Introducing web-accessibility

Making night and day difference as a developer.

smart4unity

Get inspired •

Who is Sergei Martens (11-3-1975)

What's his story?

- Oracle developer since 1998
- Started as classic developer, now APEX
- Special interest in UI / UX Bootstrap theme for APEX
- Apex UI developer SkillBuilders
- Founder of new SMART4Unity labels

New market, new labels

Smart4Unity members

smart4 apex®

Collaborating Oracle APEX Professionals

smart4 branding®

Marketing and Branding Software Products

The origin for this presentation

"The application needs to be 508 ADA compliant"

Quote customer:

Get inspired

508 ADA compliant •

“What The Heck?!“

What does web-accessibility means?

W3C says;

”People with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web.”

Visual impairment

Statistics:

- 2.3% of the U.S. population (0,1% blind)
  https://nfb.org/blindness-statistics
- 8.5% of men is color-blinded

Problems like:

- No mouse
- No keyboard support
- Auto start video or audio
- Dependence of color

Physical impairment

Statistics:

- 7.1% of the U.S. population
  http://www.cdc.gov/nchs/fastats/disability.htm
- 1 out of 5 has RSI problems (20%)

Problems like:

- No keyboard support
- No speak navigation

Hearing impairment

Statistics:

- 16.8 % of the U.S. population
  http://www.cdc.gov/nchs/fastats/disability.htm

Problems like:

- Audio and video with no subtitles
- Difficult text (English no native language)

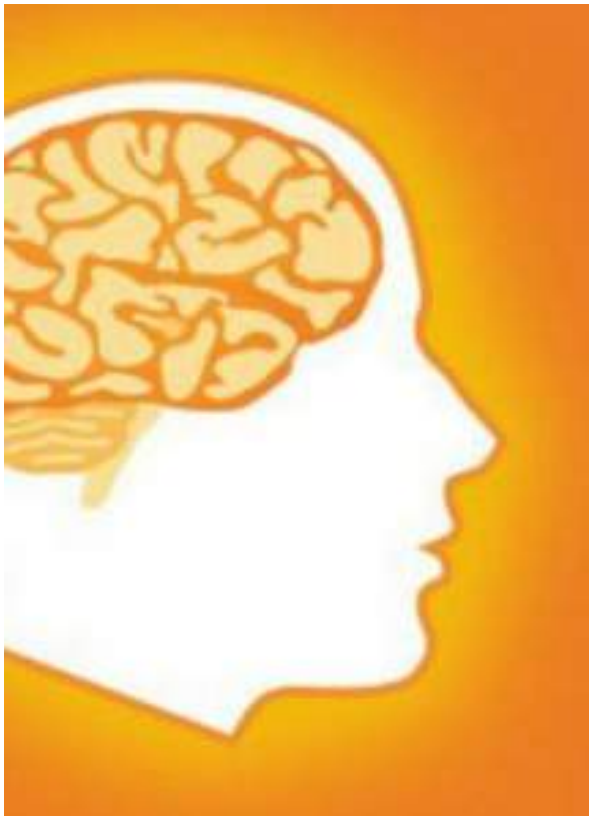
Cognitive impairment (Dyslexia, autism, etc)

Statistics:

- over 16 million people in the U.S.
  http://www.cdc.gov/aging/pdf/cognitive_impairment/cogimp_poilicy_final.pdf

Problems like:

- Inconsistency in page-navigation
- Unexpected changes on webpage
- No illustrative material
- Flickering, distracting banners
- Unclear headers and labels



Elderly

Statistics:

- 14.1% of the U.S. population (44.7 million)
- 22.1% of the U.S. population in 2040

