

Web-based Interactive Media



Tools

Checklist

- What you will need:
 - Browser(s)
 - Browser extensions for developers
 - Editor
- What you might need:
 - Rich IDE
 - Graphics software
 - MAMP / WAMP / LAMP stack
 - Dropbox account

Browsers

- Firefox (free)
- Google Chrome (free)
- Opera (free)
- Safari (free)

– What about IE?

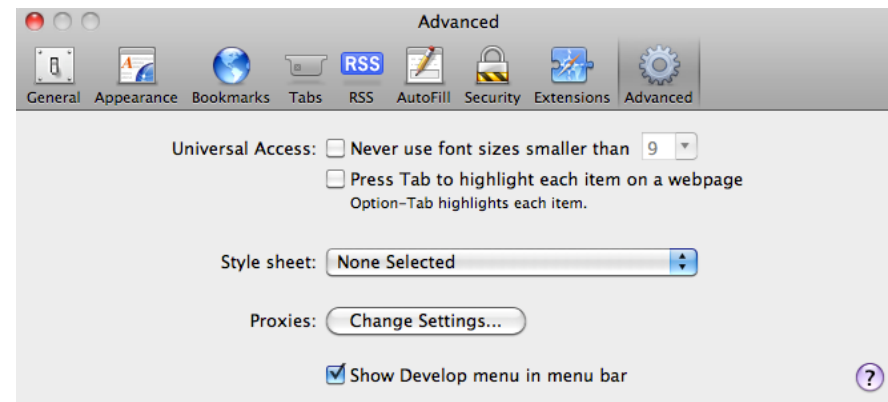


Browser extensions for developers

- Firefox
 - [Firebug](#) (also available for Chrome as [Firebug Lite](#))
 - [Web Developer](#) (also available for Chrome)
 - [Greasemonkey](#)
- Chrome (built-in)
 - View | Developer | Developer Tools
- IE
 - [IE Developer Toolbar](#)

Browser extensions for developers

- Safari (built-in)
 - Enable Develop menu (Preferences | Advanced)
 - Develop menu:
 - Web Inspector
 - Error Console
 - (Snippet Editor)
 - (Extension Builder)



Editors

- Choose one (or more) from this list:
 - [Notepad++](#) (Win) (free)
 - [TextWrangler](#) (Mac) (free)
 - [BBEdit](#) (Mac)
 - Komodo (Mac)

Rich IDEs

- [Aptana Studio](#) (free)
- [Eclipse](#) (with Aptana Plugin) (free)
- [coderun](#) (free)

Graphics software

- From [IrfanView](#) ...
- ... to ...
- ... [Photoshop](#)
 - With a million options in between

Dropbox account

- Use this link (and we both get free space 😊):



HTML5 basics

```
002 <head>  
003 <meta charset="utf-8" />  
004 <title>HTML5</title>  
005 <link rel="stylesheet" href="css/main.css" />  
006 <link rel="stylesheet" href="css/main.css" />  
007 </head>  
008 <body id="index" class="home">  
009 <div id="banner" class="body">  
010 <h1 href="#">HTML5 <strong>HTML5  
012 </ins>2009</ins></h1>
```

Anatomy of a HTML5 web page

Web pages

- Plain text
- (Broken) HTML
- (Strict) (X)HTML
- HTML5
- HTML5 + CSS

Static

- HTML5 + CSS + JavaScript
- HTML5 + CSS + JavaScript + jQuery

Rich
Media

Hands-on

- Let's create our first HTML5 page!
 - Choose your favorite editor (even Notepad or vi should do 😊)
 - Start from an empty page
 - Add *DOCTYPE*
 - Add *html*, *head*, and *body* tags
 - Add *meta* and *title* (under *head*)
 - Add a *p* (under *body*)
 - Save
 - Display in browser
 - View source

Before HTML5...

- HTML (frozen at 4.01)
- XHTML (XML version of HTML)
 - Transitional
 - Strict
- Early work on XHTML 2.0
- Opera: Web Forms 2.0
- [WHATWG](#) (Web Hypertext Application Technology Working Group)
- 2009: W3C shifts from XHTML 2.0 to HTML5

Good design before HTML5

- Excessive use of `<div>`s
- No semantic meaning associated with tags
- JavaScript selectors
- CSS for style and presentation issues

Philosophies behind HTML5

- Design principles

- Compatibility

- If HTML5 features are not supported, the behavior must degrade gracefully
 - The name of the game is evolution not revolution

- Utility

- Priority of Constituencies.
 - And as priorities go, “the user is king.”
 - Users >> authors >> implementers (browsers) >> specifiers (W3C/WHATWG) >> theoretical purity

Philosophies behind HTML5

- Design principles (cont' d)
 - Interoperability
 - The HTML5 mantra?
 - “Simple is better. Simplify wherever possible.”
 - Universal Access
 - 3 concepts:
 - Accessibility: support for users with disabilities [Web Accessibility Initiative (WAI) Accessible Rich Internet Applications (ARIA)]
 - Media Independence: HTML5 functionality should work across all different devices and platforms if at all possible.
 - Support for all world languages.

