

Demo: html5 structure

Simple Text/Web Editor

- TextEdit -make plain text (osx)
- Notepad++ (windows)
- TextWrangler (osx)
- Coda (osx)
- Sublime Text (osx & windows)
- Brackets (osx & windows)

General Workflow (nomadic version)

- Set up/verify web folder on local hard drive
- Set up FTP (file transfer protocol) client
- Download web files (html, css, jpg, gif, png) from server to local folder
- Open/create web files from the local web folder (ONLY)
- Edit web files
- Test locally in browser (chrome)
- Chrome inspect element
- Validate code (w3c validator)
- Upload web files from local drive to server
- Test live in browser (multiple browsers)

What is html?

- acronym for **h**ypertext **m**arkup **l**anguage
- hypertext means the ability to link to other documents.
- markup is a language for describing web pages.
- markup tags define the semantic structure of content in web pages
- “view source” in any browser to see the html markup of a webpage

html tags

- html markup is called “tags”
- tags are special keywords surrounded by angle brackets
 - `<html> <body> <head> <title>`
- html tags normally come in pairs
 - `<p> </p>`
 - exception is “empty tags”
 - `<meta>
 `
 - no closing tag because they don’t enclose anything they are “empty”
- First tag = start or open tag
- Second tag = end or close tag

html document = web page

- html documents contain html tags and plain text to describe web pages
- Web browsers (Chrome, Safari, Firefox, IE, Opera) read html documents and display them as web pages.
- Browsers use tags to interpret the structure (outline) of web page to create the DOM (document object model)

Example of bare minimum html document:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello World</title>
  </head>
  <body>
    Hello World!
  </body>
</html>
```

html example explained

- `<!doctype html>` = specifies html5 rule set
- `<html>` = html document
- `<head>` = instructions for the browser
 - meta, title, css, javascript
- `<meta charset="utf-8">` = specifies character encoding
- `<title>` = title of document
- `<body>` = visible page content

File extensions

- .html or .htm
- .html more popular = better

CaSE SeNSITiViTY

- html5 is no longer case sensitive (however)
- most developers keep their tags in lowercase for consistency (and as a nod to predecessor xhtml)
- good writing style

Doctype declaration

- you **MUST** specify the doctype in all of your html documents
 - tells browser what type of document to expect and what rules to follow.
 - consistent rendering across browsers
 - otherwise you get quirks mode
 - emulates 1990's era browser
 - not good
 - the doctype declaration **MUST** be on the first line in your html document and before the `<html>` tag.
 - doctype declaration for html5
 - simplified version of previous doctypes
- ```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
"http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
```

## Structure

- information
  - most important feature of webpage
- webpages are structured in outline form
  - headings
  - paragraphs
  - lists

- figures (informational images)
- parent-child relationships
  - most html elements can be nested (can contain other html elements).
- good structure
  - ACCESSIBILITY
  - SEO
  - DOM

## html5 structural markup

- <header>
- <nav>
- <section>
- <main>
- <article>
- <aside>
- <footer>

## Generic structure tags

- <div> (block level)
- <span> (inline)
- used with CSS for design

## html headings

- think outline –use html headings for the purpose of headings only
  - DO NOT use headings to make text big or bold (presentational)
- search engines use headings to index your pages (SEO)
- use headings in order of hierarchy:
 

```
<h1>This is a heading</h1>
<h2>This is a sub heading</h2>
<h3>This is a sub sub heading</h3>
```

## html paragraphs

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

## html links

- absolute link
  - full http address
  - link to external website

```
link
```
- relative link
  - local .html address
  - link to internal web page
  - internal site navigation

```
home
```

## html attributes

- html elements can have attributes
- provide additional information about an element
- always specified in the start tag
 

```
<tag attribute="value"></tag>
```
- attributes must have values
- values should be in quotes (writing style)

## html images (informational)

- informational imagery using the `img` tag should be in `<figure>` tags
 

```
<figure>

</figure>
```
- `img` tag creates a placeholder for images
- `src` attribute fills placeholder with image
  - common for images to reside in an `img` (images) folder
- `alt` attribute specifies an alternate text for an image
  - required for validation
  - visible when image cannot be displayed (broken)
  - useful for people with disabilities (accessibility)
  - also useful as keywords for search engines (seo)
- design (non-informational) imagery is handled by css
  - placed in background of elements
  - displayed visibly but hidden semantically

## html comments

- inserted into the HTML code as programmer's notes
- makes code more readable or understandable
- ignored when rendered by the browser and are not displayed
 

```
<!-- This is an html comment -->
```

## Lists

### Ordered

```



```

### Unordered

```



```

## special characters

<code>&amp;lt;</code>	is the same as <code>&lt;</code>
<code>&amp;gt;</code>	is the same as <code>&gt;</code>
<code>&amp;#169;</code>	is the same as ©
<code>&amp;nbsp;</code>	non-breaking space