http://www.aoa.acl.gov/aging_statistics/index.aspx

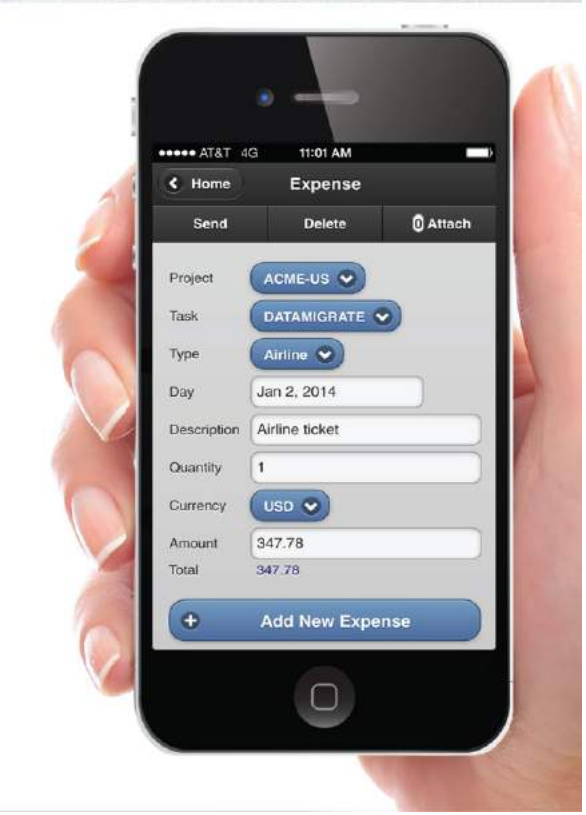
Problems like:

- Not grown up with a computer
- Increasing impairments

People in a limiting situation

Examples:

- Working on a mobile phone
- No daylight
- No updated version of plugin installed
- No Flash available
- Working on a beamer
- No mouse
- No audio
- Dutch no native language



508 ADA compliant •

“So accessibility still no issue for you?!“

Compliance

UN convention rights of persons with disabilities in 2006:

- Countries are to guarantee that persons with disabilities enjoy their inherent right to life on an equal basis with others.

Americans with Disabilities Act:

- a wide-ranging civil rights law that is intended to protect against discrimination based on disability.

United States; section 508:

- Agencies must give disabled employees and members of the public access to information that is comparable to access available to others.

Section 508

Set of standards how to create web-applications:

- Closely parallel with the WCAG 1.0 Priority 1 guidelines

Click here for website!

Proposal for update in 2015 to WCAG 2.0 AA:

- Set of standards from W3C

Click here for website!

508 ADA compliant •

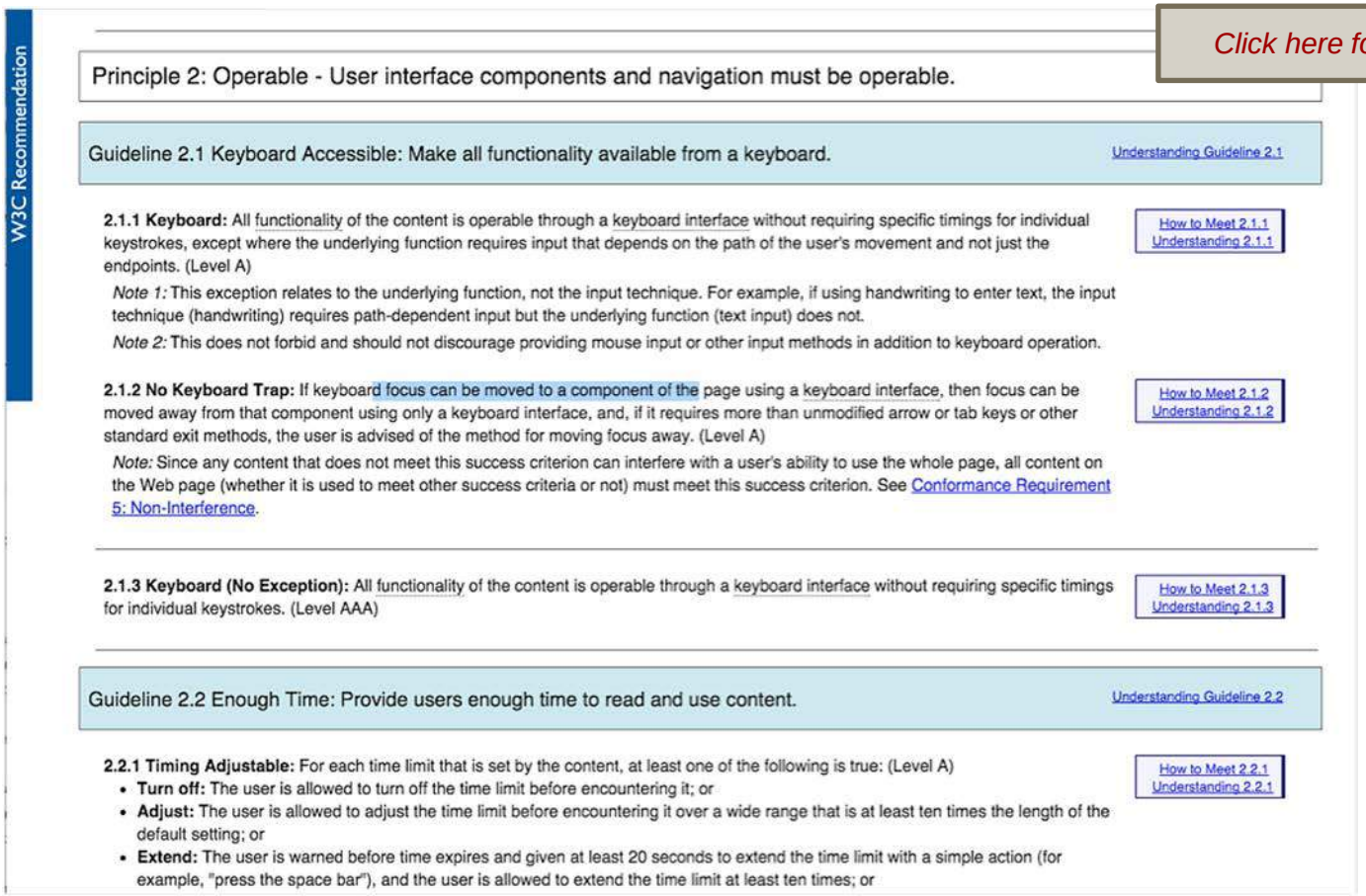
“So accessibility still no issue for you?!“

