

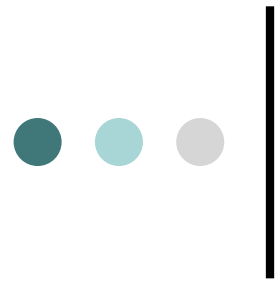
Writing content for small devices
benefits accessibility

Aaron Leventhal, IBM Accessibility Architect



Hungry Little Browsers

- 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

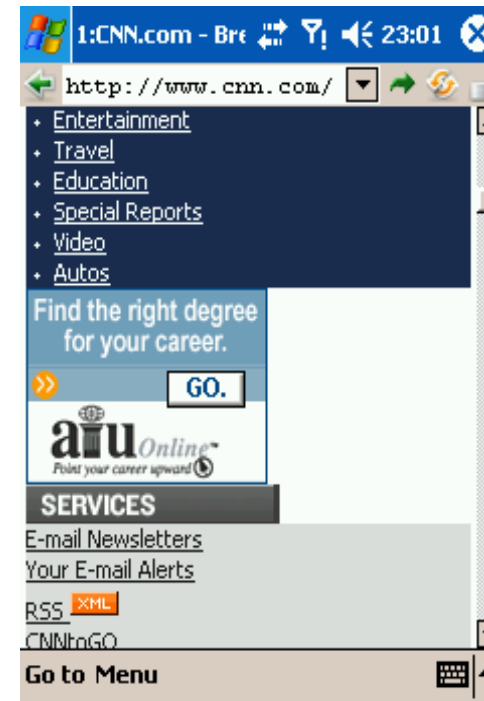


1. If You Do Nothing

- 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





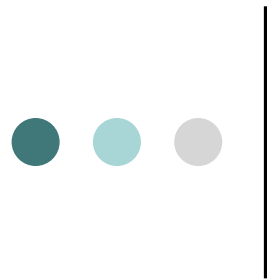
2. Follow Basic Tips

- Helps small devices
- Helps a11y
- Helps printing
- Helps search engine robots



Coding Tips 1

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



Coding Tips 2

Tip	Helps Small Devices	Helps A11y
Use headings, CSS, meaningful markup	Smaller download size, faster keyboard navigation	
Use alt text	Images may be off	Screen reader users need text
Order for importance	Helps users find what they want quickly	
Avoid layout tables	Less code to download Content order can reflect importance	
Specify height and width for images	Optimal performance for large image layout	-

● ● ● | 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); }
```



CSS Layout

- 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”

```
<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



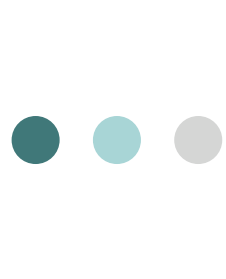
4. Create a Mobile Version

- 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

cHTML, WAP 1.0	Out of date
XHTML Basic	Limiting, no CSS
HTML	Limiting, no SVG or X+V
XHTML MP	<ul style="list-style-type: none">- Sensible mobile subset of XHTML 1.x, validation can help find errors- Allows SVG, X+V if application/xml+xhtml
XHTML 1.x	Full range of capabilities



Choice Matrix

Approach	Pro	Con
1. Do nothing	Free!	Broken!
2. Follow basic tips	Easy, good a11y	Not optimal, hard to navigate
3. Handheld CSS	Fairly easy Smaller downloads Less clutter	Not yet universal
4. Mobile version	Tailored to mobile devices & uses	Expensive & difficult



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 seaching
- *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 a11y



Resources

- Advanced CSS layout:
<http://www.html5dog.com/guides/cssadvanced/layout>
- Making Small Devices Look Great:
<http://my.opera.com/community/dev/device>
- Detecting UA and Client Capabilities
<http://www.developershome.com/wap/detection/>
- XHTML 2
<http://www.w3.org/TR/xhtml2/>