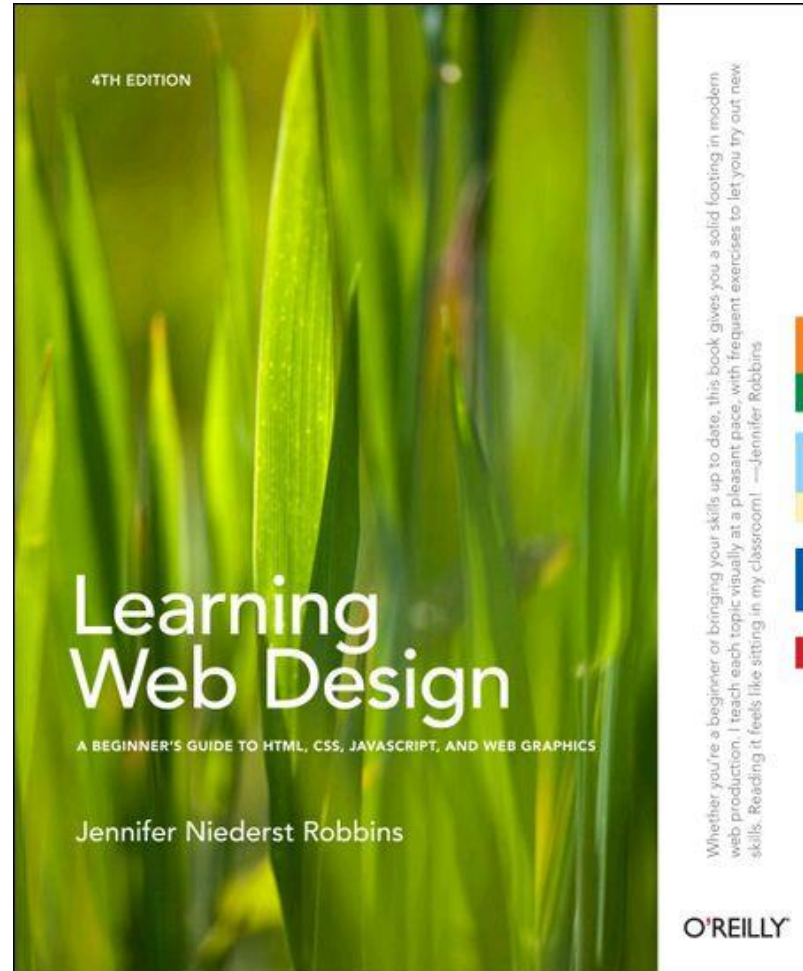


WD-400 HTML & CSS

The making of a web site

Course material

- **Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics** by Jennifer Niederst Robbins – August 2012, O'Reilly



Github resources

- Core resources
 - <https://github.com/doughoff>
- WD-400 repo
 - <https://github.com/doughoff/WD-400>





Web site admin basics

Site directory organization



- Sites you are responsible for
 - Site name
 - hosting info
 - accounting info
 - original text / graphic files
 - site name / html / www
 - the html files here

Site structure - html directory



- Flat structure, minimum depth
- Root folder (html / www/ ...)
 - images
 - js / scripts
 - css
 - sitename.css
 - default.html or index.html
 - <content folder 1>
 - images
 - <content folder 2>
 - images

HTML file names



- default.html
 - more modern
- index.html
 - old school but better supported
- suffixes – html or htm?

HTML file suffixes

- htm, html
- asp, aspx, cshtml, vbhtml
 - Pre .NET, and post .NET
- php, jsp, do, cfm
 - php, Java, Cold Fusion



Questions



- What is the most important thing to consider when putting all your web site files in one place?
- Create a web site structure for this class' exercises
 - Download tools first



The original structure of HTML

XML

XML hierarchy



- XML rules
 - basic rules for creating an XML language
- XML language
 - extra rules for element names, order, and datatypes
 - XHTML, RSS, SVG, ...
- XML document
 - the valid data that follows the XML language rules

Basic XML terms



- **XML document** – the entire text
- **element** – anything between and including the open tag and matching closing tag
 - `<a>ajdfkjhadhklha lj alskfjh ld<c>hjad jf hajldf</c>`
 - `<a> = <a/>`
- **tag** – just the open or closing part, or empty element
 - `<a>ajdfkjhadhklha lj alskfjh ld<c>hjad jf hajldf</c>`
 - `<a />`

Basic XML terms



- body – what's in between the open and closing tags
 - `<a>ajdfkjhadhklha lj alskfjh Id<c>hjad jf hajldf</c>`
 - `<a>ajdfkjhadhklha lj alskfjh Id<c>hjad jf hajldf</c>`
- attribute – meta data for the element
 - `<x id="the unknown">Dean Jagger as Dr. Adam Royston</x>`
- XHTML – HTML - XML compliant
 - DTD – document that gives rules for XML language to validate data
 - DOCTYPE – instruction for browser to recognize data

Basic XML terms



- `<a>`
 - aaaaaaaaaaaaaa
 - `<b1>`bbbbbbbbbbbbbbbbbb
bb`<b1>`
 - `<b2>`BBBBBBBBBB`<c>`B`</c>`
BBBBBBBBBB`<b2>`
 - aaaaaaaaaaaaaa
- ``

parent of **b1** is **a**
ancestor of **c** is **a** is but
is not a parent

child of **a** is **b1**
descendent of **a** is **c** but
is not a child

sibling of **b1** is **b2**

DOM manipulation



- DOM – Document Object Model
 - the in-memory version of the HTML page accessible by JavaScript
- Changing the DOM updates the rendered view of the browser
 - The user sees a very responsive interactive page.

XML rules



- more strict than HTML5
- element tag names must match
 - `<tag> body </tag>`
- elements can not be nested “out of sync”
 - Invalid: `<i> body body </i> body `
The invalid example is annotated with three dashed boxes: an orange box around `<i> body`, a red box around ` body </i>`, and a blue box around the entire sequence `<i> body body </i> body `.
 - Valid: `<i> body body </i> body `
- some characters require encoding
 - `<`; bold `>`;

Questions



- Do you use XML?
 - Microsoft document formats .docx, .xlsx
 - Scalable Vector Graphics (SVG)
 - Feedly - RSS
 - Apache Open Office - OASIS Open Document Format for Office Applications file format



HTML structure

HTML purpose



- To add **meaning** to text
 - Not to get text onto a browser
- To provide a **structure** which browsers render with CSS
- For coders, structural meaning makes it
 - Easier to understand
 - Easier to maintain
 - Easier to blend code into

HTML versions



- HTML 1,2,3,4
 - DHTML meant that they considered CSS and JavaScript
 - HTML 4.01 strict, transitional (loose), frameset
- XHTML
 - compliant with XML
 - XHTML 1 strict, transitional, frameset
 - XHTML 1.1
 - XHTML5
- **HTML5**
 - not compliant with XML, but backwards compatible

Web page file structure



doctype

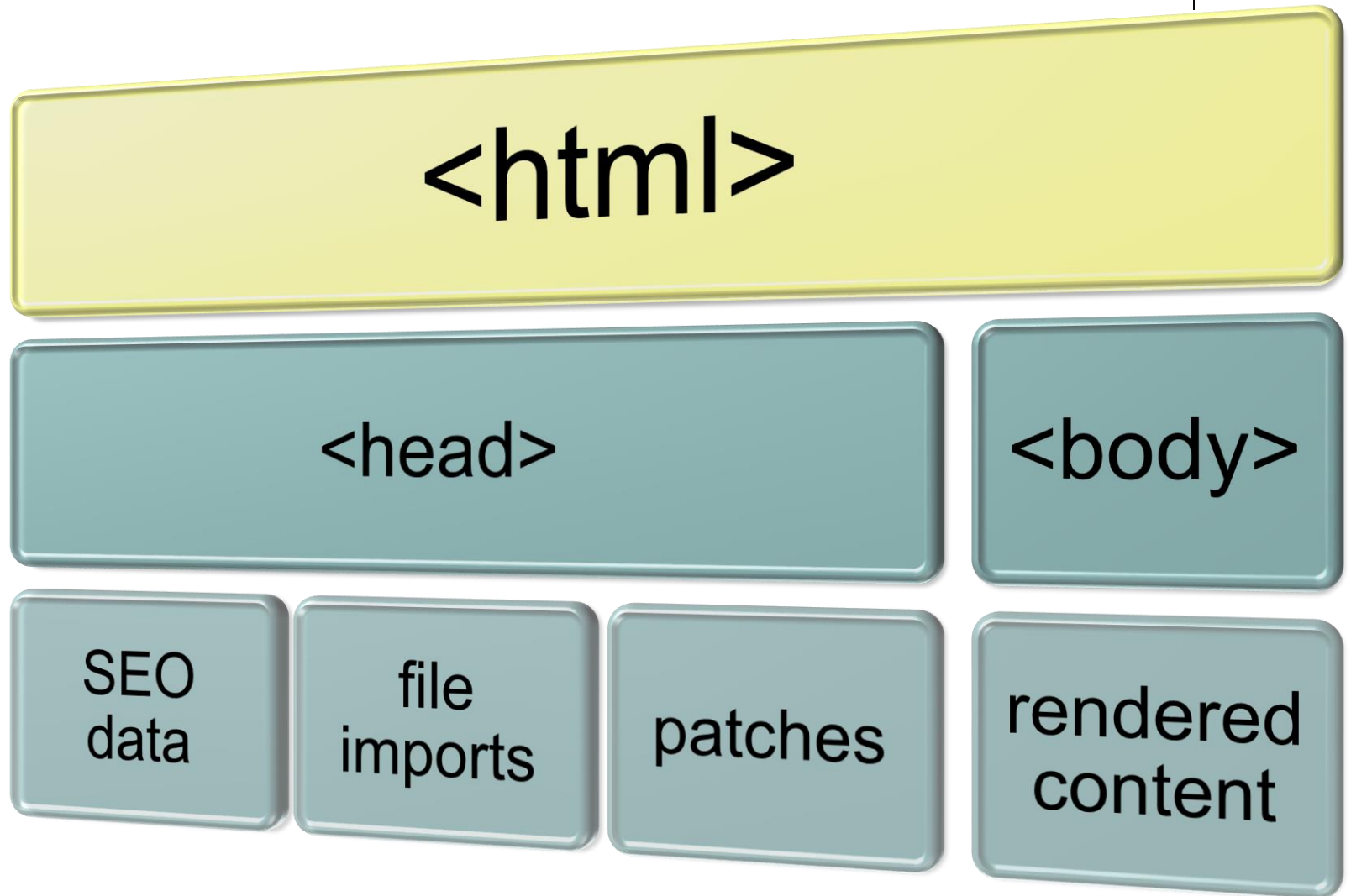
<html>

Web page file structure



- !DOCTYPE
 - <!DOCTYPE html> - HTML5 declaration
- <html>
 - <html lang="en"> ... </html>
 - root element

Web page file structure



<html> element

- The ancestor element to all elements
 - also called the root element
- <head>
 - <meta charset="utf-8">
 - <title>page title</title>
- <body>
 - contains all visible elements



```
<!DOCTYPE  
<HTML>  
<HEAD>  
<TITLE>RA  
<LINK REV  
<META NAM
```

<head>

- The <head> element contains metadata
 - <title> - shows body text in the browser's "chrome"
 - <link> - tells browser to load another file
 - CSS files
 - Favicon files (unneeded, place png in domain root)
 - Examples
 - `<link rel="stylesheet" type="text/css" href="css/theme.css" />`
 - `<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/3.0.3/normalize.min.css" />`



<head> <meta>



- Extra metadata can be added but does not affect SEO
- Examples
 - `<meta name="description" content="Free Web tutorials" />`
 - `<meta name="keywords" content="HTML,CSS,JavaScript" />`
 - `<meta name="author" content="Ståle Refsnes" />`
 - `<meta name="revised" content="Kai Refsnes, 13/01/2011" />`
 - `<meta charset="UTF-8" />`
- Delay for redirect
 - `<meta http-equiv="refresh" content="5; page.htm" />`

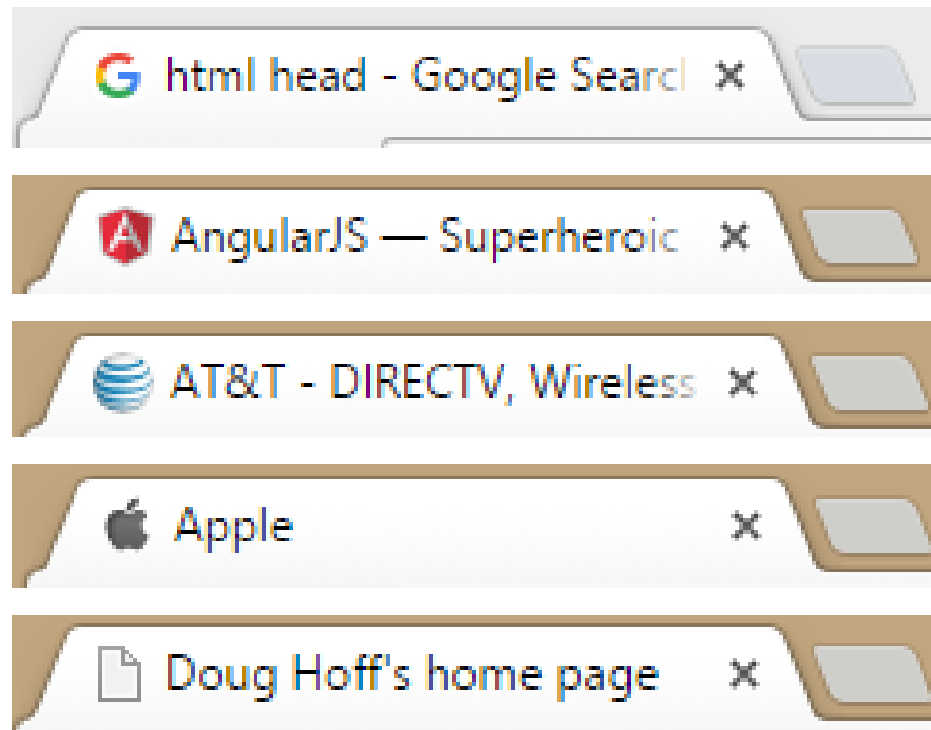


Exercise

- Create a default html:5 page with Emmet in VS Code.
- Create a page that automatically sends you to Google after five seconds.