Web Content Accessibility Guidelines 2.0

Created by W3C (the “inventors” of the internet)

- Based on 4 principles:
  - Perceivable for everyone
  - Operable for everyone
  - Understandable for everyone
  - Robust for assistive technology
- The 4 principles are implemented by 12 guidelines
- Every guideline has one or more success criteria (61 in total)
- Every guideline is documented with:
  - Sufficient technics
  - Advisory technics
  - Common failures
- Three levels of compliance: A, AA and AAA

Hierarchy WCAG2.0



W3C Recommendation

Principle 2: Operable - User interface components and navigation must be operable.

Guideline 2.1 Keyboard Accessible: Make all functionality available from a keyboard. — Understanding Guideline 2.1

2.1.1 Keyboard: All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints. (Level A) — How to Meet 2.1.1 / Understanding 2.1.1

Note 1: This exception relates to the underlying function, not the input technique. For example, if using handwriting to enter text, the input technique (handwriting) requires path-dependent input but the underlying function (text input) does not.

Note 2: This does not forbid and should not discourage providing mouse input or other input methods in addition to keyboard operation.

2.1.2 No Keyboard Trap: If keyboard focus can be moved to a component of the page using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it requires more than unmodified arrow or tab keys or other standard exit methods, the user is advised of the method for moving focus away. (Level A) — How to Meet 2.1.2 / Understanding 2.1.2

Note: Since any content that does not meet this success criterion can interfere with a user's ability to use the whole page, all content on the Web page (whether it is used to meet other success criteria or not) must meet this success criterion. See Conformance Requirement 5: Non-Interference.

2.1.3 Keyboard (No Exception): All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes. (Level AAA) — How to Meet 2.1.3 / Understanding 2.1.3

Guideline 2.2 Enough Time: Provide users enough time to read and use content. — Understanding Guideline 2.2

2.2.1 Timing Adjustable: For each time limit that is set by the content, at least one of the following is true: (Level A) — How to Meet 2.2.1 / Understanding 2.2.1

- **Turn off:** The user is allowed to turn off the time limit before encountering it; or
- **Adjust:** The user is allowed to adjust the time limit before encountering it over a wide range that is at least ten times the length of the default setting; or
- **Extend:** The user is warned before time expires and given at least 20 seconds to extend the time limit with a simple action (for example, "press the space bar"), and the user is allowed to extend the time limit at least ten times; or

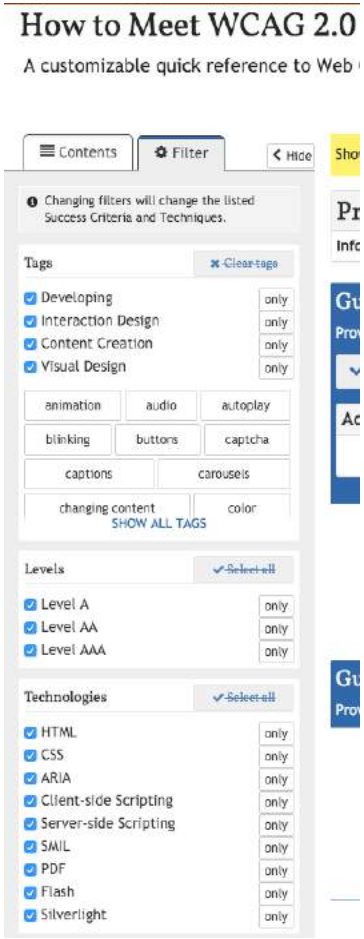
Click here for website!