HTML5 highlights

- More descriptive markups
- Multimedia with less reliance on plug-ins
- Better web-based apps without third-party technologies
- Cross-document messaging
- Web Sockets (persistent connection to server)
- Client-side storage
- Better interfaces
- Improved accessibility

Styling HTML5 with CSS

- CSS stands for Cascading Style Sheets
- Styles define how to display HTML elements
- Styles were added to HTML 4.0 to solve a problem
- External Style Sheets can save a lot of work
- External Style Sheets are stored in CSS files
- CSS enforce separation between presentation and content
- Let' s take a walk in the [garden](#)...

HTML5: new features

- The next few slides will provide a summary of new features in HTML5.
- They are primarily targeted at those who have been exposed to earlier versions of (X)HTML.

New DOCTYPE and character set

- Before:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01 Transitional//EN" "http://  
www.w3.org/TR/html4/loose.dtd">
```

- After:

```
<!DOCTYPE html>
```

- Before:

```
<meta http-equiv="Content-Type"  
content="text/html; charset=utf-8">
```

- After:

```
<meta charset="utf-8">
```

New sectioning elements

Sectioning Element	Description
header	Header content (for a page or a section of the page)
footer	Footer content (for a page or a section of the page)
section	A section in a web page
article	Independent article content
aside	Related content or pull quotes
nav	Navigational aids

New sectioning elements



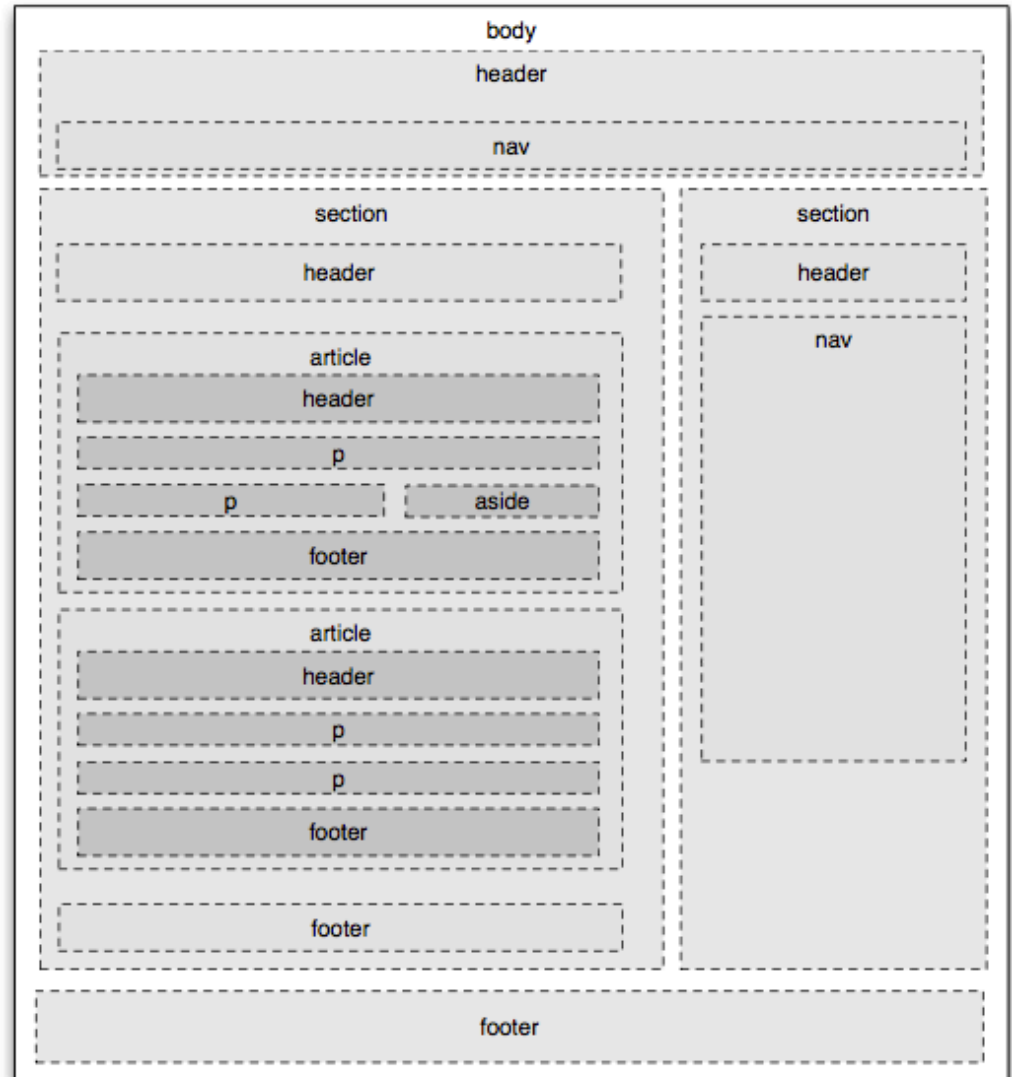
Hands-on

- Let's get rid of "Divitis" ...
 - Compare:
 - Example3_2.html (many `<div>`s)

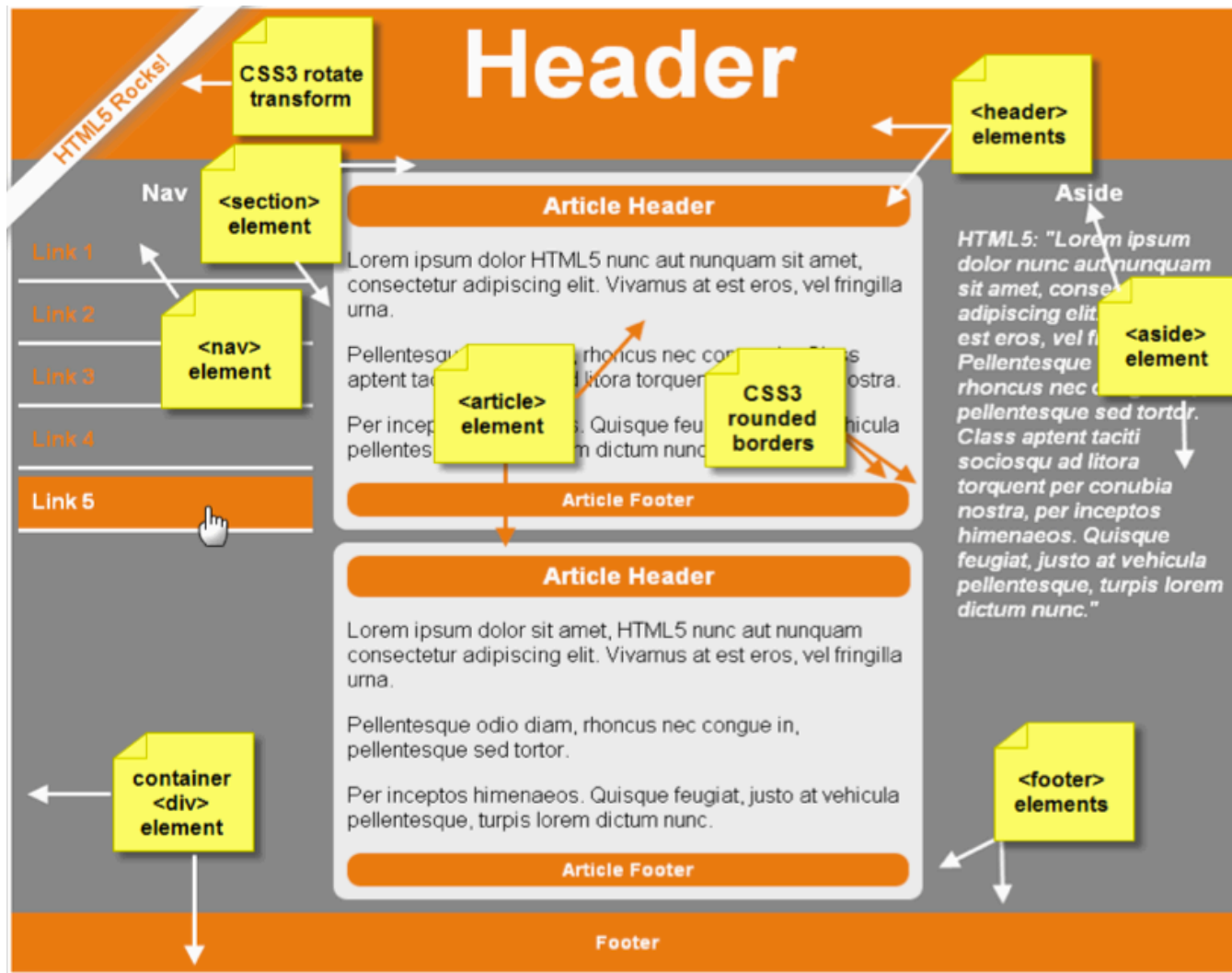
with
 - Example3_3.html (HTML5 sectioning elements)

Example from book I

- Starts on page 16
- html5newtags/index.html
- [Live version](#)



Another example (not from textbook)



Code: Example3_4.html + html5.css

HTML5 content types

- HTML5 introduces many new markup elements, which it groups into seven different content types.

Content Type	Description
Embedded	Content that imports other resources into the document, for example <code>audio</code> , <code>video</code> , <code>canvas</code> , and <code>iframe</code>
Flow	Elements used in the body of documents and applications, for example <code>form</code> , <code>h1</code> , and <code>small</code>
Heading	Section headers, for example <code>h1</code> , <code>h2</code> , and <code>hgroup</code>
Interactive	Content that users interact with, for example <code>audio</code> or <code>video</code> controls, <code>button</code> , and <code>textarea</code>