<head> favicon

- <link rel="icon" href="anyname.ico" >
- Use an online icon editor to create one.
 - <http://www.xiconeditor.com/> - 16x16





Exercise

- Create a 16x16 favicon with your initials on it and show it on a page

<head> updates



- Mobile friendly
 - `<meta name="viewport" content="width=device-width">`
- JavaScript HTML5 shiv for IE
 - allows <IE8 versions to recognize HTML5 elements and style them.
 - Use local version
 - <https://github.com/aFarkas/html5shiv>
 - download ZIP and use minified version if needed
 - do I need to still support IE8?

more info on <head>

- <https://html5boilerplate.com/>
- Read html docs



Comments



- `<!--` this is a comment that will show in the source code `-->`
- use comments to start major sections of code
- Visible to anyone in the browser source code



Exercise


- Create an html template called `_template.html` to use for all other pages.
- Update it throughout the class for your improvements you learn and like.



HTML blocks and inline

<body>

- The body element contains all of the visible code you see in the browser
- It may also contain JavaScript



```
<!DOCTYPE html>
<html>
  <head>
    .
  </head>
  <body>
    .
    .
  </body>
</html>
```

Rendering text



- Source code is editable in an editor
- Rendering source code is controlled by the browser rendering engine
 - Webkit is used for Chrome, Android, Safari, Silk, ...
- Multiple spaces are rendered as one space
 - 'normalized'



Exercise

- Demonstrate that no matter how much white space you have on a page, it will normalize.
- Look at the source code.

Block elements



- **Next element always renders on new line**
 - i.e. a word processing paragraph of text
- Has margins and padding space style
- Expands naturally to fill its available width
 - word wrap

Block elements - text



- Usually holds text and other inline elements
- Tag names
 - **p**
 - **h1, h2, h3, h4, h5, h6**
 - **ul + li**
 - **ol + li**
 - **dl + dt + dd**
 - **blockquote**

Block level elements - layout



- Used for containers of other blocks and inline elements.
- Provides styling scope.
- Tag names
 - **div**
 - section, article, aside, main, header, footer, nav
 - table + tr + td
- Graphic elements
 - hr (horizontal rule)

Heading elements



- **h1, h2, h3**, h4, h5, h6
 - Use only h1, h2, and possibly h3.
 - Use others for special styles if necessary.
 - h1 is used at the top of the page and frequently contains the same text as the title of the page.

break



- `
` or `
`
- Creates a block element with no content
- A new line
- Used when no spacing is needed between lines
- Try not to use it and use CSS instead



Exercise

- Create a page of misc. block elements
 - header
 - h1
 - several paragraphs
 - ul
 - ol



Exercise

- Create a page of misc. block layout elements
 - header
 - table
 - footer

Lists



- `` Unordered lists
 - same as a bullet list
 - used for menus when styled
- `` Ordered lists
 - same as a numbered outline
 - attributes: reversed, number=5 (start at), type=1/A/a/I/i
- `` the list item
 - common for menus also
 - attributes: value=5 (start at)
- <http://css.maxdesign.com.au/listamatic/>

Definition lists



- `<dl>` Definition list
 - `<dt>` **Definition term**
 - `<dd>` Definition data
- Used for questions and answers, FAQs
- The dd is indented by default

Exercise

- Create a page of lists
 - ordered
 - unordered
 - definition



Inline elements

- Contain body and other inline elements
- No top or bottom space or new lines added
- Flow as the page width changes
- A box that acts like text, bookends.

SEI MALESUADA FAMES AG TURPI
LUM **INLINE ELEMENT** VITAE, U
MODD SIT AMET ANTE DONEYCEI

Inline elements - common

- **span** - used to style text in a block with CSS
- **img** - requests and renders an image file



Inline elements - semantic



- **abbr**
 - provides an expansion of an abbreviation
 - `<abbr title='Overland Park Convention Center'>OPCC</abbr>`
- **time**
 - provides a machine readable time to JavaScript
 - `<time datetime='2013-06-15'>June 15, 2013</time>`
- **dfn**
 - marks a definition
 - `<dfn title="The web's markup language to provide meaning and structure to text.">HTML</dfn>`

Inline elements - styled



- Not used much without CSS
 - **i, em** – italicized, doesn't look good
 - **b, strong** – bold
 - **code** – to display code – not used much
 - **sub, sup** – subscript, superscript
- Use as alternate tags to style in CSS in place of classes



Exercise

- Create a page of inline elements
 - b, i
 - strong, em
 - sup, sub
 - img

Using fake text

- Also called greeked text

- Marked up text snippets

- <http://html-ipsu.com/>

- Emmet - better

- lorem

- lorem*5



Lorem Ipsum

"Consectetur adipiscing elit" "Eiam venenatis tortor sed lectus."

Lorem Ipsum Pulvis Porta Auctor

Lorem Ipsum

Lorem ipsum do lor sitamet, consectetur adipi scing elit. Duis tellus. D onec ante dolor, lacula nec, g ravidu ac, curtus in, er os. Mauris vestibulum, tella ere gestu ullamcorper, purus nibh vehicula sem, eu eges. Eu ante nisl non justo. Fu ace tincidunt, lorem nax dapib us. consectetur, leo orci mollis ipsum, eget suscip iteros. purus in ante.

Atipsum vitae eu. tacinia tincidunt. Maecenas e litorci, grvida ut, mole stie non, venenatis vel, lorens. Sed tacinia. Suspendisse potenti. Sed ultricies. curtus. lectus. In id magna sitamet nibh suscipit euismo d. Integer enim. Donec sapien ante, accumsan ut, sodales. conmodo, auctor quis, lacus. Maecenas a elit tacinia urna pos uere sodales. Curabitur pede pede, molestie id, blandit vitae, varius. ac, purus. Maur is at ipsum vitae est tacinia tincidunt. Maecenas. elit orci, grvida ut, molestie non, venenatis vel, lorens. Sed tacinia. Suspendisse potenti. Sed ultricies. curtus. lectus. In id magna sitamet nibh suscipit euismo d. Integer enim. Donec sapien ante, accumsan ut, sodales. conmodo, auctor quis, lacus. Maecenas a elit tacinia urna posuere sodales. Curabitur pede pede, molestie id, blandit vitae, varius. ac, purus.

Idorbi dictum. Vestibulum adipiscing pulvinar quam. In aliquam rhoncus. sem. In nisl erat, sodales. eget, pretium interdum, malesuada ac, augue. Aliquam sollicitudin, massa ut vestibulum posuere, massa arcu. elementum purus, eget veh icula lorem metus. vel libero. Sed in duis id lectus. conmodo elementum. Eiam rhoncus. tortor. Proin a lorens. Ut nec. ve llt. Quisque varius. Proin nonum my justo dictum sapien tincidunt lacula. Duis lobortis. pellente eque risus. Aenean ut tortor. Imperdiet dolor scelerisque bib endum. Fusce metus. nibh, adipiscing id, ullamcorper at, con. equata, nulla.

Phasellus. orci. E tiam tempor elit auctor magna. Nullam nibh velit, vestibulum ut, elellend non, pulvinar eget, an in. Clusa. aptent tacit soci orqui ad litora torquent per con ubia nostra, per inceptos. hyeme nascos. Integer velit mauris, co nvollla. a congue sed, pla cersat id, odio. Eiam venenatis. tortor sed lectus. Nulla non orci. In egestas. portitor quam. Duis. nec diam eget nibh mattis. tempus. Curabitur accumsan pede id odio. Hunc vitae libero. Aenean condimentum diam et turpis. Vestibulum non risus. Ut consectetur grvida elit. Aenean est tunc, varius. se d, aliquam eu, fugiat sitamet, metus. Sed venenatis. odio id eros.

Phasellus. place rat purus. vel mi. In hac. habitas. se platea dictumst. Donec. aliquam porta odio. Ut facilisis. Donec ornare ipsum urna massa. In tellus. tellus, imperdiet ac, ac curtus an at, aliquam vitae, velit.

Lorem Ipsum
Dolor
Sed
Consectetur
Adipiscing
Ei
Sed de Glom ed
McMort et Labors
Adip
Egestas
Yllo
Yllo
Laculis
Cursus
Lectus
Yllo
Tellus
Pellentes
Accumsan
Egestas
Torquent
Phasellus
Tempor
Auctor

Common attributes - id



- **id='uniqueName'**
- used to provide a way to talk to the specific part of the page.
- always good to put on any element of interest
- provides a hook for JavaScript programming
- `<p id='author'>Author name</p>`

Common attributes - style



- **style**='rule:value'
- Adds CSS rules to the element directly
- Good for temporary use
- CSS should be promoted to the internal stylesheet or an external stylesheet
 - to provide formatting without a stylesheet
 - more of a hack, don't use if possible
 - `<p style='color:red; font-size:110%>Big red text<p>`

Common attributes - class



- **class='reusableTrait'**
- provides a way to group elements for a selection
- grouped selections are used in
 - **CSS styling often**
 - JavaScript actions sometimes
 - provide an area to style using CSS
 - can reused, can use multiple style names
 - `< p class='bright-red font110'>Big red text</p>`

Common attributes - other



- title
 - adds a tooltip text to the element on hover
 - `<abbr title='Overland Park Convention Center'>OPCC</abbr>`
- onclick, onhover, on...
 - inserts JavaScript for this element. Old style. Don't use.
- name
 - for form inputs
 - provides a key to associate the value with from this element

Character entities



- **syntax** - `&identifier;`
- non-breaking space - ` `
- copyright - `©`
- straight quotes
 - `'`; `"`
- curly quotes
 - `‘`; `’`
 - `“`; `”`
- trademark/registered - `™`; `®`
- cent - `¢`
- accents - `é`
- ampersand - `&`
(not required in HTML5)
- em dash - `—`
- bullet - `•`

Character entities

- What went wrong here? (browsers have fixed this!)