Checking your compliance for WCAG 2.0

How to Meet WCAG 2.0

- A customizable quick reference to Web Content Accessibility Guidelines 2.0 requirements (success criteria) and techniques.

Click here for website!



How does Oracle meet these requirements?

"Oracle is committed to creating accessible technologies and products that enhance the overall workplace environment and contribute to the productivity of our employees, our customer, and our customers' customers"

Safra Catz, President and CFO

What does this mean for Oracle Application Express?

"Oracle Application Express 5.0, including the Universal Theme (UT) follow Oracle's corporate web accessibility guidelines, which themselves are based on WCAG 2.0 to 'A' and 'AA' Level and the applicable standards of Section 508."

Joel Kallman, productmanager Apex

508 ADA compliant •

"Wow… So the Apex-team has done an amazing job for us?!"

But there is a catch •

"How well we meet these standards is described in the VPAT for APEX 5.0."

Joel Kallman, product manager Apex

But there is a catch •

1. Current bugs mentioned in the VPAT (Apex-builder issues not included)
   - Inline date picker not usable with screen reader. Oracle bug 20696874
   - The modern calendar region type has accessibility issues. Oracle bug 20696874.
   - No declarative way of defining row headers. Oracle bug 14198644
   - Item types with an option to submit the page when the item's value changes. Oracle bug 20697122.
   - Errors displayed in Interactive Report display a red border around the field in error, with no text explaining what the error is. Oracle bug 20697217.
   - Potential duplicate ID values of 'LINK' when multiple interactive reports on a page. Oracle bug 20687795.
   - The 'Text Field with autocomplete' native item type fails to convey the autocomplete list to assistive technologies. Oracle bug 18918806.

But there is a catch •

1. Exluded in the VPAT:
   - Theme Roller & All sample applications

Break on the “theory”; a real world example

JAWS screenreader at work;

Form on a table with report in Universal theme

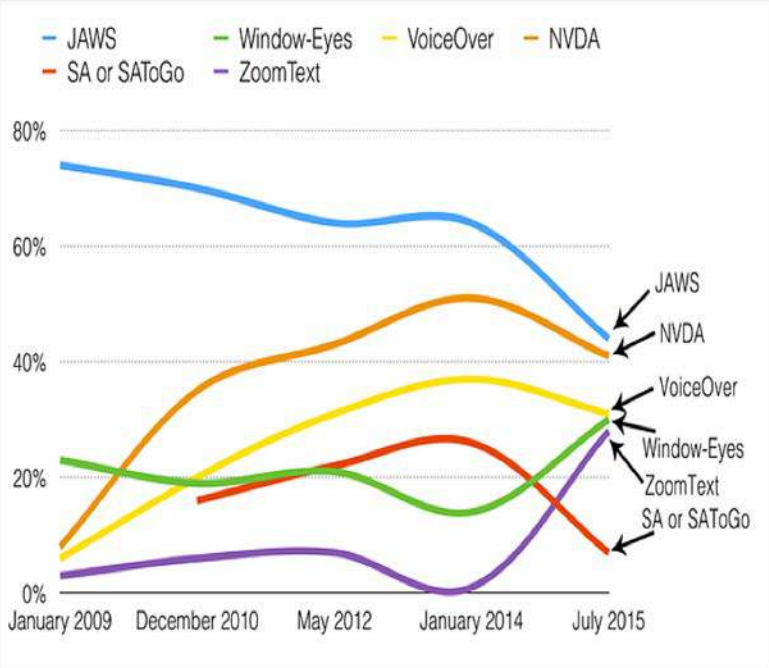
Our observation from the demo:

As long as we use Apex out-of-the box,
we are “relatively save”.

How does a screen reader work?