Metadata	Elements—commonly found in the head section—that set up the presentation or behavior of the rest of the document, for example <code>script</code> , <code>style</code> , and <code>title</code>
Phrasing	Text and text markup elements, for example <code>mark</code> , <code>kbd</code> , <code>sub</code> , and <code>sup</code>
Sectioning	Elements that define sections in the document, for example <code>article</code> , <code>aside</code> , and <code>title</code>

Deprecated tags

- `basefont`
- `big`
- `center`
- `font`
- `s`
- `strike`
- `tt`
- `u`

Deprecated tags

- Replacements
 - *acronym* gets replaced by *abbr*
 - *applet* gets replaced by *object*
 - *dir* gets replaced by *ul*

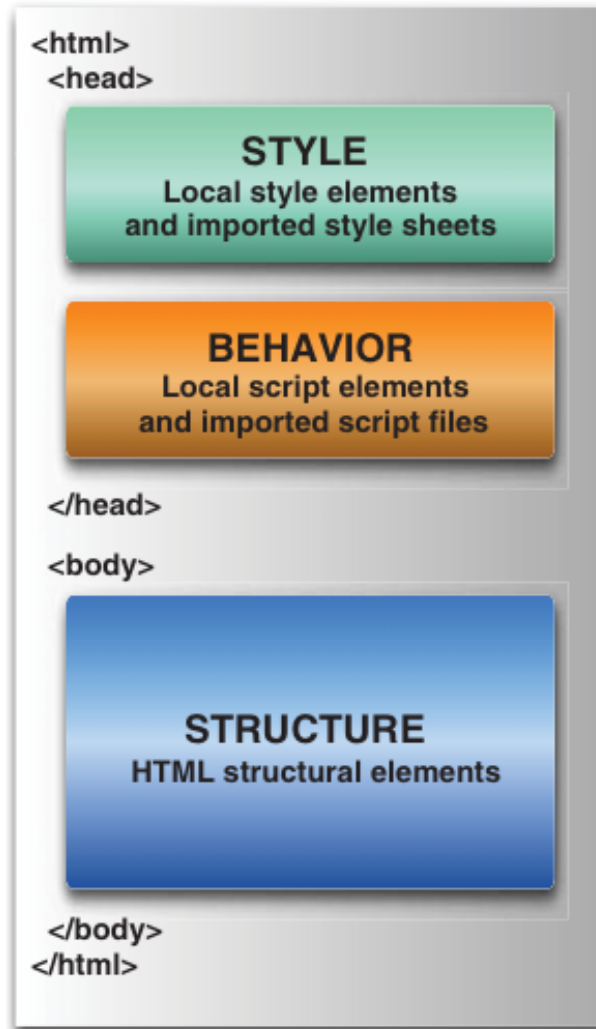
Frames

- Support for frames has been removed.
 - Good riddance!
- That means these elements are gone:
 - frame
 - frameset
 - noframes

Deprecated attributes

- Many attributes that are no longer valid.
 - These include presentational attributes such as:
 - *align*
 - *link*, *vlink*, *alink*, and text attributes on the *body* tag
 - *bgcolor*
 - *height* and *width*
 - *scrolling* on the *iframe* element
 - *valign*
 - *hspace* and *vspace*
 - *cellpadding*, *cellspacing*, and *border* on table

HTML, CSS, and JavaScript



Any question at this point

?

And now do it yourself