Unicode & numeric entities

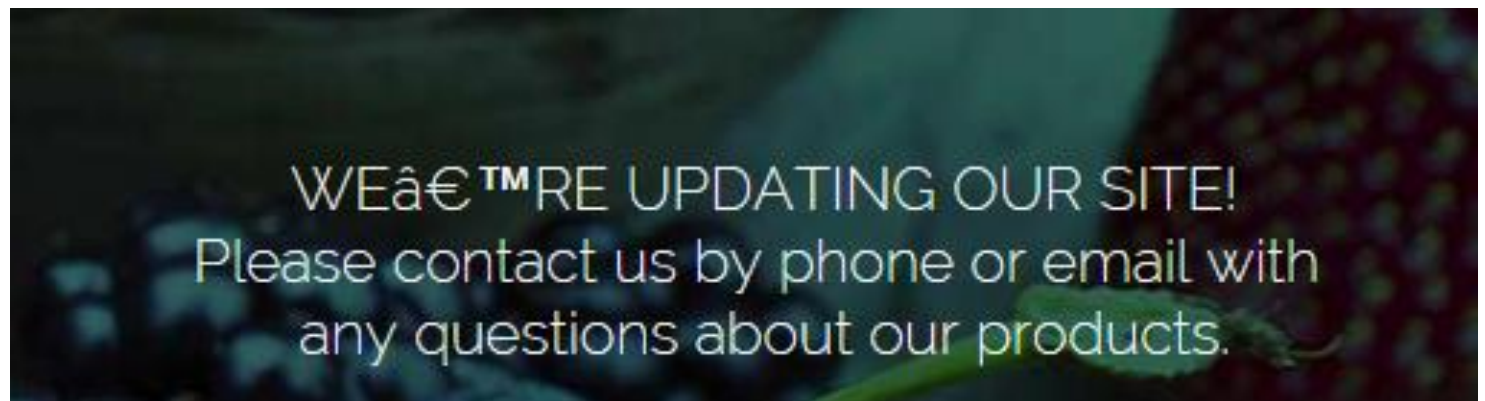
- Unicode charts
 - <http://unicode.org/charts/>
 - English = Latin character set
- &#x<Unicode hex number>;
&#<Unicode decimal number>;
 - Euro quotes «
» « »
- &what; - searchable Unicode names
 - <http://amp-what.com/>

quote			
Help or click to try: quote chess > >> / ▢ w			
			
<code>&quot;</code> ; <code>&#34;</code> ; <i>quotation mark - apl quote - double quote - quote literal mark - double-clitch</i>	<code>&apos;</code> ; <code>&#39;</code> ; <i>apostrophe - apostrophe-quote - apostrophe - single quote - quote - prime</i>	<code>&Diacritical</code> ; <code>&#96;</code> ; <i>grave accent - spacing grave - backquote - left quote - left single quote - open</i>	<code>&lsquo;</code> ; <code>&#8216;</code> ; <i>left single quotation mark - single turned comma quotation mark - quote -</i>
			
<code>&#9054;</code> ; <i>apl functional symbol quote quad - miscellaneous technical</i>	<code>&laquo;</code> ; <code>&#171;</code> ; <i>left-pointing double angle quotation mark - left pointing guillemet - latin-1</i>	<code>&raquo;</code> ; <code>&#187;</code> ; <i>right-pointing double angle quotation mark - right pointing guillemet - latin-1</i>	<code>&horbar;</code> ; <code>&#8213;</code> ; <i>horizontal bar - quotation dash - general punctuation</i>
			
<code>&#10077;</code> ; <i>heavy double turned comma quotation mark ornament - dingbats</i>	<code>&#10078;</code> ; <i>heavy double comma quotation mark ornament - dingbats</i>	<code>&#10079;</code> ; <i>heavy low single comma quotation mark ornament - dingbats</i>	<code>&#10080;</code> ; <i>heavy low double comma quotation mark ornament - dingbats</i>

Character encoding



- Some character sets do not match Unicode numbering
- Culprits include
 - database character encoding schemes
 - old web pages set to ISO-8859-1 and others
 - pasted text from old Word documents
- Symptoms: odd characters or empty boxes for a character





Exercise

- Create a few character and numeric entities.

½ off day!

“ This is quoted. ”

As a valued MyPanera® member Lorem ipsum
id cupiditate placeat, iusto vel sed, re
numquam animi, laborum rerum poss



Exercise

- The Black Goose blog page (ex 5-3 Robbins) – mark up the text with semantic elements.
 - text to use: **basics/black-goose-blog.txt**
 - Use elements of header, h1, h2, h3, nav, ul, ol, li, article, section, aside, p, date, abbr, dl, dt, dd, dfn, cite, b, q, footer and any others you like.



Links

<a> Anchors (links)



- what does it do? – it makes a request
- use an id on the page (named)
 - `Linked text`
 - `<div></div>`
- `href='email:email@doughoff.com'`
 - not recommended
 - email `<at>` doughoff.com

href attribute values - external



- starts with protocol (http://, https://, etc.)
- no protocol (//...) – uses current browser protocol, not recommended.

href attribute values - internal



- Relative
 - start with **directory/file** to reference subdirectory files
 - same directory: use no folder or ./
 - parent directory: start with ../file or ../directory . Use multiple times if necessary ../../../ etc.
 - best method
- Absolute
 - start with a slash (/articles/frontpage.html)
 - the root of the web site
 - only for hosted web sites
 - start with tilde (~) to represent the ASP.NET site root

Link attributes



- target =
 - _blank - opens in a new tab
- <anyElement id= "uniquePlaceOnPage">
 - use to indicate a position to go to with
 - also link from another page with



Exercise






- Add some links in a list to a page to take you to:
 - CNN
 - Google
 - Kansas City Star
 - Your home page





<iframe>

- Used to insert page from another web site
- Embed
 - YouTube, Vimeo, Google Maps, etc.
- Attributes
 - src - the URL
 - height - in pixels
 - width - in pixels
 - scrolling, frameborder - don't use in HTML5
 - seamless - HTML5 no scrollbars – not supported



Google Maps

-  Earth
-  Traffic
-  Transit
-  Bicycling
-  Terrain

-  My Maps
-  Your contributions
-  [Share or embed map](#)
-  Print



Images



Images



- HTML initially loads the list of image links
- Each link to an image requires a separate download unless cached.
 - each download is called a **hit**
 - each set of downloads for a page is called a page view



- Attributes

- **src** - the URL to the image which can be on your site or another site
- **alt** - the text description that Google wants to know about. **Always** use it. Shows when image is not there.
- **title** - additional info that will appear in a tooltip when the cursor hovers over the image
- **height, width** - helps browser render the page but CSS is preferred. Using only one will cause the other to be set proportionately.
- **align** - use CSS instead

 Images



- Formats
 - gif, jpeg/jpg, ico, png
 - svg – not an image
-
 - adjust width & height
 - adjust only one to keep size proportional
 - Use attribute - old style, don't use px because of RWD
 - width="70%"
 - Use CSS
 - style="width : 200px;" or style="width : 70%;"

Path navigation review

- ``
- ``
- ``
- ``

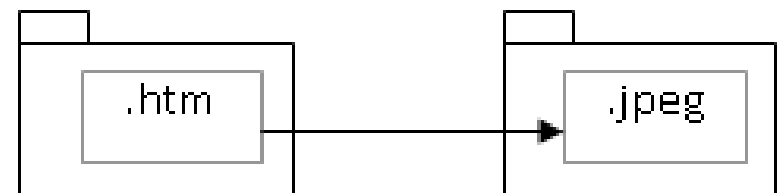
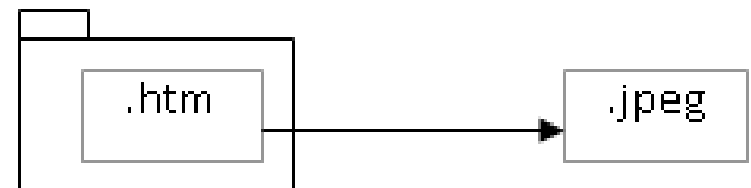
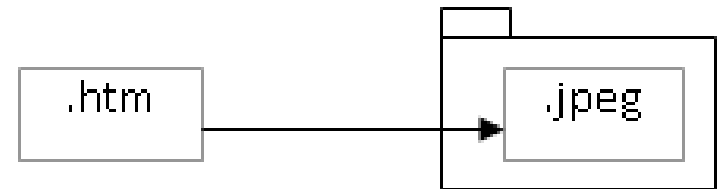
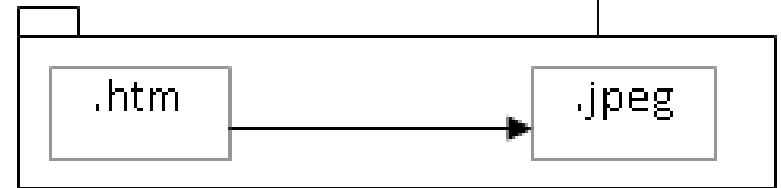


Image maps

- Lack of support in tools
- Usage is for specific purposes
 - UI navigation on maps, architectural plans
- Tools
 - <http://Image-maps.com>



Using PNG



- 8-bit, 24-bit and 32-bit (24 bit with transparency)
- Always keep a copy of the original size.
- Resize image for what is necessary on the site
 - reduces download time

Image sources



- Flickr - creative copyright search
- Google Images
 - Use image size and usage search
 - Always pick the original image to copy – not thumbnail
- Creative Copyright usage -
<http://search.creativecommons.org/>
- \$\$\$ stock photos
- Don't copy other people's images unless
 - You link to them
 - You modify them significantly

Image process



- Find image. Keep it as an original and don't ever change it.
- Modify image to the size you plan to use
- Modify image to the resolution you need
 - 100 ppi (pixels per inch) is likely good enough
 - Users with tablets want to zoom in so make it larger if it makes sense.

Other image sources



- Vector images
 - SVG is becoming more popular (Chart widget)
- Animated GIFs
 - a GIF with multiple frames to create a very short movie
- Transparent GIFs
 - better to use a PNG with transparency. Great to not have an image background color. Remember to save to a compatible background color. White is default which makes white background web sites easier to do.

Other image sources



- Flash
 - 2015 most browsers don't support
- Vimeo and YouTube
 - use and embed players with iframes

Resources - images



- Free images
 - Noupe - <http://www.noupe.com/inspiration/photography/the-ultimate-list-of-completely-free-images-43-handpicked-services-89011.html>
- License searchable
 - Google – search with Search Tools / Usage Rights
 - make sure to view the image and then get the URL, not just right click to get the reduced size of the original
 - Flickr - <https://www.flickr.com/commons>

Resources – editors



- Paint.NET
- GIMP
- Adobe Photoshop
 - <http://www.photoshop.com/tools>
- Pixlr
 - <https://pixlr.com/>
- Snapseed - mobile

Resources - tools



- Canva - <https://www.canva.com/>
 - limited but good designs and templates
- Pablo - <https://buffer.com/pablo>
- Befunky - <https://www.befunky.com/features/photo-editor/>
- PicMonkey - <http://www.picmonkey.com/>
- Easelly - <http://www.easel.ly/> infographics
- Social Image Resizer Tool
 - <http://www.internetmarketingninjas.com/seo-tools/favicon-generator-crop-images/>
- Recite - <http://www.recitethis.com/> quotes – gone?
- Fotor - <http://www.fotor.com/features/collage.html> collages
- Dynamicdrive.com – icon conversion
<http://tools.dynamicdrive.com/favicon/>

Font Awesome



- uses CSS
- create link from <https://cdnjs.com/> and put in <head>
 - <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css" />
- Examples
 - <i class='fa fa-camera-retro'></i>
 - <i class='fa fa-pencil-square fa-3x'></i>
 - <i class='fa fa-pencil fa-5x fa-spin' style='color:red'></i>
 - <i class='fa fa-home fa-3x' style='color:green'></i>
- Search for other icons - <http://fontawesome.io/icons/>



Future



- WebP image format
 - supports lossy and lossless compression, as well as animation and alpha transparency
 - only supported by Chrome for now



Exercise

- Create a page and put these on it
 - image from Google Images sized to 300px high
 - image from Flickr sized to 300px high
 - several font icons from Font Awesome
 - a font icon (fa-home) wrapped in a link to make it clickable



Exercise

- Put a home icon on your images page
- Update your home page with links to most of your pages.
- Add the home icon to your template.
- Extra: Go back and add the home icon to your other pages.

Tables

[illegible]

<table>



- Tabular data display
- Container (parent) <table> **required**
 - table header <thead> optional
 - table row <tr>
 - table heading <th>
 - table body <tbody> optional
 - table row <tr> **required**
 - table data (cell) <td> **required**
 - table footer <tfoot> optional
 - table row <tr>

Structuring elements for rows

- `<thead>` table header contains rows with all `<th>`
- `<tbody>` table body contains rows beneath heading
- `<tfoot>` table footer contains rows beneath body



Special `<td>` or `<tr>`



- Column headers
 - Change the first row's `<td>` to `<th>`
 - Default style is bold and centered
- Table title/caption
 - Change a `<tr>` into a `<caption>` without any `<tr>` and it will span across table on top.

<table>



- td attributes
 - **colspan=2** - allows you to combine columns for a cell
 - **rowspan=2** - allows you to combine rows for a cell

Heading cell.	
First cell.	Second cell.
First cell.	Second cell.

First cell. First row.	Second cell. First row.
	First cell. Second row.