1. Interfaces with the browser accessibility API.
2. Technically does not read the screen.
3. Different support on different browsers for each screen reader
4. Over 300 short keys to help interpreting the page fort he user, like:
   a. List Headings = INSERT-F6
   b. Next heading = H
   c. Previous heading = SHIFT-H

   Click here for website!
5. Interfaces with the browser accessibility AP
6. “Beep” when cursor in a form element



Caution is needed as soon as we add our own HTML, CSS or Javascript

For example:

- Importing plug-ins
- Region of type PL/SQL dynamic region
- Region of type Static Content
- Dynamic actions with action “Execute Javascript Code”
- Region Header & Footer text
- Item Pre & Post text
- Link text report column
- Html expression in report column
- Much more…

Common mistakes • #1 No valid HTML

- Example:

```
<ul>
   <li><a>link text</li>
</ul>
```

- Problem:

  Screen reader potentially sends wrong or incomplete information to the user.

- Extra attention needed when:

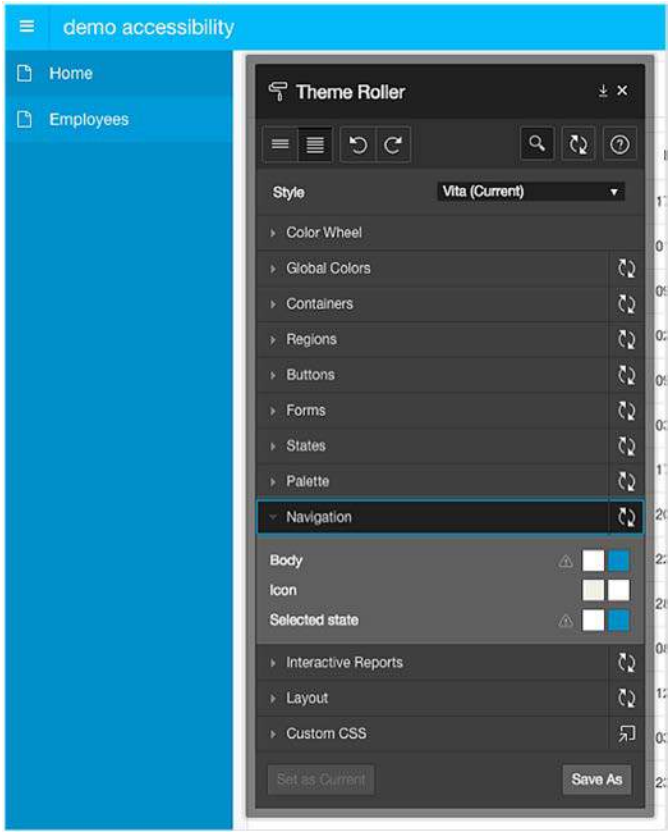
  Generating (complex) html with sys.htp.p in the pl/sql code of a plugin or “Region of type pl/sql”.

- Solution:

  Check the page on errors with third party validators;e.g. W3C-validator (https://validator.w3.org/)

Common mistakes • #2 Wrong color contrast

- Example: >>
- Problem:
  Tiring for the eyes, difficult to read for people with poor vision
- Extra attention needed when:
  Changing colors with the theme-roller
- Solution:
  Pay attention to contrast-check icon in the theme roller. Adjust colors until ok.



Common mistakes • #3 Not keyboard navigable

- Example:

```
<div onclick='alert("clicked")'>
Click me to show alert
</div>
```

- Problem:

  Not accessible for users with no mouse

- Extra attention needed when:

  Creating own HTML-code

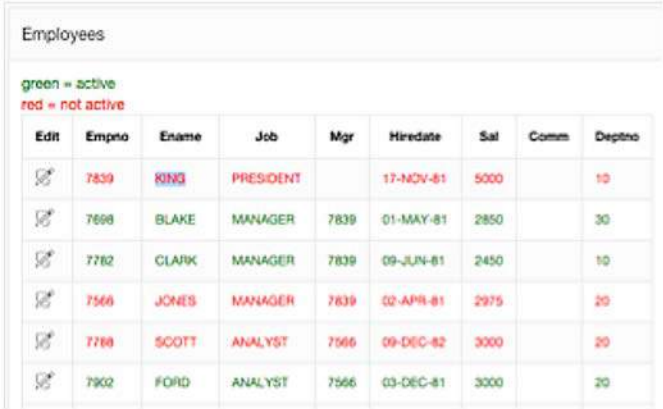
- Solution: Add tabindex=”0” to the element

```
<div onclick='alert("clicked")’ tabindex="0">
  Click me to show alert
</div>
```

tabindex = 0; natural order on page
tabindex = -1; not keyboard navigable
tabindex > 0; sequence when navigating thru page

