290	240

```
<table summary="Wholesale and
retail prices of vegetables in
Seoul and Busan. There are two
levels of row headings.">
<caption>Vegetable prices in
Seoul and Busan (Won)</caption>
```

2. Table – Data Table – Identifying rows and columns

Identify row and column headers.

- Using scope attribute

	Midterm Examination	Final Examination
John	10	20
Elisa	30	40

```
<table border="1" summary="Individual scores of midterm and final exams">
<caption>Individual Scores</caption>
<thead>
  <tr>
    <th></th>
    <th scope="col" abbr="midterm">Midterm Examamination</th>
    <th scope="col" abbr="final">Final Examination</th>
  </tr>
</thead>
<tbody>
  <tr>
    <th scope="row">John</th>
    <td>10</td>
    <td>20</td>
  </tr>
  <tr>
    <th scope="row">Elisa</th>
    <td>30</td>
    <td>40</td>
  </tr>
</tbody>
</table>
```

Identify row and column headers.

- Using `id` and `headers` attributes

	Midterm Examination	Final Examination
John	10	20
Elisa	30	40

```
<table border="1" summary="Individual scores of midterm and final exams">
<caption>Individual Scores</caption>
<thead>
  <tr>
    <th></th>
    <th id="midterm" abbr="midterm">Midterm Examination</th>
    <th id="final" abbr="final">Final Examination</th>
  </tr>
</thead>
<tbody>
  <tr>
    <th id="John">John</th>
    <td headers="midterm John">10</td>
    <td headers="final John">20</td>
  </tr>
  <tr>
    <th id="Elisa">Elisa</th>
    <td headers="midterm Elisa">30</td>
    <td headers="midterm Elisa">40</td>
  </tr>
</tbody>
</table>
```

3.1. Semantic Markups

Do not rely on presentational cues to deliver messages.

Your intentions	Bad implementation	Good implementation
Sometimes seeing is not all.	Sometimes seeing is not all.	<p>Sometimes seeing is not all.</p>
<ul style="list-style-type: none"> Firstly, you need to be proactive. 	* Firstly, you need to be proactive.	Firstly, you need to be proactive.
	 ...	{text-align: justify}
W O R L D	W O 	{letter-spacing: 1em}
Heading 1	Heading 1 	h1 {margin-bottom: 2em} <h1>Heading 1</h1>
Heading 2		
Empty cell	 	table {empty-cells: show}
I truly believe it.	truly truly	truly truly

3.2. CSS - Accessibility Benefits

How beneficial for ACCESSIBILITY the Style Sheet is!

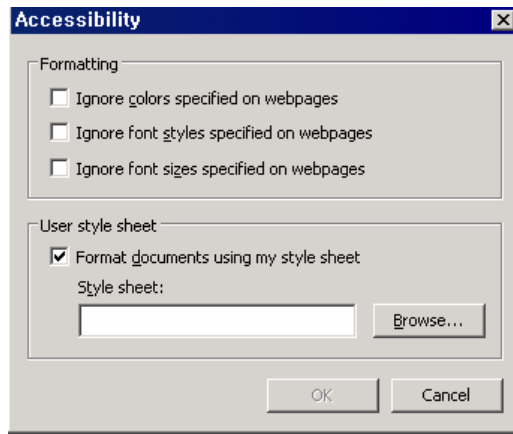
- CSS benefits accessibility primarily by **separating document structure from presentation**.
- CSS allows precise control over spacing, alignment and positioning. Authors can thereby **avoid "tag misuse"**.
- In addition to preventing element misuse, style sheets can help **reduce image misuse**.
- CSS provides **precise control** over font size, color, and style.
- CSS allows users to **override author styles**.
- CSS provides support for **automatically generated** numbers, markers, and other content that can help users stay oriented within a document.
- CSS supports **aural style sheets**.
- CSS provides more precise control over the **display of alternative content** than HTML alone.