Table style

- `table, th, td {
 border: 1px solid black;
 border-collapse: collapse;
}
th, td {
 padding: 15px;
}`



Table style rules



- empty-cells: show, hide, inherit
 - whether to show borders or not
- border-spacing:
 - distance between adjacent cells
 - two values can spec horizontal and vertical
- border-collapse: collapse
 - borders made into single border when possible
- border-collapse: separate
 - borders detached and uses spacing and empty-cells

<table> vs. ? for layout elements



- <table>
 - longer load times – display only after all table content has loaded
 - transparent images needed, adds time
 - maintenance is hard
 - accessibility issues
 - use with tabular data

<table> vs. ? for layout elements



- <div>
 - a better solution for layouts
 - fixed or relative width less than page
 - float right or left which allows another div to sit side by side
- flexbox – uses block elements and CSS
- grid – best simple modern choice

Accessibility



- For text readers
- Web Accessibility Initiative, Accessible Rich Internet Applications, Web Content Accessibility Guidelines
- <https://w3c.github.io/aria-practices/examples/landmarks/>
- WCAG 2.0 AA Is the New Accessibility Standard for Federal Agency Websites (2017)
 - <http://www.lexology.com/library/detail.aspx?g=e627e1be-eddf-44cb-9459-c8f72de48334>



Exercise

- Create several tables with some data
 - 1x1 with multiple elements
 - 2x2
 - 4x4 with a full width row on top
- Add thead, tbody, and tfoot on the 4x4 table

Exercise

layouts/ table-tv-schedule-complete.html



- Start with **layouts/tv-schedule.html**
- Add styles. Use images for networks.



WEDNESDAY	8pm	8:30pm	9pm	9:30pm	10pm	10:30pm
	The Middle	Better With You	Modern Family	Cougar Town	The Whole Truth	
	Survivor: Nicaragua		Criminal Minds		The Defenders	
	America's Next Top Model		Hellcats		Local	
	Hell's Kitchen				Local	



Search Engine Optimization

SEO



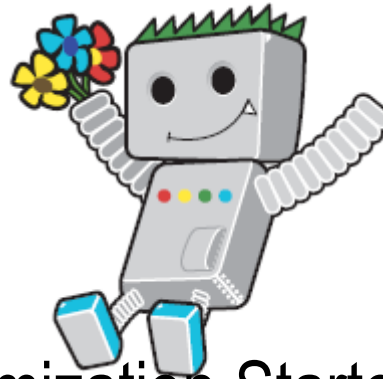
- SEO is trying to help your site appear as a search result closer to the top.
- On-page techniques
 - choosing and using keywords on pages, no use using meta tags
 - using alt tags in images
 - using good page titles
- Off-page techniques
 - get as many related sites to link to you as you can with good link text

SEO



- `If you want to go to Kansas Lottery then click here!`
- `Visit the Kansas Lottery site!`

SEO

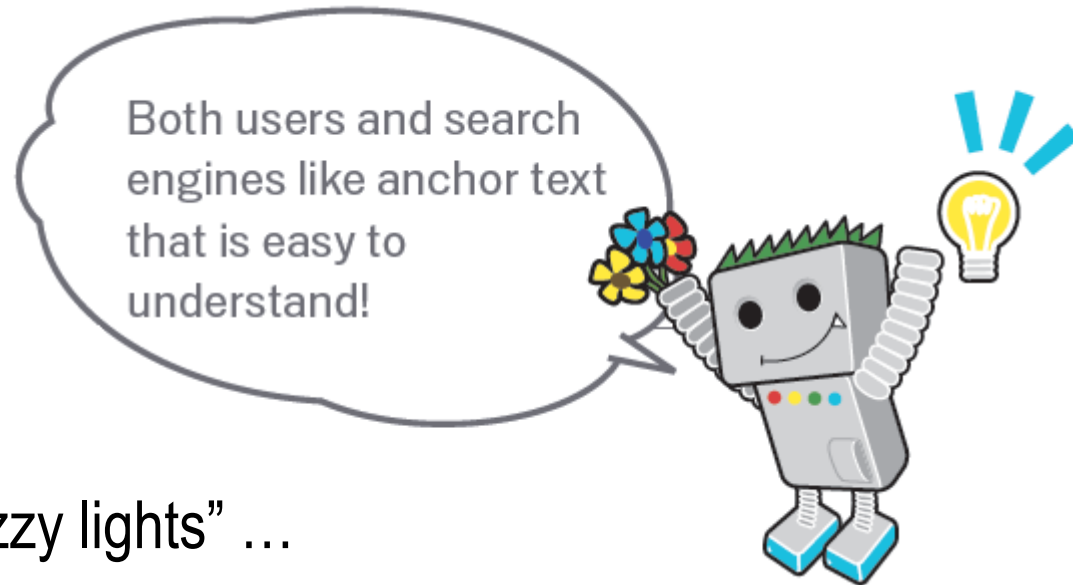


- Google Search Engine Optimization Starter Guide
 - http://static.googleusercontent.com/external_content/untrusted_dlcp/www.google.com/en/us/webmasters/docs/search-engine-optimization-starter-guide.pdf
- Main points
 - Well-named titles
 - `<meta name="description" content="???">`
 - Well-named URL paths
 - General to specific folder structure
 - Sitemap

SEO



- Google's main tips
 - Useful 404 page
 - Quality content
 - Better anchor text – [click here](#)
 - Image text `<img alt="fuzzy lights" ...`
 - Headings as outline main points
 - Use robots.txt
 - Links – nofollow for unmanaged content
 - Mobile sites are special, redirects
 - Site promotion



Exercise: SEO



- Check your site
 - Domain Authority
 - Page Authority
 - Spam score
- <https://moz.com/researchtools/ose>

Know your browsers



- Google Analytics
 - free
 - <http://www.google.com/analytics/>
 - You add a piece of tracking code to every page you want tracked.
 - Google keeps track of the visitors and lets you run reports on it.
- Web hosting analytic packages
 - most have some web log analysis program for you



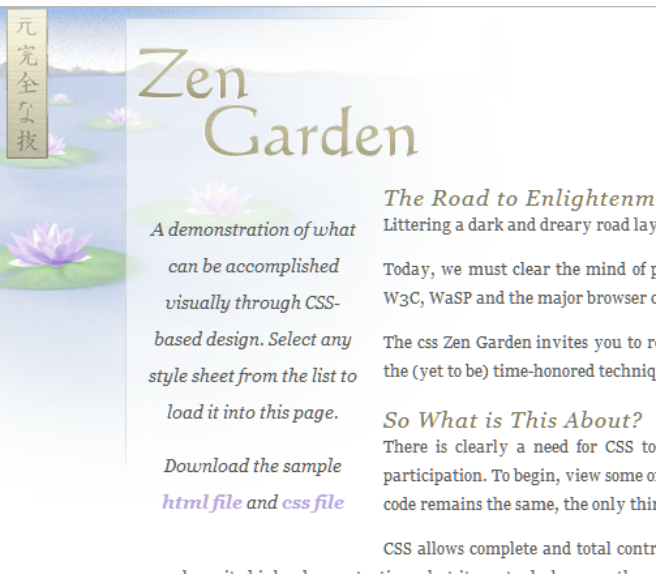
CSS intro



CSS Zen Garden



- <http://www.csszengarden.com/>
- One html document
 - hundreds of CSS/graphics variations



CSS fundamentals



- a different language syntax completely!
 - showing age
 - improvements - LESS/SASS, a scripting language that outputs CSS
- Cascading Style Sheets
 - file with code usually ends with .css and is called a style sheet or CSS file.

Terms



- **selector** – the rule declaration that matches the html for the style
 - **h1** { color: navy; }
- **declaration (rule)** – a property and value pair
 - h1 { **color: navy;** }
- **rule set** – all the declarations in the block after the selector

Terms



- **property** – the style portion that is affected
 - `h1 { color: navy; }`
- **value** – how to change the property
 - `h1 { color: navy; }`
- **inheritance** – (not OO) application of style to child elements either by default or explicit

Comments



- `/*` the only kind of comment for CSS `*/`
 - using `//` will stop CSS from working
- Always start a CSS sheet with a comment that has
 - name of CSS sheet
 - your name
 - date last edited (if it matters)

Reference CSS in <head>



- external stylesheet - **<link>** to external file
 - `<link rel="stylesheet" href="http://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.css">`
 - `<link rel="stylesheet" href="css/sitename.css">`
- use multiple links
 - order is important, later styles will add to and override earlier styles

Reference CSS in <head>



- internal stylesheet - **<style>** element for this page
 - Provide styles to one page only.
 - Best method for speed
 - Refactor
 - Rename class & move to external file.
 - Will add to and override external stylesheets

```
<head>
  <title>My First Page</title>
  <style type="text/css">
    .style1
    {
      color: #3399FF;
    }
  </style>
</head>
<body>
```

```
<p style="font-weight: 700; color: #3399FF;">
  Something <span class="style1">
</p>
```

Inline <style> attribute



- Change the font/color of **one** element on one page.
- No selector, uses a style attribute
- Attaching method:
 - `<h1 style="font-family: Arial, Helvetica, sans-serif">`
 - Title of page
 - `</h1>`
- Not recommended! Good for testing.
- Refactor
 - Name with class
& move to external file.

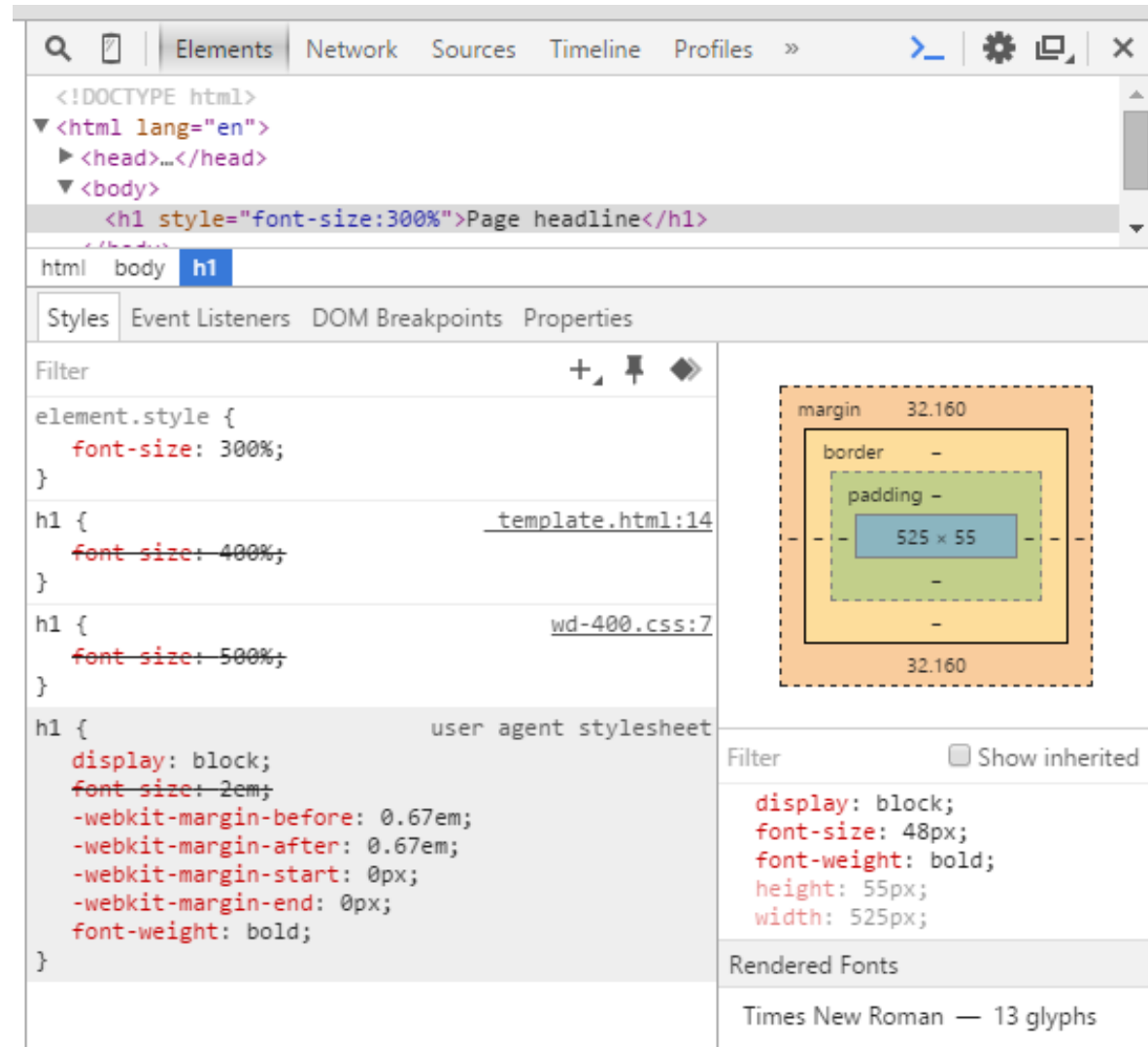
CSS precedence (order of application)



- Each Inherited style from parent elements and then specific style
 - Browser defaults (user agent stylesheet in Dev Tools)
 - External files in order
 - Internal embedded (<style>...</style>)
 - Internal inline attribute (style='...')
 - **!important** styles declared anywhere
- actual precedence includes other selectors

Chrome styles inspector

- Click on element
- Styles applied from bottom to top
- Strikeover rules are overridden
- user agent = browser
- Calculated styles under box model



Demo



- Load a site
- View the Elements tab
 - Dig into the DOM – open the triangles, Inspect
 - Look at the styles hierarchy
 - Look at the box model and results of rules
- View the Console – any errors?
- View the Network tab
 - Reload the page to see timings
- View the Sources tab
 - Find the images and CSS files



Exercise

- Create a stylesheet with the name the same as your web site and add to the css folder
- Add a link to it in your template.
- Create a page that has three access methods to CSS on it using this file for one of them.
- Inspect with Chrome one rule at a time.

