Accessible Graphics for High Pixel Density Era

Presented by: Takeshi Kurosawa

Slides are at: <http://www.slideshare.net/mlca11y/accessible-graphics-for-high-pixel-density-era>

UI elements such as the Facebook icon which is implemented through high density pixel fonts often aren’t accessible.

# Background of icon fonts

High pixel density devices have become popular

* 1 CSS pixel > 1 device pixel
* 1 x graphics will produce artifacts

Do you create graphics for each pixel density? However there are several graphics that would be needed for each image. This also wouldn’t address newer devices that have higher density pixels.

Instead use vector graphics, you can create a graphic and make the devices scale it. Examples of this are:

* Icon fonts
* Scalable Vector Graphics (SVG)

Icon fonts are popular and can be easily deployed, but

* Lack of alternative text
* Unrelated characters may be displayed if specified icon fonts are not used

# Introduction of icon fonts

Icon fonts are glyphs that are icons

They are crisp on high density devices.

Two types.

PUA characters don’t correspond to other characters and therefore the meaning is not conveyed

Examples include FontAwesome

Ligature Icon fonts use ligatures that are a series of glyph forms that create a series of characters.

Ffi -> ffi

For example, a table (or cell) could use Unicode e127

# Issues of private use areas (PUA) icon fonts

Iconic glyphs = PUA characters

CSS pseudo elements use of :before/:after create accessibility issues because lack of lack of alt text or spurious characters.

i.e. Link to a settings page

<a href=”/setting”>

<I class=”fa fa-cog”></i>

Add invisible texts <span clas=”hidden”>settings</span> to improve the accessibility

</a>

But is this enough? For screen readers, yes. But maybe not for other users. I.e. Users that replace your fonts with a Dyslexic font. Therefore PUA characters are not rendered as icons, but as bad characters.

Paciello Group Lessons Learned presentation recommended using Adobe Blank.

<span class=”adobeblank”>Settings</span> instead

This shows appropriate if the PUA disabled.

Alternative texts remain invisible if Adobe Blank is used but icon fonts are note.

Use of unrelated characters being displayed still remains.

Each font uses their own characters for PUA purpose. Therefore different fonts become problematic. (i.e. Seogoe UI symbols U+E106 = Delete, but Linux Libertine U+E106 = Spread Sheet)

# Benefits of ligature icon fonts

<a href=”/setting”>

<span class=”lsf-icon”>setting</span>

</a>

Texts are written in HTML. The purpose of the link is clear for everyone. If the specified fonts aren’t used then it still spells out the right thing.

# Use scenes of icon fonts

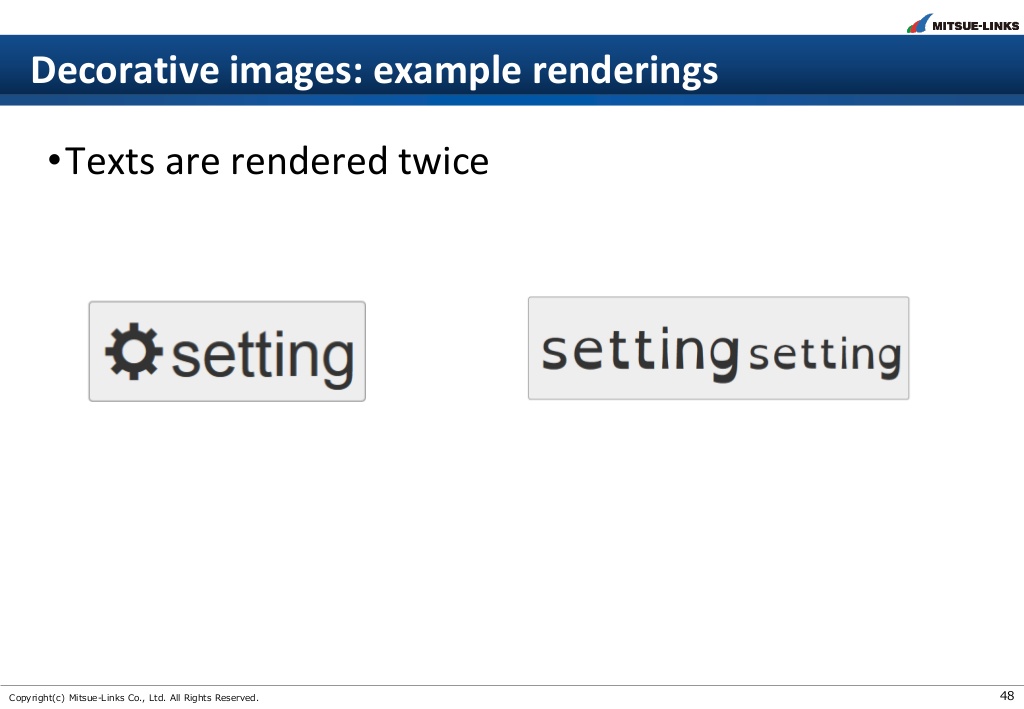
The use of ligature fonts has some limitations.

How

* Using ligature icon fonts (texts must be written in HTML)

For

* Texts with a little orthographic variations
  + Orthographic variations
    - “setting” and “settings” – different ligatures
    - “Settings”, “preference”, “options”, and so on
* Non decorative images
  + If you do so text is replicated twice (setting setting)



* Internationalization
  + Requires ligatures for each language
    - “Settings” in English, “\_\_\_\_” in Japanese

Limitations of icon fonts

* Icon fonts require at least one character to render icons
* ARIA-Hidden cannot solve the issues of renderings

You can use SVG for decorative images (use the right graphics in the right place)

One recommendation is:

Use bootstrap with Grunt to render SVG to modern browsers and png to legacy browsers