Quoted from <http://www.w3.org/TR/CSS-access>

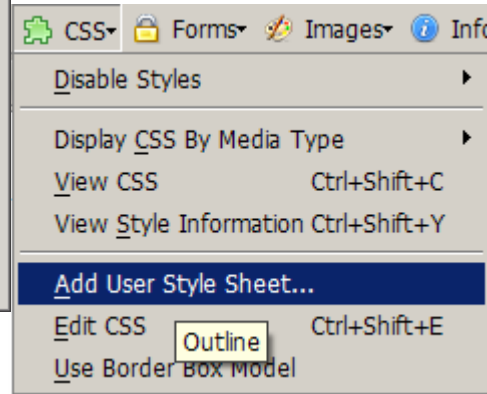
3.2. CSS – User Override of Styles

People with disabilities can use their own adaptable style while reading the page.

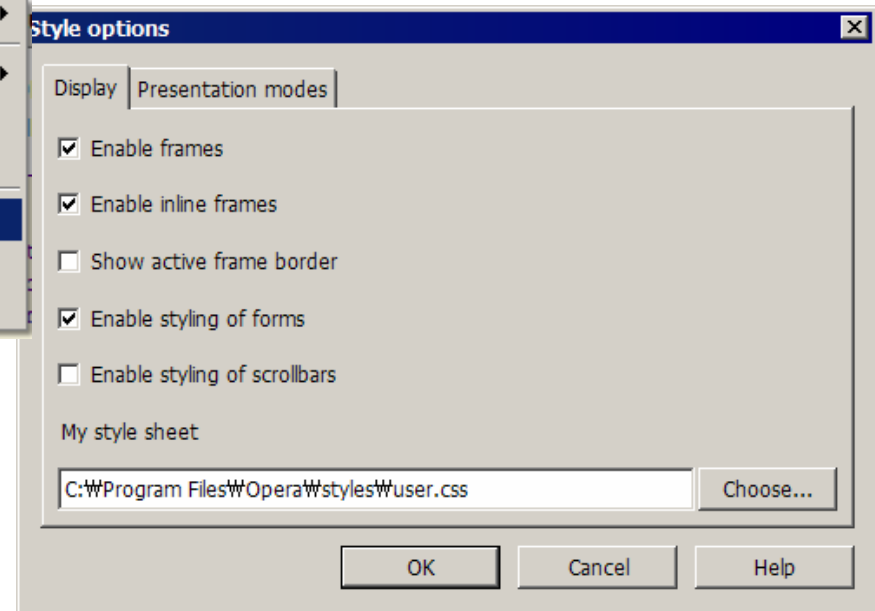
Internet Explorer 6 & 7



Firefox with
Web Developer Toolbar



Opera 9



Example of User Style Sheet

```
* {  
  color: black ! important ;  
  background: white ! Important;  
  font-size: 20pt ! Important;  
}  
:focus { outline: thick solid blue }  
:active { outline: thick solid red }
```

Refer to the site below for other browsers:
<http://www.howtocreate.co.uk/userStyle.html>

3.2. CSS – Content generation, Alternate representation

Generated content can serve as markers to help users navigate a document and stay oriented.

Showing accesskey after each element

```
* [accesskey]:after {
    content: attr(accesskey);
    text-decoration: underline;
    padding-left: 0.5em;
    font-size: smaller; }
```

Cueing header content

```
H1 {
    voice-family: paul;
    cue-before: url("ping.mp3")
}
```

Showing end of list

```
ul:after {
    content: End of Unordered List;
}
```

Showing special content when printing

```
@media print {
    acronym:after {
        content: "(" attr(title) ")"
    }
    a:after {
        content: " "attr(href);
        font-size: smaller; }
}
```

Use appropriate quotation marks

```
q:lang(en)
    { quotes: ' ' ' ' ' ' ' ' }
q:lang(ko)
    { quotes: " ⑆ " " ⑇ " " ' ⑆ ' ' ⑇ ' ' }
```

4. Forms – Form components

Forms consist of input or interaction elements.