CSS and browsers



- Browsers have defaults.
 - We don't like them.
- Best practice: use a reset style sheet to 'zero out' all browser settings

```
html {  
    margin:0;  
    padding:0;  
    border:0;  
}
```

```
body, div, span, object, iframe,  
h1, h2, h3, h4, h5, h6, p, blockquote, pre,  
a, abbr, acronym, address, code,  
del, dfn, em, img, q, dl, dt, dd, ol, ul, li,  
fieldset, form, label, legend,  
table, caption, tbody, tfoot, thead, tr, th, td,  
article, aside, dialog, figure, footer, header,  
hgroup, nav, section {  
    margin: 0;  
    padding: 0;  
    border: 0;  
    font-size: 100%;  
    font: inherit;  
    vertical-align: baseline;
```

Resources - Reset stylesheets



- CSS stylesheets that remove defaults
 - Eric Meyer's Reset - <http://meyerweb.com/eric/tools/css/reset/>
 - Nicolas Gallagher and Jonathan Neal's Normalize - <http://necolas.github.com/normalize.css/>
- Find a link for these at <http://cdnjs.com>

Resources - stylesheets



- HTML5boilerplate
 - <http://html5boilerplate.com/>
 - Normalize.css, good docs for an html template
 - Contains html5shiv and other fixes for HTML5
- Blueprint
 - <http://www.blueprintcss.org/>
 - Typography, reset, print
- CDNJS
 - <https://cdnjs.com/>
 - A content delivery network for JavaScript and CSS

CSS and browsers



- To make sure what your page looks like, test in each browser you have.
- To really make sure what your page looks like, use a service to test in other browsers
 - <http://BrowserCam.com>
 - <http://BrowserShots.org>
 - <http://CrossBrowserTesting.com>



Exercise

- Add the Normalize.css to your styles.html page
- Add the Normalize.css stylesheet to your template before your site css file.



CSS selectors

selector property value declaration end

↑ ↑ ↑

declaration start property/value separator declaration separator

```
a { background-color: yellow; }
```

Matching element to CSS selector



Type	HTML	CSS
Tag name	<p>	p { CSS }
Class name	<p class="className className2">	.className { CSS }
ID name	<p id="idName">	#idName { CSS }

Selectors - elements



- Any valid html element tag name
- Examples
 - **body** {
 margin: 0;
}
 - **a** {
 color: #00e;
}

HTML Attributes



- Common optional attributes for all elements are
 - title="descriptive text" - tooltip
 - style="inline style description" - inline CSS
 - id="unique identifier" - for a few CSS and most JS references
 - class="class names" – best for reusable CSS
- Events are common and valid, but don't use inline like this
 - onclick="javascript here"

Selectors – classes



- Applies to any value in a class attribute
 - `<p class="bigred">`
- CSS requires a dot before the class name
- Reusable throughout a page and site
- Examples
 - `.bigred { color: red; font-size: 150%; }`
 - `.ui-helper-hidden { display: none; }`
 - `.pull-quote { padding: 6px 0; line-height: 20px; text-transform: uppercase; font-size: 12px; }`

Selectors - IDs



- Applies to any unique id attribute
 - `<div id="content-wrapper">`
- CSS requires a pound sign before the ID name
- Can only be used once per page in a site
- Examples
 - `#content-wrapper { background: none; }`
 - `#downloadLegacyTheme { color: #666; }`
 - `#picker { position: absolute; left: 0; top: auto; width: 200px; }`

Selectors - combining



- Element names, classes and (not necessary) ids
- `div.posting { ... }`
 - `<div class='posting' ...`
- `div.posting.mainoffice { ... }`
 - `<div class='posting mainoffice' ...`
- `div#main { ... }`
 - `<div id='main'...`

Selectors - Pseudo Classes



- Examples for link states (use this order):
 - **a:link** – same as a
 - remove underline on links except when being hovered over
 - **a:visited** { color: #551a8b; }
 - dim the color on links that probably won't be used again.
 - **a:hover** { outline: thin dotted; }
 - give feedback to show the link is hovered on, maybe an underline
 - use for any element
 - **a:active** { outline: 0; }
 - only when pressing on the mouse, not so useful

Selectors - Pseudo elements



- CSS uses one or two colons (one is sufficient now)
- Examples
 - `element: first-child`
 - `element: first-letter`
 - `element: only-child`
 - `element: first-of-type`
 - `p:first-of-type:first-letter`
 - `element: last-child`
 - `tr:nth-child(even)`, `tr:nth-child(3)`

Selectors - Pseudo elements



- Only use one colon, not two
 - was a requirement for some browsers
- Resources
 - <https://www.smashingmagazine.com/2016/05/an-ultimate-guide-to-css-pseudo-classes-and-pseudo-elements/>



Exercise

- Make a copy of your black-goose-blog.html
- Add different styles and try others... (css/black-goose-blog-styled.html)
 - `body { padding-left: 20px;}`
 - `ul {`
 - `font-family: 'Times New Roman', Times, serif;`
 - `font-size: 18px; font-variant: small-caps;`
 - `color: darkolivegreen; border: 5px dotted lawngreen;`
 - `width: 30%; font-weight: bold; }`
 - `h3 { color: cornflowerblue; background: lightyellow; }`
 - `h3:first-of-type { font-size: 200%; }`

Operators - grouping



- Use multiple selectors separated by a comma
- Applies rule to all
- Equivalent rules
 - `h1 { font-family: sans-serif }`
`h2 { font-family: sans-serif }`
`h3 { font-family: sans-serif }`
 - `h1, h2, h3 { font-family: sans-serif }`
- Add another rule to enhance any of them
 - `h1, h2, h3, h4 { font-family: sans-serif ; color:red; }`
 - `h3 { color: green; }`

Operators - descendant



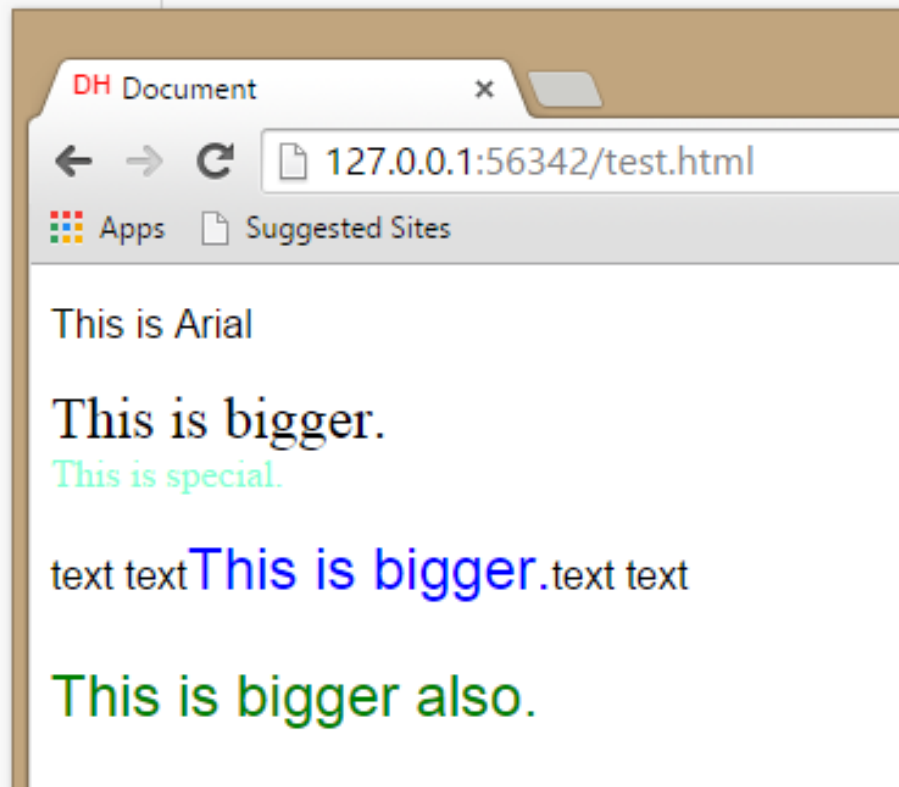
- Uses two selectors separated by space
- Applies to any descendant matching 2nd selector within the 1st selector
- Examples
 - `nav ul, nav ol { list-style: none; list-style-image: none; }`
 - `table button, table input { *overflow: auto; }`

Operators



```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Document</title>
6      <link rel="stylesheet"
7      href="css/test.css">
8  </head>
9  <body>
10     <p>This is Arial</p>
11     <section class='size110'>
12         This is bigger.
13     </section>
14     <aside id="special">This is
15     special.</aside>
16     <p>text text<span
17     class="size110">This is bigger.
18     </span>text text</p>
19     <p class="size110">
20         This is bigger also.
21     </p>
22
23 </body>
24 </html>
```

```
1  p          { font-family: Arial,
2              sans-serif;}
3  .size110    { font-size: 150%;}
4  #special    { color:aquamarine;}
5  p .size110  { color: blue;}
6  p.size110   { color: green;}
```



Selectors - attributes



- Applies to element with attribute of a defined value
- Examples
 - `input[type="checkbox"], input[type="radio"] {
 box-sizing: border-box; padding: 0;
}`
 - `button, html input[type="button"], input[type="reset"],
input[type="submit"] {
 cursor: pointer; -webkit-appearance: button;
}`

Selectors - other



- * { }
- rule set will apply to all elements
- **!Important** rule
 - include !important in the rule body
 - overrides any cascade order
 - Use in extreme cases

Resources



- Selectutorial - <http://css.maxdesign.com.au/selectutorial>

Link selectors



- a – for all links
- a:hover – only when mouse is over
- a:visited – after page has been viewed
 - clear browsing history to reset
- Useful styles
 - text-decoration: none;
 - color: yourChoice; - make visited colors darker
 - text-decoration: underline;
 - font-weight: bold;



Exercise

- Change the style of links to
 - have no underline except when hovering over them
 - use another color other than blue
 - be bold all the time



CSS text

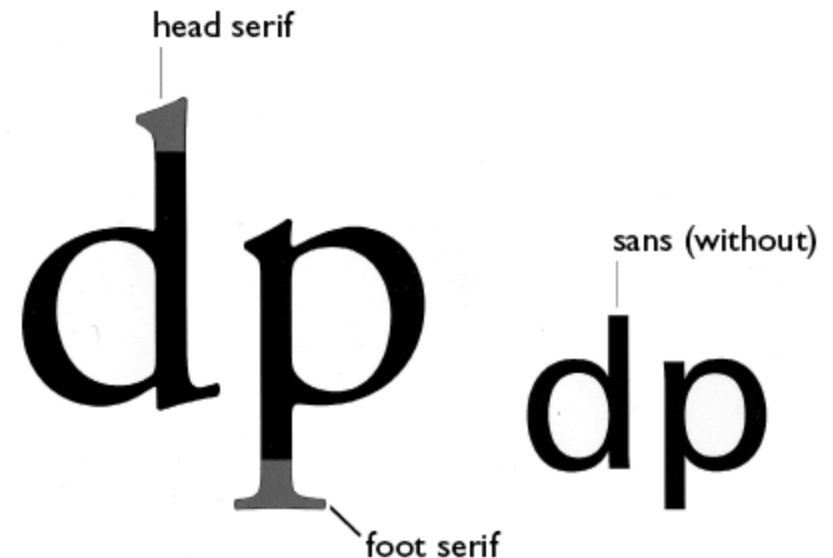
Любовь не отвергает, мило любит, любовь не забывает, любовь не презирает, не гордится, не бессиленствует, не ищет своего, не раздражается, не мыслит зла. Не радуется неправде, а сокрушается истине; Все покрывает, всему верит, всего надеется, все переносит. Любовь никогда не перестает, хотя и пророчества прекратятся, и языки умолкнут, и знания упредаются. Что бы отчасти знаешь и отчасти пророчествуешь; когда все настанет совершенное, тогда то, что отчасти, прекратится. А теперь пребывают сии три: Веры, Надежды, Любовь; но любовь из них больше. Достигайте любви...



