Structure
Benefits
All

Writing content for small devices benefits accessibility

Aaron Leventhal, IBM Accessibility Architect
Many smart phones out there now – more usable content needed

Developer’s dilemma:
  • “Write once, run anywhere”  
    or
  • Create a separate mobile version  
    or
  • Somewhere in between
Possibilities

1. Do Nothing
2. Follow Basic Tips
3. Use Handheld CSS
4. Create Mobile Version
5. Add Landmarks
1. If You Do Nothing

- **Large images** eat bandwidth
- **Extra content** makes site slow, difficult to navigate
- **Dynamic effects** often break
- **Missing ALT text** a problem for users browsing w/ images off
- **CSS float & absolute pos** are ignored
- **Might be usable** with Opera & Opera Mini
Opera’s Small Screen Rendering

- Reformats pages to fit screen width
- No horizontal scrolling required
- Pocket IE, NetFront, Minimo also try to do this
2. Follow Basic Tips

- Helps small devices
- Helps a11y
- Helps printing
- Helps search engine robots
## Coding Tips 1

<table>
<thead>
<tr>
<th>Tip</th>
<th>Helps Small Devices</th>
<th>Helps A11y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avoid popups</td>
<td>Distracting, only 1 window at a time</td>
<td>Distracting context change</td>
</tr>
<tr>
<td>Avoid dynamic effects</td>
<td>Cannot assume use of mouse, dynamic changes still difficult for screen readers</td>
<td></td>
</tr>
<tr>
<td>Avoid frames or use titles or <code>&lt;noframes&gt;</code></td>
<td>Frames listed by title, not shown together</td>
<td>Disorients screen reader users</td>
</tr>
<tr>
<td>Avoid low contrast, reliance on color (helps printing also)</td>
<td>Sunlight, weak screens, all text B&amp;W</td>
<td>Color-blind users</td>
</tr>
</tbody>
</table>
## Coding Tips 2

<table>
<thead>
<tr>
<th>Tip</th>
<th>Helps Small Devices</th>
<th>Helps A11y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use headings, CSS, meaningful markup</td>
<td>Smaller download size, faster keyboard navigation</td>
<td></td>
</tr>
<tr>
<td>Use alt text</td>
<td>Images may be off</td>
<td>Screen reader users need text</td>
</tr>
<tr>
<td>Order for importance</td>
<td>Helps users find what they want quickly</td>
<td></td>
</tr>
<tr>
<td>Avoid layout tables</td>
<td>Less code to download</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Content order can reflect importance</td>
<td></td>
</tr>
<tr>
<td>Specify height and width for images</td>
<td>Optimal performance for large image layout</td>
<td>-</td>
</tr>
</tbody>
</table>
The Real Tip:
Test, Think About, Scenarios
3. Use Handheld CSS

- Images, CSS loaded only if needed
  - Cheaper, faster
- More readable, less clutter

```
<link rel="stylesheet" type="text/css" media="handheld" href="hand.css">

img.decorative { display: none; }
nav.redundant { display: none; }
img.as-text { content: attr(alt); }
```
Use CSS absolute positioning or floats for desktop “screen”
Reverse in handheld stylesheet:
{ position: static } or { float: none; }
Make navigation bars { display: none } for “print”

```html
<div id="maincontent">
  <h1>My heading</h1>
  <p>My main content</p>
</div>

<ul id="nav">
  <li><a href="/foo">foo</a></li>
  <li><a href="/bar">bar</a></li>
  <li><a href="/baz">baz</a></li>
</ul>

#maincontent {
  margin-left: 10em;
}

#nav {
  position: absolute;
  top: 0;
  left: 0;
  width: 10em;
}
CSS Problems

- Some devices ignore “handheld”
- Some apply both "screen" & "handheld"
- Some ignore both

Strategies:
- Make handheld CSS reverse any screen CSS
- Test with a variety of devices, if possible
- Count on technology moving forward
- Consider separate handheld content
4. Create a Mobile Version

- Faster download speeds
- Consistent results
- Optimized end user experiences
  - Faster
  - Easier to read
  - Special features - “send SMS when…”
- Requires more developer effort
How to Serve

- Special URLs
  - foo.mobi or www.foo.com/mobile
    harder to remember or type

- User Agent sniffing
  - difficult to code, but possible
  - nicer for user
No “Correct” Doctype

<table>
<thead>
<tr>
<th>Doctype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cHTML, WAP 1.0</td>
<td>Out of date</td>
</tr>
<tr>
<td>XHTML Basic</td>
<td>Limiting, no CSS</td>
</tr>
<tr>
<td>HTML</td>
<td>Limiting, no SVG or X+V</td>
</tr>
<tr>
<td>XHTML MP</td>
<td>- Sensible mobile subset of XHTML 1.x, validation can help find errors</td>
</tr>
<tr>
<td></td>
<td>- Allows SVG, X+V if application/xml+xhtml</td>
</tr>
<tr>
<td>XHTML 1.x</td>
<td>Full range of capabilities</td>
</tr>
<tr>
<td>Approach</td>
<td>Pro</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>1. Do nothing</td>
<td>Free!</td>
</tr>
<tr>
<td>2. Follow basic tips</td>
<td>Easy, good a11y</td>
</tr>
<tr>
<td>3. Handheld CSS</td>
<td>Fairly easy</td>
</tr>
<tr>
<td></td>
<td>Smaller downloads</td>
</tr>
<tr>
<td></td>
<td>Less clutter</td>
</tr>
<tr>
<td>4. Mobile version</td>
<td>Tailored to mobile devices &amp; uses</td>
</tr>
</tbody>
</table>
5. Future: Add Landmarks

- Good navigation is important
- How can I get to the main content quickly?
- How can I search for something?
- What’s related?
Setting the role

- In XHTML (usable in XHTML 1.x, defined in 2.0):
  role="maincontent" | "note" | "navigation" | "search" | "contentinfo" | "note" | "secondary" | "seealso"

- In HTML,
  <link rel="maincontent" href="#main">

- Potential Microformat for HTML, e.g.
  class="axs navigation"
How user agents will use it

- **Mobile browser**
  Can start off scrolled to main content
  Provide a shortcut for searching

- **Screen readers**
  Start user in main content or provide shortcut keys for direct access

- **Screen magnifiers**
  Key to toggle visibility of secondary content

- **Key navigation extensions**
  Key to navigate quickly to search, navbar
Summary

- Convergence is happening, special markup not required for mobile
- Confusion still reigns supreme
- Usability problems abound

- It is possible to be fairly mobile-friendly and accessible using simple techniques
- The future will provide new tricks that help both mobile device use and accessibility
Resources

- Advanced CSS layout: http://www.htmldog.com/guides/cssadvanced/layout
- Detecting UA and Client Capabilities http://www.developershome.com/wap/detection/
- XHTML 2 http://www.w3.org/TR/xhtml2/