Components	HTML markups	Accessibility Recommendations
Push buttons	<code><input type="button" ...></code>	
Graphical buttons	<code><input type="image" ...></code>	Add the "alt" attribute.
Generalized buttons	<code><button ...></code>	Not recommended
Text entry fields	<code><input type="text" ...></code> <code><input type="password" ...></code>	Match with <code><label></code> . Use "title" when necessary.
Text areas	<code><textarea ...></code>	Match with <code><label></code> .
Radio buttons	<code><input type="radio" ...></code>	Use <code><label></code> and "title". Share "name"s, use unique "id"s. Group using <code><fieldset></code> and <code><legend></code> .
Check boxes	<code><input type="checkboxes" ...></code>	Match with <code><label></code> .
List boxes	<code><select></code>	Do not use "onchange".
File	<code><input type="file" ...></code>	

4. Forms - Buttons

Make sure to add alt-text on image buttons.



A screenshot of a web form. It contains two input fields: one labeled '아이디' (ID) and one labeled '비밀번호' (Password). To the right of the password field is a blue button with a key icon and the text '로그인' (Login). Below these are two more buttons: '회원가입' (Sign Up) and 'ID/PW찾기' (Find ID/PW).

```
<input type="image" src="/images/main/MemberLoginBtn.gif" alt=" " >
```

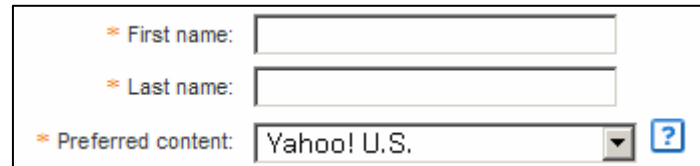


A screenshot of a search form. It features a text input field with the placeholder text '자료검색' (Search for materials) and a blue button with a magnifying glass icon and the text '검색' (Search).

```
<label for="main-search" > </label>
<input id="main-search" type="text" class="type-text" name="ss"
value=" " >
<input type="hidden" name="md" value="search" >
<input type="image" src="/images/main/SearchBtn.gif" alt=" " >
```

4. Forms – Text Input

- Use `<label>` for prompting text.



* First name:
* Last name:
* Preferred content: ?

```
<label for="lname">Last name:</label>
<input type="text" name=".ln" value="" size="30" maxlength="50"
id="lname">
<label for="intl">Preferred content:</label>
<select name=".intl" id="intl">
```

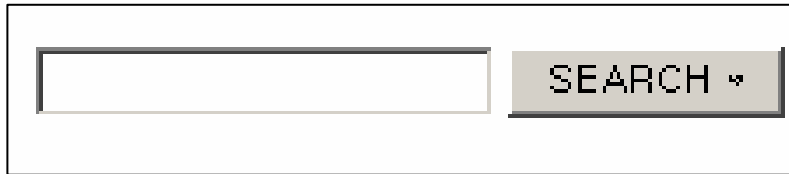
- 회원약관에 동의합니다. 동의함 동의안함

```
<fieldset>
  <legend> .</legend>
  <input id="chk1" name="stipulation" value="Y" type="radio">
  <label for="chk1"> .</label>
  <input id="chk2" name="stipulation" value="N" type="radio">
  <label for="chk2"> .</label>
</fieldset>
```

4. Forms – When Labels won't Work

Use the `title` attribute.