font-family - logical



- mapped to an actual font on every machine
- serif
 - fonts with details at the ends of the strokes of the letters
 - common reading font in US
- sans-serif
 - “without” serifs, common reading font in Europe
- monospace
 - fixed-width, good for code or typewriter look



font-family



- font-family: 'serif'
 - will choose the default font of that O/S listed as serif
 - other logical fonts are sans-serif, monospace, cursive, fantasy
- font-family: 'Arial, sans-serif'
 - will choose Arial if it's on the target browser's system
 - otherwise it will fall back to the logical font
- font-family: 'Verdana, Helvetica, sans-serif'
 - will look for Verdana (PCs), then for Helvetica (Macs), then fall back

Measurement scales - fixed



- points
 - print scale, 72pt = 1 inch
 - use only for printed pages
- pixels
 - similar to points but dependent on screen resolution and is not be dependable
 - best for positioning elements exactly
 - not responsive

Measurement scales - relative



- use in preference to px
- parent-child related sizes
 - each child element is relative to the parent element size
 - based on the default size of the browser's font which is 16px
- percentages – 150%
- ems – 1.5em (1em starts at 16px for all browsers)
- rems (CSS3 – IE9+)
 - same as ems but relative to the root element size only
 - simplifies size thinking

Font styles properties



- Family: `font-family` (`font-face`: for web fonts)
- Size: `font-size`
- Weight: `font-weight`
- Style: `font-style`
- Variant: `font-variant`
- Line height - `line-height`
- Shorthand syntax: `font`: style weight variant size/line-height family,
 - Font-size and font-family are required for the shorthand syntax

font-weight, font-style



- Weight
 - light, medium, **bold**, **black**
- Style
 - normal, *italic*, ***oblique***
 - italic is usually a different font with cursive aspects
 - oblique is angling the font, like what a browser does when it doesn't have the italic font.

Font properties

- **font-variant:** small-caps
- **font-weight:** **bold** selects a thicker face in the font-family.



- **font-style:** **italic** is used only if the font is available, otherwise it's obliqued

a a a



Veggieburger Light - Lorem ipsum
eiusmod tempor incididunt ut la
quis nostrud exercitation ullamc
aute irure dolor in reprehenderi
pariatur. Excepteur sint occaec
deserunt mollit anim id est labor

Veggieburger Regular - Lorem ipsum
do eiusmod tempor incididunt ut
veniam, quis nostrud exercitation
consequat. Duis aute irure dolor
fugiat nulla pariatur. Excepteu
officia deserunt mollit anim id est

Veggieburger Bold - Lorem ipsum
eiusmod tempor incididunt ut
quis nostrud exercitation ullamc
aute irure dolor in reprehende
pariatur. Excepteur sint occaec
deserunt mollit anim id est labor

Font properties



- font-stretch – condensed, normal, expanded (no Safari)

e e **e** e **e** e e e e

condensed normal expanded

- font-size-adjust – preserving x-height on font fallback (Firefox only)

b b b

b b b

Exercise (font-styles.html)



- Create a page with internal styles of
 - font80 - font-size: 80%; line-height: 80%;
 - font110 - font-size: 110%; line-height: 70%;
 - font130 - font-size: 130%; line-height: 150%;
 - caps – font-variant: small-caps;
 - italic – font-style: italic;
 - bold – font-weight: bold;
- Create five divs of lorem text and apply combinations of the classes.

Free font service resources

- Google Fonts online - <http://www.google.com/webfonts>

The screenshot displays the Google Fonts web interface. At the top, the navigation bar includes 'Google Fonts', 'DIRECTORY' (highlighted in red), 'FEATURED', 'ABOUT', and a search icon. Below the navigation bar, a status line reads 'Viewing 217 of 808 font families'. The main content area is a grid of font family cards. Each card shows the font name, designer, a red plus icon, and a sample of text. The cards visible are: Open Sans (Steve Matteson, 10 styles) with the text 'All their equipment and instruments are alive.'; Roboto (Christian Robertson, 12 styles) with the text 'Almost before we knew it, we had left the ground.'; Lato (Łukasz Dziedzic, 10 styles) with the text 'It was going to be a lonely trip back.'; Oswald (Vernon Adams, 3 styles) with the text 'My two natures had memory in common.'; Roboto Condensed (Christian Robertson, 6 styles) with the text 'Silver mist suffused the deck of the ship.'; and Source Sans Pro (Paul D. Hunt, 12 styles) with the text 'The face of the moon was in shadow.' A 'SEE SPECIMEN' link is located between the Roboto and Roboto Condensed cards. On the right side, a sidebar contains filters: 'Categories' (Serif, Sans Serif (checked), Display, Handwriting, Monospace), 'Sorting' (Trending), 'Languages' (All Languages), 'Number of styles' (a slider), 'Thickness' (a slider), 'Slant' (a slider), and 'Width' (a slider). At the bottom right of the sidebar is a blue 'Add to Collection' button.

Google Fonts

DIRECTORY FEATURED ABOUT

Viewing 217 of 808 font families

Open Sans
Steve Matteson (10 styles)

Roboto
Christian Robertson (12 styles)

Lato
Łukasz Dziedzic (10 styles)

Oswald
Vernon Adams (3 styles)

Roboto Condensed
Christian Robertson (6 styles)

Source Sans Pro
Paul D. Hunt (12 styles)

SEE SPECIMEN

Categories

- ☐ Serif
- ☒ Sans Serif
- ☐ Display
- ☐ Handwriting
- ☐ Monospace

Sorting

Trending

Languages

All Languages

Number of styles

Thickness

Slant

Width

Add to Collection

Exercise (font-families.html)



- Add several web fonts to your site CSS file as classes.
- Use on a new page of five divs of lorem text.

Free font service resources



- Adobe Web Fonts – 500+ <https://edgewebfonts.adobe.com/>
- Paid / free
 - [Typotheque](#), [Webtype](#), [Fontdeck](#), and [Fonts.com](#)

Font download resources



- [Font Squirrel](#) download kit includes
 - TrueType, EOT, WOFF, SVG, Cufón and fonts
 - Demo.html and stylesheet.css
 - Upload your own and generate kit

Undercover | 1 Font
[View Font](#) | [View @ff Demo](#) | [Get Kit](#)

UNDERCOVER

Vanilla | 1 Font
[View Font](#) | [View @ff Demo](#) | [Get Kit](#)

Vanilla AaBb

Veggieburger | 3 Fonts
[View Font](#) | [View @ff Demo](#) | [Get Kit](#)

Veggieburger A

Web font usage



- In HTML internal style sheet
 - `<link href='http://fonts.googleapis.com/css?family=Raleway:400,900' rel='stylesheet' type='text/css'>`
 - `<style>`
 - `body { font-family: 'Raleway', sans-serif; }`
- In CSS rules
 - `@import url(http://fonts.googleapis.com/css?family=Frijole);`
 - `h1, h2, h3, h4, h5, h6 {`
 - `font-weight: normal;`
 - `font-family: "Frijole", Arial, Helvetica, sans-serif;`
 - `}`

Text styles



- **Text alignment** - text-align:
 - left, right, center, justify
- **Vertical alignment** - vertical-align:
 - baseline, sub, super, top, text-top, **middle**, bottom, text-bottom, #em, %ofLineHeight
- **Indenting** - text-indent:
 - sometimes used to place item off page with a negative indent

Text styles



- **Spacing** - letter-spacing:, word-spacing:
- **Leading** - line-height: (1.2 – 1.4 usually)
- **Casing** - text-transform:
 - uppercase, lowercase, capitalize
- **Decoration** - text-decoration:
 - none, **underline**, overline, line-through
- **Shadows** - text-shadow: hor ver blur color



Exercise

- Start a greeting card using the following CSS
 - @import url('https://fonts.googleapis.com/css?family=Yesteryear');
 - .xmas-font {font-family: 'Yesteryear', cursive; }
 - .rem6 { font-size: 6rem; }
 - .bright-color {color: orangered;}
 - .white-shadow {text-shadow: 4px 4px 8px grey }
- HTML
 - <div class='centered-content rem6 bright-color white-shadow xmas-font'>
 - <div class='bg-white-glass' >Merry Christmas 2017!!!</div>
 - </div>

:after, :before



- `div:after { content: "– a div by DH"; }`
- `div:before {content: url(/path/to/image.jpg); }`
- `ol > li:before {`
 - `content:counter(li);`
 - `counter-increment:li;`
- `}`



Exercise

- Create a page with five divs with lorem text and a bullet list of five items
- Use these styles to see what they do
 - `div:after { content: "– a div written by me"; }`
 - `div:before { content: url(images/favicon.ico); }`
 - `ul { counter-reset:items;}`
 - `ul>li:before {content: counter(items); counter-increment: items; }`

Icon fonts



- Icons fonts
 - <https://speckyboy.com/2014/05/14/more-free-icon-fonts/>
- Use
 - with links
 - as footnotes
 - as list item bullets

Process

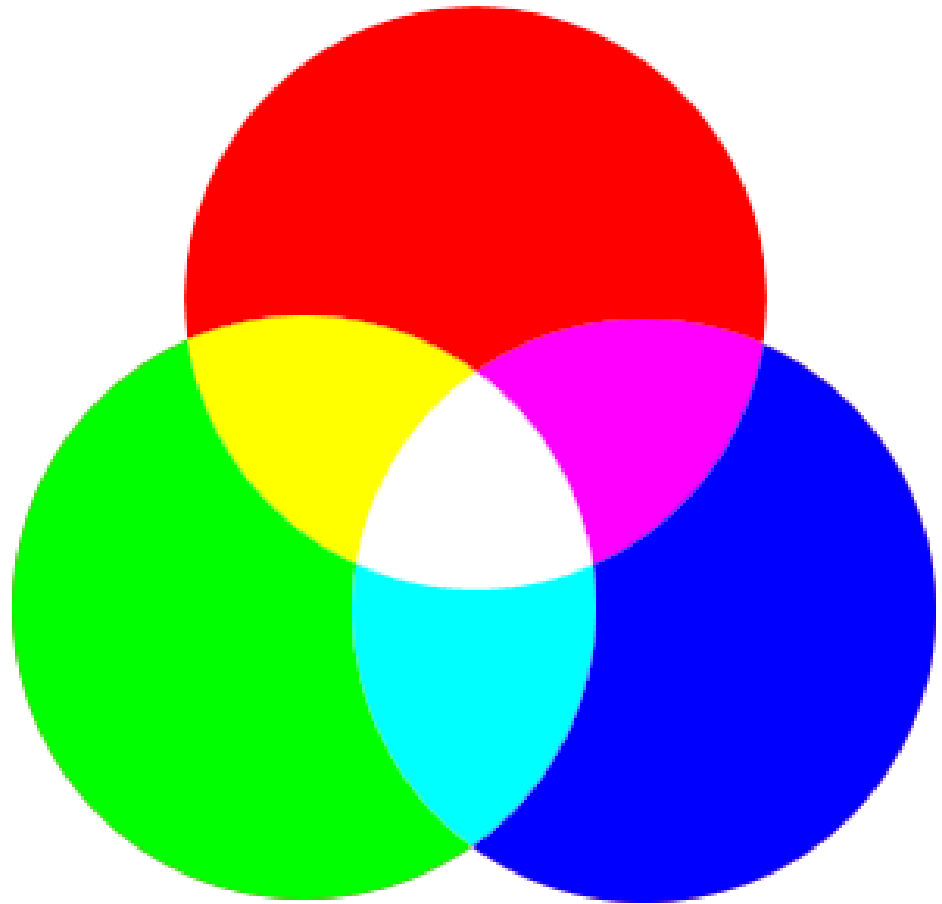


- Get text and images (maybe?)
- Draw sketches / prototypes
- Use a template or
 - Create the page stub (template)
 - Create the layout structure with container (block) elements
 - Create basic CSS classes and apply
- Add the text and images
 - Get feedback here.
 - Write CSS style if not written
- Apply CSS style and test



specifying subtractive color

CSS color



Color names



- Names
 - http://www.w3schools.com/cssref/css_colorsfull.asp
 - pre-defined by the browser supporting CSS2/3
 - 147 names in CSS3

yellow	gold	orange
darkorange	tomato	orangered
red	crimson	firebrick
darkred	brown	saddlebrown
maroon	chocolate	sienna
indianred	sandybrown	peru
salmon	lightsalmon	darksalmon
coral	lightcoral	rosybrown

lightseagreen	mediumturquoise	darkturquoise
paleturquoise	turquoise	cadetblue
lightcyan	cyan - aqua	darkcyan
lightblue	powderblue	teal
lightskyblue	skyblue	deepskyblue
cornflowerblue	royalblue	dodgerblue
darkblue	mediumblue	blue
indigo	midnightblue	navy

RGB - decimal

- Red Green Blue
 - additive system of light, not subtractive like color printing
- RGB(), RGBA()
 - red, green, blue, (alpha)
 - 0-255, 0-255, 0-255, (0.0 - 1.0)



RGB - hex

- Hex – digits 0 - F
- #RRGGBB
 - #000000 = black
 - #FFFFFF = white
 - #FF0000 = red, #00FF00 = green, #0000FF = blue
 - #111111, #777777, #A7A7A7 = grey shades
- #RGB – a shortcut
 - numbers are duplicated e.g. #123 = #112233



HSL

- Hue Saturation Lightness
 - hue - the basic color
 - saturation - amount of gray in color
 - brightness - amount of black in color, also called value
- HSL(), HSLA()
 - hue, saturation, lightness, (alpha)
 - 0-360, 0-100%, 0-100%, (0.0 - 1.0)



Exercise

- [css3/color-sampling.html](#)



CMYK



- A print color system is not used on the web called CMYK
- Cyan, Magenta, Yellow, Key/Black
- The colors able to be printed do not match the colors able to be shown on a monitor.