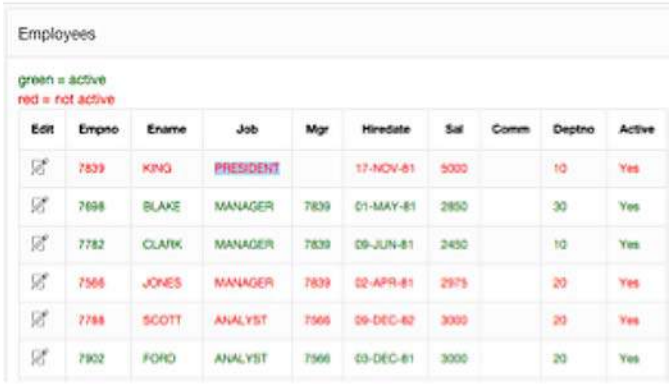
Common mistakes • #4 Wrong color use

- Example: >>
- Problem:
  Difficult to read for people with poor vision, color blindness, etc.
- Extra attention needed when:
  Working with CS
- Solution: >>
  Add the same information with text or graphics


Employees

green = active
red = not active

Edit	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
	7839	KING	PRESIDENT		17-NOV-81	5000		10
	7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
	7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
	7566	JONES	MANAGER	7839	02-APR-81	2975		20
	7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
	7902	FORD	ANALYST	7566	03-DEC-81	3000		20


Employees

green = active
red = not active

Edit	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno	Active
	7839	KING	PRESIDENT		17-NOV-81	5000		10	Yes
	7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30	Yes
	7782	CLARK	MANAGER	7839	09-JUN-81	2450		10	Yes
	7566	JONES	MANAGER	7839	02-APR-81	2975		20	Yes
	7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20	Yes
	7902	FORD	ANALYST	7566	03-DEC-81	3000		20	Yes

Common mistakes • #5 No label available

- Example: >>
- Problem:

  Screenreader can not interpret the value

  Will probably speak “unpronounceable” when working with webfont)
- Extra attention needed when:

  Working with icons or webfonts
- Solution: >>

  - Hide webfont for AT
  - Add a hidden label

  (class u-VisuallyHidden For UT)


```
<style>
.apex-sr-only {
    position: absolute;
    width: 1px;
    height: 1px;
    padding: 0;
    margin: -1px;
    overflow: hidden;
    clip: rect(0,0,0,0);
    border: 0;
}
</style>

<span>
  <span class="t-Icon fa-#GENDER#" aria-hidden=”true”>
</span>
  <span class="apex-sr-only">#GENDER#</span>
</span>
```

WAI-ARIA

Web Accessibility Initiative – Accessible Rich Internet Applications

W3C: *Defines a way to make Web content and Web applications more accessible to people with disabilities.*

It especially helps with dynamic content and advanced user interface controls developed with Ajax, HTML, JavaScript, and related technologies.

- > 100 “aria-*” attributes to choose from
- not fully supported by any browser or screen reader
- many issues; browsers and screen readers respond differently
- recommendation status on W3C

Click here for website!

Commonly used attributes

- aria-label:
  Defines a string value that labels the current element
- **aria-labeledby:**
  Identifies the element that labels the current element.
- **aria-hidden:**
  Identifies the element that labels the current element.
- **aria-live:**
  Indicates that an element will be updated, and describes the types of updates the user agents, assistive technologies, and user can expect from the live region.