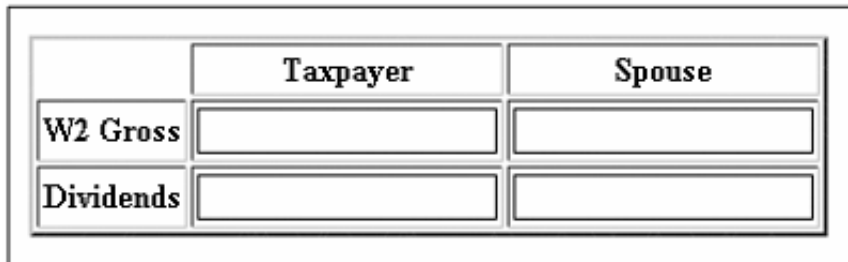
No on-screen text



A search form consisting of a text input field and a button labeled "SEARCH".

```
<input name="search" id="search" type="text" value="" title="Search" />
<input type="submit" value="Search &raquo;" />
```

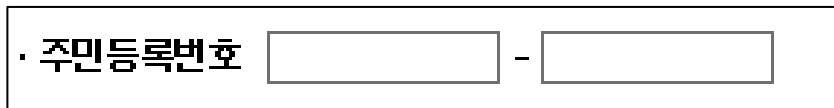
Forms in Tables



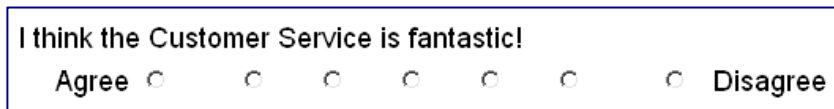
	Taxpayer	Spouse
W2 Gross	<input type="text"/>	<input type="text"/>
Dividends	<input type="text"/>	<input type="text"/>

```
<label for="ssid1">
</label>:
<input type="text" id="ssid1" title="6" /> -
<input type="text" id="ssid2" title="7" />
```

Shared prompting information



주민등록번호 -



I think the Customer Service is fantastic!
Agree Disagree

4. Forms – Fieldset & Legend

Use fieldset and legend elements for sets of controls.

Payment details

Credit card

American Express

Mastercard

Visa

Blockbuster Card

Card number

Expiry date

Submit my details


```
<fieldset>
<legend>Payment details</legend>
  <fieldset class="radio">
    <legend>Credit card </legend>
    <input id="card1a" type="radio" name="card1">
    <label for="card1a">American express</label>
    <input id="card1b" type="radio" name="card1">
    <label for="card1b">Mastercard</label>
    <input id="card1c" type="radio" name="card1">
    <label for="card1c">Visa</label>
    <input id="card1d" type="radio" name="card1">
    <label for="card1d">Blockbuster card</label>
  </fieldset>
  <label for="cardnum">Card number</label>
  <input id="cardnum" name="cardnum">
  <label for="expiry">Expiry date</label>
  <input id="expiry" name="expiry">
  <input class="submit" type="submit" value="submit
my details">
</fieldset>
```

4. Forms – Implicit Labeling

Do not use labels as containers.



```
<label>  
  username:  
  <input name="username" type="text">  
</label>
```



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

4. Forms - Validation

Use always server-side validation, use client-side validation as an auxiliary.

- Some people turns off the Javascript.
- Assistive technologies do not support every bite of Javascript.
- If a user skips the client-side only validation, harmful data can be submitted causing server disaster.

Validating Form

2 Errors in Submission

Please review the following

1. [Please enter your age](#)
2. [Please enter your email address](#)

Personal Details

Please enter your forename

Please enter your age

Please enter your email address

4. Forms - Help

Use new window or dynamic content for form help.

IMEI Code: ?

Explanation of IMEI Code

The International Mobile Equipment Identity (IMEI) number is a unique 15-digit code used to identify an individual GSM mobile telephone. The number can be found on most mobiles by typing in *#06#. If this combination doesn't work on your mobile phone, please call our support centre on +44 (0) 1252 xxxx xxx.

[Back to IMEI input field.](#)

```
<p>
  <label for="imei">IMEI Code: </label>
  <input type="text" id="imei" />
  <a href="imeihelp">What is IMEI?</a>
</p>
<div id="imeihelp">
  <h3>Explanation of IMEI</h3>
  <p>Help content here....</p>
  <p><a href="#fid">Back to IMEI input field</a>.</p>
</div>
```

*For full script and information, visit
<http://juicystudio.com/experiments/numbertransfer.php>*