Alpha

- alpha channel = opacity (see-through)
 - 0.0 = transparent, 1.0 = opaque
- used for color specification in RGBA and HSLA
- also used for element fading
 - opacity:



foreground / background



- **color:** black;
 - will set the text color of the selector
- **background-color:** white;
 - will set the color of everything behind the text of the selector

Design principles



- Contrast makes text easier to read.
 - Choose foreground and background colors for readability.
- **Reversed out text**
 - Increase line height
 - Increase font weight

Gradients

- <http://www.colorzilla.com/gradient-editor/>
 - `html { height: 100%; }`
 - `body class='yourGradient'`



Orientation: vertical Size: 370 x 277 ☐ IE

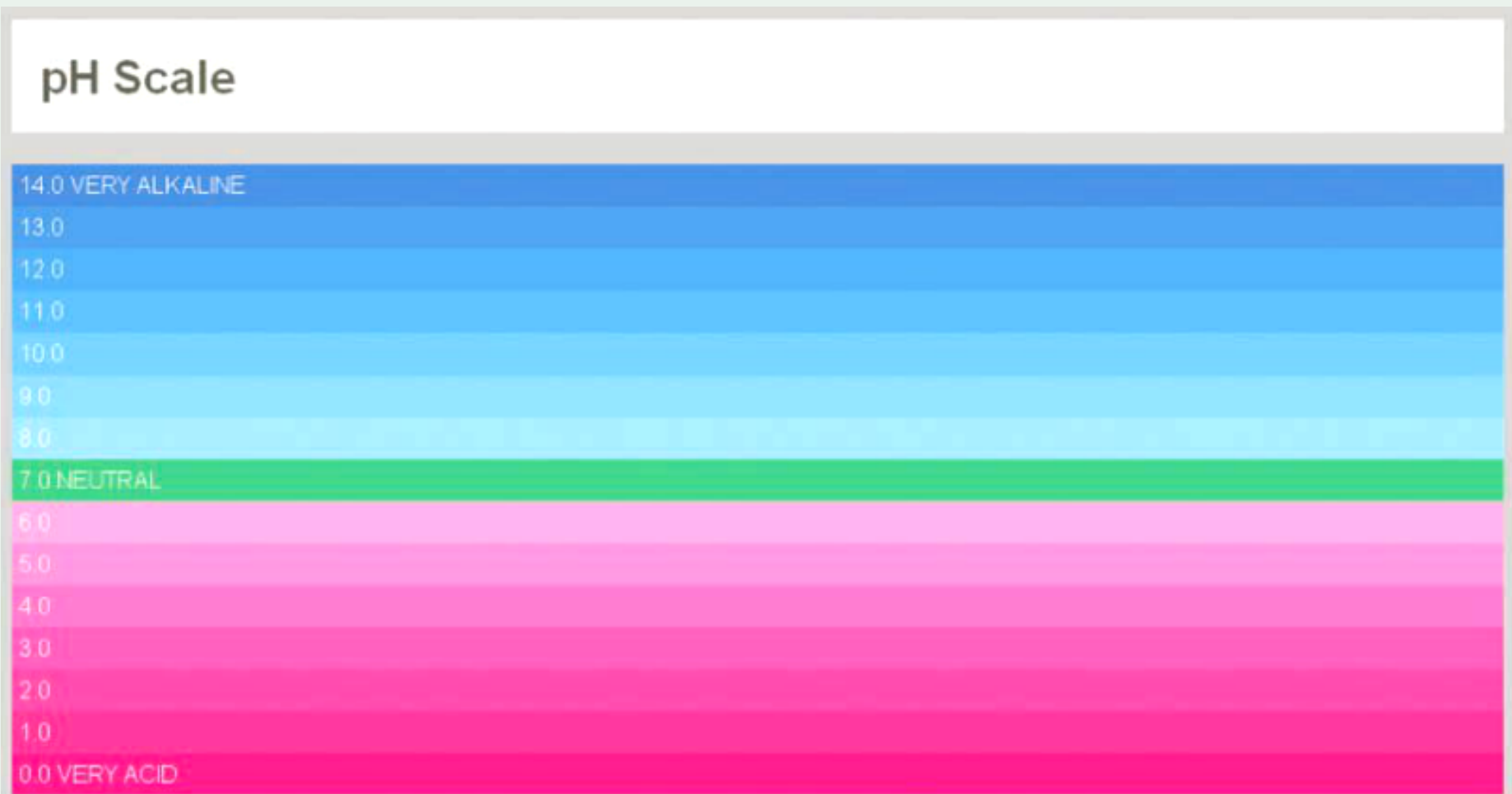
– CSS – [switch to scss](#)

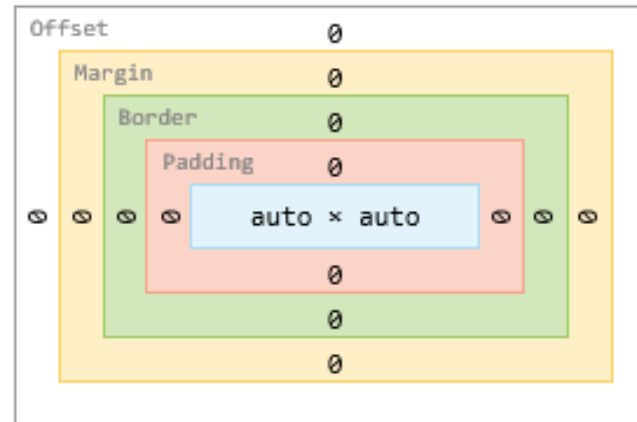
```
/* Permalink - use to edit and share this gradient:
http://colorzilla.com/gradient-
editor/#2968ce+0,89c4ff+100 */
background: #2968ce; /* Old browsers */
/* IE9 SVG, needs conditional override of 'filter' to
'none' */
background:
url(data:image/svg+xml;base64,PD94bWwgdmluc2lvdj0iMS4wI.
background: -moz-linear-gradient(top, hsla(217,67%,49%,1)
0%, hsla(210,100%,77%,1) 100%); /* FF3.6+ */
background: -webkit-gradient(linear, left top, left
bottom, color-stop(0%,hsla(217,67%,49%,1)), color-
stop(100%,hsla(210,100%,77%,1))); /* Chrome,Safari4+
*/
background: -webkit-linear-gradient(top,
hsla(217,67%,49%,1) 0%,hsla(210,100%,77%,1) 100%); /*
Chrome10+,Safari5.1+ */
background: -o-linear-gradient(top, hsla(217,67%,49%,1)
0%,hsla(210,100%,77%,1) 100%); /* Opera 11.10+ */
background: -ms-linear-gradient(top, hsla(217,67%,49%,1)
0%,hsla(210,100%,77%,1) 100%); /* IE10+ */
background: linear-gradient(to bottom,
hsla(217,67%,49%,1) 0%,hsla(210,100%,77%,1) 100%); /*
W3C */
filter: progid:DXImageTransform.Microsoft.gradient(
startColorstr='#2968ce',
endColorstr='#89c4ff',GradientType=0 ); /* IE6-8 */
```

Exercise (css3/ph-scale.html)



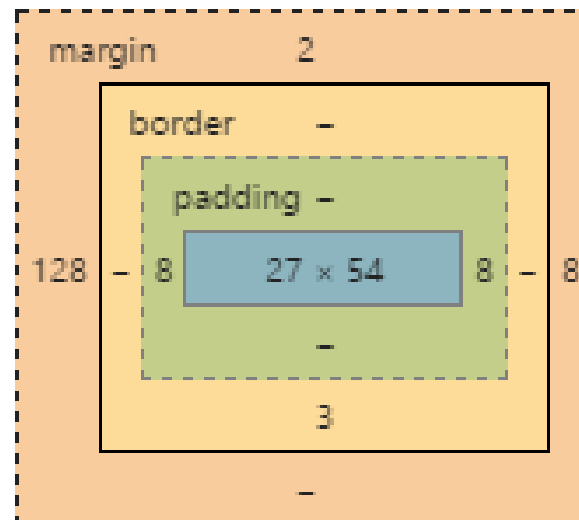
- Create this look:





Blocks have spacing control

CSS box model



display:



- All block elements use the box model to control space around its rectangle.
- Inline elements do not need a box model.
 - Some 'inline' elements are really not, like ``
- inline-block combines the block spacing with flow
- Use display: to change the type
 - `img {display: block }`
 - `div { display: inline-block }`
- Visualize: <http://codepen.io/carolineartz/full/ogVXZj/>

display:

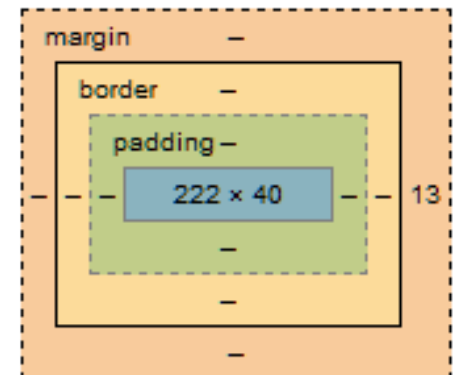


- display:
 - **block** - adds space as if a break is following
 - **inline** - will follow on the same line as the previous inline element and allow other inline elements to continue on the same line.
 - **inline-block** - flows like an inline element but is still a block.
 - **none** - hides element and removes space
- display for LI horizontal navigation
- display for JavaScript use for hiding elements in print views or collapsing layouts.

border



- border – the line of the box
 - border: (width) (style) (color);
 - border-top:, border-right:, border-bottom:, border-left:
 - uses (width) (style) (color)
 - border-width: 2px; (or thin, medium, thick)
 - border-style: solid (larger widths can do dotted, dashed, double, groove, ridge or inset)
 - border-color: black;
 - border-[side]-[property]
- See the Metrics in the Web Inspector



margin & padding



- margin – space outside the border
 - margin: (all sides);
 - margin: (top and bottom) (right and left);
 - margin: (top) (right and left) (bottom);
 - margin: (top) (right) (bottom) (left);
 - margin-top:, margin-right:, margin-bottom:, margin-left:
- padding – inside the border
 - same combinations as above
- No difference usually if you're not using a border or background

width: & height:



- width:
 - min-width: , max-width:
- height:
 - min-height: , max-height:
- auto
 - sets width based on parent or containing element
 - sets height based on content of element
 - default value

box-sizing (optional)



- IE8+
- = content-box
 - default
- = border-box
 - will include any padding and border width (not margin) in the width/height
 - better when doing 60%/40% page blocks
 - no need to calc
 - best to set at html { }

auto



- Set width, margin-left, or margin-right
 - to auto
 - Calculated to make the element box's width equal to the parent element's width.
- Set the other
 - to specific values
- Used for centering block elements often

Collapsing vertical margins



- Collapsing behavior applies only to margins
 - Padding and borders, where they exist, never collapse with anything

overflow:



- What to do when the text is larger than the container element is capable of showing because of CSS constraints
- Value
 - **hidden** - hide it
 - **scroll** - create scroll bars on the side

visibility:

- Leaves space where element should be
 - Not that good looking, use display: instead
- hidden, visible



Exercise



- Compare these:
 - `.show {`
 - `display: block;`
 - `}`
 - `.invisible {`
 - `visibility: hidden;`
 - `}`
 - `.translucent {`
 - `opacity: 0.2;`
 - `}`

background-image:



- background-image:url('image.png')
- background-repeat: value
 - repeat - both directions
 - repeat-x, repeat-y
 - no-repeat
- background-attachment:
 - scroll - moves with page scrolling
 - fixed - does not move with page scrolling
 - inherit - whatever the parent does
- margins and padding must be in graphic, not CSS

background-image:



- background-position:
 - left | center | right
 - left | center | right top | **center** | bottom
 - x% y% (from top left)
 - #px #px
- background: (shorthand)
 - color image repeat attachment position
 - strict order but all optional
- background-size
 - cover

Exercise – (text/greeting-card-bg.html)



- `.bg-snow-card {`
 - `background-image: url('http://www.linkcrafter.com/wp-content/uploads/2016/06/holiday-cards-bsilvia-christmas-card-designs-to-make-christmas-card-design-competition.jpg');`
 - `background-size: cover;`
 - `background-repeat: no-repeat; background-color: hsl(207, 62%, 75%); }`
- `div.centered-content {height: 5em; display: flex; align-items: center; justify-content: center; text-align: center;}`
- `.bg-white-glass { padding: 2rem; background-color:hsla(0,100%,100%, .4); }`

images/greeting-card-bg.html



Exercise

- Change the text of the greeting card and the background image for a new type of card.

text/newspaper-article.html

Exercise

- Create a page that looks like this newspaper article.