ROLE-attribute

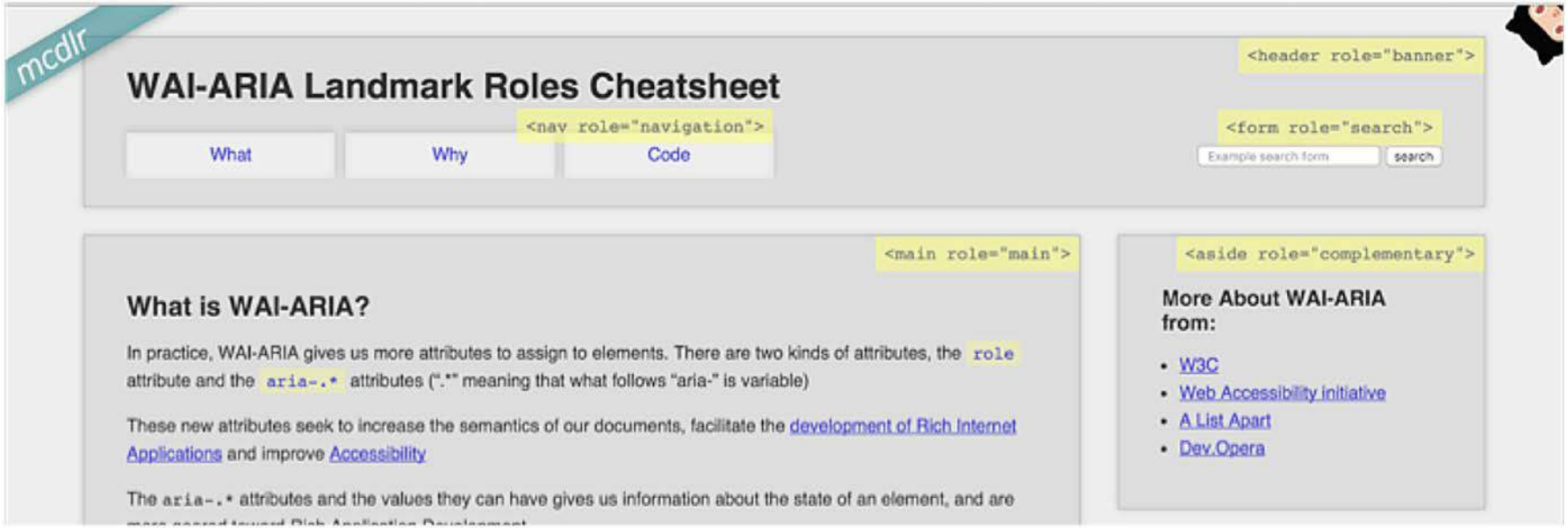
Tells the browser to tell the assistive technology that the HTML element used is not actually what the element name suggests, but something else.

- Corner stone of WAI-ARIA.
- > 50 different roles
- not all combinations of “aria-*=” and “role=” are valid
- **Example:**
  Identifies the element that labels the current element.
- **aria-label:**

```
<div onclick='alert("clicked")' role="button" tabindex="0" aria-label="Click to show alert">
  <span class=" fa fa-play” aria-hidden=”true”></span>
</div>
```

ROLE-attribute

Also used to describe the layout of the webpage.



Commonly used roles

- role=”button”

  An input that allows for user-triggered actions when clicked or pressed

- role=”dialog”

  A dialog is an application window that is designed to interrupt the current processing of an application in order to prompt the user to enter information or require a response.

- role=”menu-item”

  An option in a set of choices contained by a menu or menubar.

- role=”presentation”

  An element whose implicit native role semantics will not be mapped to the accessibility API.

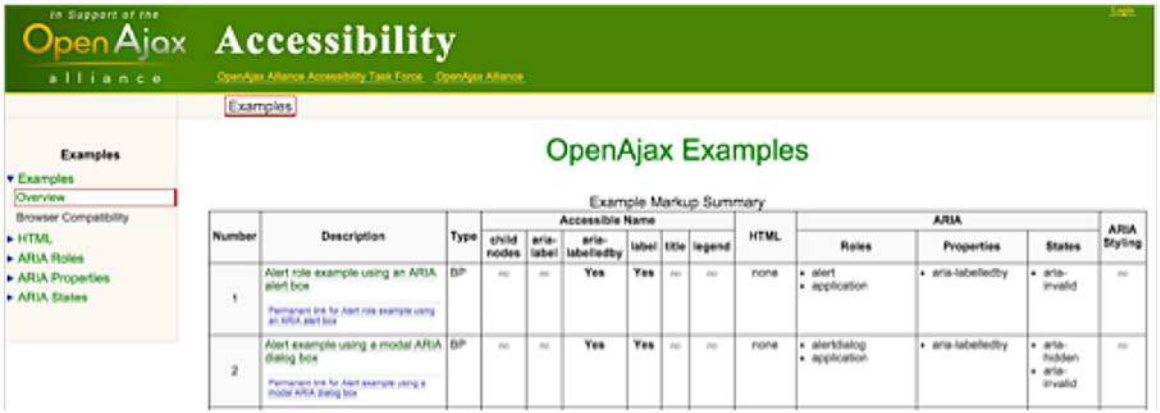
WAI-ARIA Summary

1. Can improve accessibility a lot
2. Not fully supported by AT and browsers
3. No “easy material”
4. APEX development team has done an awesome job for us already.
5. When coding own ui-elements, try to look at “best practices”.

