

Client-Side Technologies

Ivan Pepelnjak (@ioshints, ip@ioshints.info)
NIL Data Communications



ipSpace

Topics

Mandatory technologies:

- HTML
- CSS
- JavaScript
- Libraries/Frameworks

Not discussed:

- HTML5 additions (canvas)
- Graphics
- Video

<https://www.evernote.com/pub/ioshints/scaleMatters>

HTML



ipSpace

HTML – Basic Page Structure – Header

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xml:lang="en" lang="en" dir="ltr">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta http-equiv="Content-Style-Type" content="text/css" />
    <meta name="keywords"
        content="Customized webinars,ExpertExpress,Recordings" />
    <link rel="shortcut icon" href="/favicon.ico" />
    <link rel="search" type="application/opensearchdescription+xml"
        href="/wk/opensearch_desc.php" title="ipSpace.net" />
    <title>ipSpace.net by @ioshints</title>
    <link rel="stylesheet" type="text/css" href="/css/wiki20.css?207" />
</head>
```

Basic Rules

Well-behaved HTML:

- Proper tag nesting
- Closing (`text`) or self-closing (`
`) tags
- Quoted attribute values
- Lowercase tag and attribute names

Browsers survive lots of stupidities, but don't push them too far

HTML5 just made matters way more complex

HTML or XHTML?

- HTML is SMGL-based syntax
- XHTML is HTML content in XML format (more strict than SMGL)
- Strict XHTML has failed

Use XHTML-compliant HTML whenever possible (makes it easier to consume by server-side scripts)

- Follow basic rules of good behavior
- Add DOCTYPE
- Use W3C validator

The DOCTYPE Saga

Initial ideas:

- DOCTYPE inherited from XML (specifies validation schema)
- DOCTYPE schema should indicate HTML type (HTML or XHTML, Strict or Transitional)

Reality:

- DOCTYPE is another failed experiment
- Used primarily to turn off Quirks mode
- Use `<!DOCTYPE html>`

The Magic LINK Element (REL attribute)

alternate	Hyperlink	Gives alternate representations of the current document.
author	Hyperlink	Gives a link to the current document's author.
help	Hyperlink	Provides a link to context-sensitive help.
icon	External Resource	Imports an icon to represent the current document.
license	Hyperlink	Indicates that the main content of the current document is covered by the copyright license described by the referenced document.
next	Hyperlink	Indicates that the current document is a part of a series, and that the next document in the series is the referenced document.
prefetch	External Resource	Specifies that the target resource should be preemptively cached.
prev	Hyperlink	Indicates that the current document is a part of a series, and that the previous document in the series is the referenced document.
search	Hyperlink	Gives a link to a resource that can be used to search through the current document and its related pages.
stylesheet	External Resource	Imports a stylesheet.

HTML – Basic Page Structure – Body

```
<body class="mediawiki ltr ns-0 ns-subject page-Main_Page">  
  ...  
  <!-- scripts and debugging information -->  
  <script language="javascript"  
    src="/libs/jquery/1.4.2/jquery.min.js"></script>  
  <script language="javascript"  
    src="/libs/jqueryui/1.8.5/jquery-ui.min.js"></script>  
</body>  
</html>
```

Learn From Waterfalls

URL	Status	Domain	Size	Remote IP	Timeline
▶ GET Main_Page	200 OK	ipSpace.net	14.8 KB	193.110.145.52:80	315ms
▶ GET SelectCategoryTa	200 OK	ipSpace.net	2.1 KB	193.110.145.52:80	11ms
▶ GET wiki20.css?207	200 OK	ipSpace.net	12.5 KB	193.110.145.52:80	87ms
▶ GET ajax.js?207	200 OK	ipSpace.net	4.7 KB	193.110.145.52:80	63ms
▶ GET ipSpaceLogo.jpg	200 OK	ipSpace.net	5.2 KB	193.110.145.52:80	66ms
▶ GET Wklvan.png	200 OK	ipSpace.net	1.4 KB	193.110.145.52:80	47ms
▶ GET WkURHere.png	200 OK	ipSpace.net	7 KB	193.110.145.52:80	64ms
▶ GET IOS_Hints_home-	200 OK	ipSpace.net	82.8 KB	193.110.145.52:80	131ms
▶ GET jquery.min.js	200 OK	ajax.googleapis.com	24 KB	173.194.67.95:80	88ms
▶ GET jquery-ui.min.js	200 OK	ajax.googleapis.com	49.4 KB	173.194.67.95:80	163ms
▶ GET wiki20.js	200 OK	ipSpace.net	2 KB	193.110.145.52:80	54ms
▶ GET bodybg.jpg	200 OK	ipSpace.net	1.1 KB	193.110.145.52:80	11ms
▶ GET headerbg.jpg	200 OK	ipSpace.net	9.5 KB	193.110.145.52:80	17ms
▶ GET wikiBullet.gif	200 OK	ipSpace.net	827 B	193.110.145.52:80	16ms
▶ GET Pipi_150x150.jpg	200 OK	ipSpace.net	7.7 KB	193.110.145.52:80	34ms
▶ GET ga.js	200 OK	google-analytics.com	13.1 KB	173.194.67.100:80	85ms
▶ GET __utm.gif?utmwv=	200 OK	google-analytics.com	35 B	173.194.67.100:80	43ms
17 requests			238.2 KB	822ms (onload: 828ms)	

- Page components start to load ASAP
- CSS loading blocks page rendering
- SCRIPT loading blocks rendering beyond the SCRIPT tag
- TABLE renders after it's completely loaded

Lessons Learned

- CSS at the top
- SCRIPT at the bottom
- TABLE is not a layout tool

Additional speedups:

- Non-blocking script loading (after the page has loaded)
- Delayed content loading with AJAX (ex: Twitter)



Client-Side Technologies – Generic Guidelines



Basic Hygiene

- Separate content (HTML), formatting (CSS) and actions (JS)
- Formatting and actions should be reusable
 - ➔ Use external CSS and JavaScript files
- Validate your markup

NEVER

- Define an event handler with HTML element attribute
`<body onload="loaded()">`

NO!!!

Try not to

- Use STYLE tags in HTML elements (use in-page stylesheet)
- Embed code with SCRIPT tags

More Usage Guidelines

- Read the HTML specs (really)
- Use semantic markup (HEADER, FOOTER ... instead of DIV)
- Google is your friend (let him find you)
 - ➔ Use markup, not JavaScript to build menus
 - ➔ All content must be reachable through A or LINK elements

Graceful degradation: things should work even when something is broken

- Example: Pull-down menus without JavaScript

Progressive enhancement:

- Start with baseline markup
- Add formatting (in-page styles ➔ external CSS)
- Enhance with JavaScript

Make Your Web Page Accessible

Addressing disabilities (visual, hearing, physical ...) will also make your web pages better and faster

- Have meaningful titles and bypass links
- Use headings and (form) labels
- Maintain focus order
- Provide text alternatives for all non-text content (use TITLE and ALT HTML attributes)
- Create content that can be presented in a meaningful way (ex: screen reader)
- Some people are color-blind
- Have mechanisms to stop/mute the audio
- Text should be resizable (hint: don't use pixels in CSS)
- All functionality should be accessible from the keyboard

More @ <http://www.w3.org/TR/WCAG20/>

Putting It All Together

- Two audiences: visitors + search engines
- Use semantic markup
- Special considerations: mobile audiences
 - Different UI expectations
 - Smaller screens
 - Touchscreen interactions
- Understand the web page composition
- Progressive enhancement & Graceful degradation
- Why are tables bad?
- CSS on top, JavaScript on bottom

Questions?