WAI-ARIA Best practices

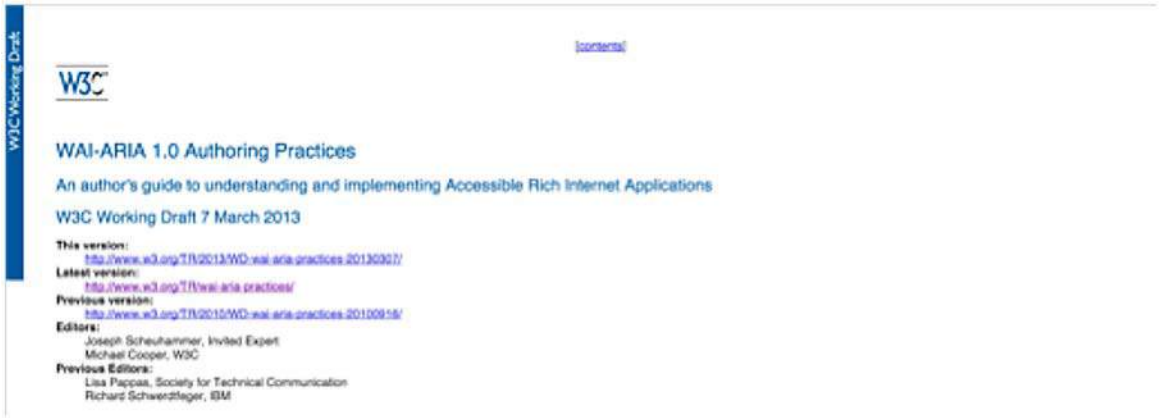
- OpenAjax alliance:

Click here for website!

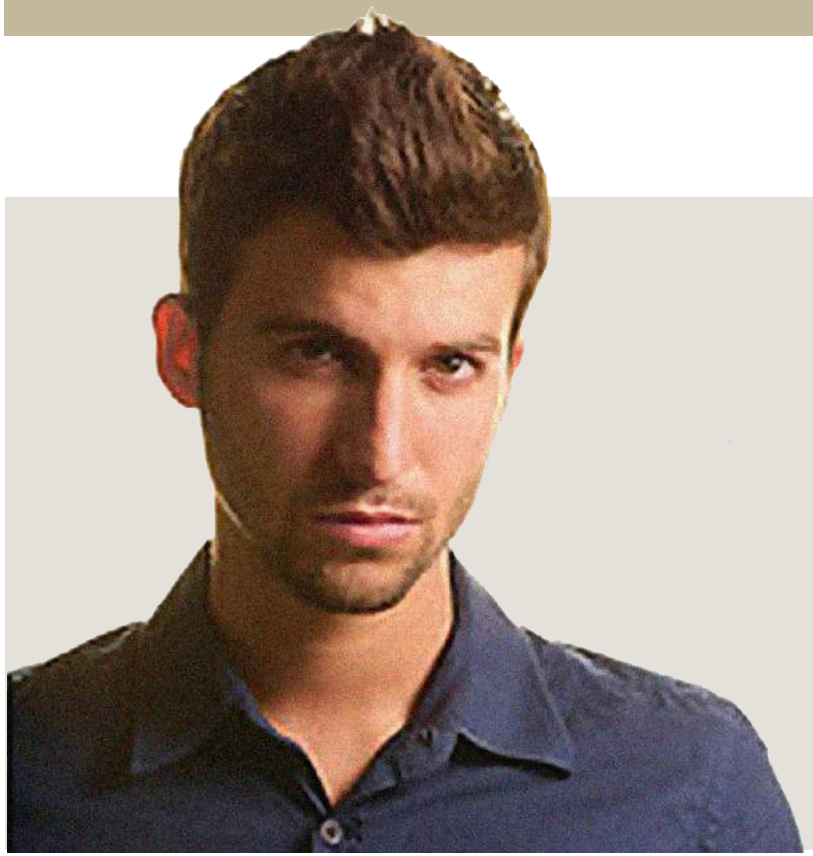


- W3C:

Click here for website!



Thanks you for your attention •



“Questions?“

Tools & links that can help

- Chrome developer toolbar: Audits
- Web accessibility quality control
- See q42: Chrome plugin for simulating eye disabilities
- JAWS shortcuts
- WEBAIM contrast checker
- Oracle & accessibility
- UT accessibility
- Examples of WAI-ARI
- Keyboard support
- APEX API: APEX_UTIL.IS_HIGH_CONTRAST_SESSIONAPEX_UTIL.IS_SCREEN_READER_SESSION and APEX_UTIL.IS_SCREEN_READER_SESSION_YN.