BORDER WAR CONTINUES |
Tax incentives trump tradition

KANSAS PREVAILS IN BID FOR AMC


Longtime downtown company is going to new \$30 million building in southern Leawood's Park Place.

By KEVIN COLLISON
The Kansas City Star

AMC Entertainment is leaving downtown Kansas City for Leawood, ending a year-long contest that was the poster child for the metro area's business-poaching border struggle.



Forms

Name *		Time *	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
First	Last	HH	MM
			AM <input type="button" value="v"/>
		AM/PM	
Email *		Date *	
Please use your office email address.			
<input type="text"/>		<input type="text"/>	<input type="text"/>
		MM	DD
			YYYY
			
Address *			
<input type="text"/>			
Street Address			
<input type="text"/>			
Street Address Line 2			
<input type="text"/>	<input type="text"/>		
City	Region		

<form>



- submitting a form requests a page
 - sends the data from the form to the server with the request
 - data is put into key-value pairs based on the **name** attribute and the value of the input element
- action – page or route requested
 - to process data
- methods – GET, POST
 - how the data is bound to the request

methods - GET



- Data is scraped from the form
- Values placed in a query string
 - an appended string to the URL starting with a ? and using a key-value pair for each piece of data in form of key=value&key=value&...
 - spaces are encoded with + or %20
 - get+request
 - to save the URL with the data by the user

methods - POST

- Data is scraped from the form and placed in a special section of the request



methods - PUT, DELETE, HEAD



- Other less used HTTP commands can be used to set up REST web services
 - PUT – a create operation
 - GET – a read operation
 - POST – an update operation
 - DELETE – a delete operation
- Or retrieve information about a request
 - HEAD – gets size info of response

<input>



- Use the name attribute so that the data has a name for accessing in a script
 - `City<input type="text" name="billCity" />`
- Use a label element tied to the name of the input so a click in a radio button works on the label as well as the button
 - `<label for="billCity">City</label><input type="text" id="billCityID" name="billCity" />`

<input> - type attribute



- Text fields: type=text
- Checkboxes: type=checkbox
- Radio buttons: type=radio
- Send data button: type=submit
- Programmable button: type=button, type=image
- Masked: type=password
- File uploads: type=file
- Programmer entered: type=hidden
- Erase all data: type=reset

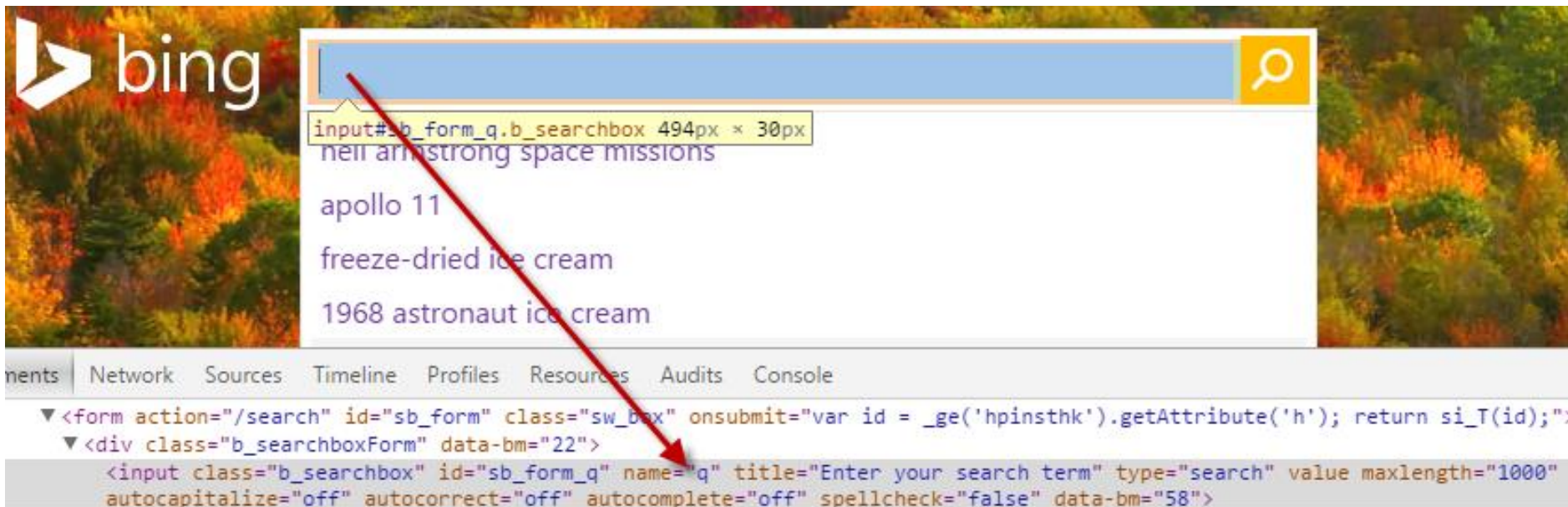
<input type=text



- Attributes
 - maxlength - restricts input
 - autofocus - use to place cursor in the text field on load
 - placeholder="helper text"
 - use to put suggestion in field
 - use to replace label to limit width on small displays supporting attribute
 - value="text to transmit" – default text
- Alternate HTML5 attributes
 - color, date, datetime, datetime-local, email, month, number, range, search, tel, time, url, week

<input name=

- the key used to associate the data with, used to compile the query string
 - name=q
 - <http://www.bing.com/search?q=attribute&go=Submit&...>



<label>



- `<label>Text for a label:<input.../></label>`
- `<input type="radio...">`
`<label for="idOfInputElement">label text</label>`
 - allows users to click on label to select option
- Placement of labels
 - above or left - text input/areas, select boxes, file uploads
 - right - checkboxes, radio buttons

<input type=submit



- Creates a submit button which requests the page in the form action attribute with the data in the form
- `<input type="submit" value="textYouSee">`
- Best for text on button
- Use the button element or, better, use the input type of 'image'



Exercise

- Create a basic form that has one label and text input to submit to the same page.

<input type=checkbox



- <input type="checkbox" name="name"
- Attributes
 - name - several items checked will all be submitted with this name as the key of the query string property
 - It's not necessary to put all your checkboxes in the same group. They can be unique names.
 - value - this text will be submitted to the server
 - checked='checked' - will pre-check an option
 - checked by itself will also work

<input type=radio



- <input type="radio" name="groupName">
- the body contains the display text
- attributes
 - name - use this same name to choose among these named options. The key of the query string property.
 - value - the text that gets submitted to the server
 - checked - will pre-select an option
- Once selected, user can not unselect an option. Provide all options. Use checkboxes for booleans.

<input type=file



- <input type="file">
- Used for file uploads.
- Allows user to select from files on client.
- Requires server code.
- Add enctype to form
 - **<form action="verify.php" method="get" enctype="multipart/form-data">**

<input type=hidden



- <input type="hidden">
 - used to submit data that the programmer wants to put there
- Very useful for state management in ASP and JavaScript

<button>

- Button can also be a submit or reset type
- Use any html content
- <button> will submit without type=button
- type=button



<input type=image, type=button



- <input type="image" src="nicerLookingButton.png">
- type='button'
 - better than submit button
 - needs an event handler (JavaScript)
 - put html inside, :before, :after

Submitting the form – enter key



- One input field
 - hitting enter in this field triggers form submit
- 2+ input fields and no buttons or `input[type=submit]`
 - hitting enter doesn't trigger submit
- One or more input fields and one or more buttons or `input[type=submit]`
 - hitting enter in any of the input fields will trigger the click handler on the *first* button or `input[type=submit]` *and* a submit handler on the enclosing form



Exercise

- Switch your basic form to submit a request to Google
 - `<form action='http://google.com/search' method='GET'>`
 - `<input name='q' type='text' ...`
- Change the result language to Spanish
 - `<input type='hidden' value='es' name='hl' />`
- Adapt it to search only a single domain
 - `<input type='hidden' name='sitesearch' value='centriq.com'>`



Exercise

- Create a form to search your own site

Form inputs - other

- Multi-line text fields - `<textarea>`
- Drop-down menus - `<select>` + `<option>s`
 - Emmet: `select+`



<textarea>



- <textarea> - multiline input with scroll bars
- default text to appear in box can be put in body
- attributes
 - cols & rows - use CSS width and height instead
- change text style

Comments :

A text area can hold an unlimited number of characters, and the text renders in a fixed-width font (usually Courier).

<select>



- Drop down lists
- <select>
 - <option value="valueSentToServer">Name of value</option>
- Attributes
 - name=... - the key of the query string property
- <select name="sel_state" style="font-size:.8em">
- <option>
 - value=... - the value sent as part of the query string
 - selected - which item is shown in the list
 - selected='selected' is XHTML

<select size



- the list box always shows more than one option
- `<select size="3" multiple="multiple">`
 - `<option value="valueSentToServer">Name of value</option>`
- Attributes
 - size - turns the drop down select box into one that shows more than one option without clicking.
 - multiple - multiple options can be selected with CTRL/Command or SHIFT but most users don't know how to use this.

<select> <optgroup>



- Indent your options under a heading
 - <select name='state'>
 - <optgroup label="USA">
 - <option value='AL'>Alabama</option>
 - <option value='AK'>Alaska</option>
 - <option value='AZ'>Arizona</option>
 - </optgroup>
 - <optgroup label="Canada">
 - <option value='ALB'>Alberta</option>
 - <option value='BCO'>British Columbia</option>
 - <option value='MAN'>Manitoba</option>
 - </optgroup>
 - </select>

Alabama

USA

- Alabama
- Alaska
- Arizona

Canada

- Alberta
- British Columbia
- Manitoba

attribute tabindex=#



- used on any element
- Controls the order which the TAB key will give focus to the element
- Starts at the low value and tabs to the next highest value.
 - No need to be consecutive
- Inputs without tab order are last in order.
- HTML5 does not require values to be in quotes so `tabindex=5` is OK.

<fieldset><legend>



- <fieldset>
 - helpful to determine edges of a form by using a border to surround the form elements
 - <legend>Text to insert in the parent fieldset</legend>

A visual example of an HTML form using the `<fieldset>` and `<legend>` tags. The form is enclosed in a rectangular border. At the top left, there is an orange box containing the text "This is my form". Below this, the form contains two input fields: "Name" and "E-mail", each followed by a yellow input box. At the bottom center, there is an orange "Submit" button.

cursor:



- auto, default
- Firefox/Mac, Safari/Mac, Chrome/Mac don't support PNG and JPG cursors. IE and Edge only supports cursors in the CUR format.
- limit yourself to the size 32×32 for maximum compatibility
- <http://www.cursor.cc/>



styling - :focus



- Will accept keyboard input, indicated by a visual boldness
- `input[type='text']:hover,`
`input[type='text']:focus`
`{`
- `}`
- `:blur`

Resources

- <https://www.jotform.com/>
- <https://www.google.com/forms/about/>
- <https://www.wufoo.com/>



Exercise

- Set up a form to use
 - text fields for tabindex
 - select with options



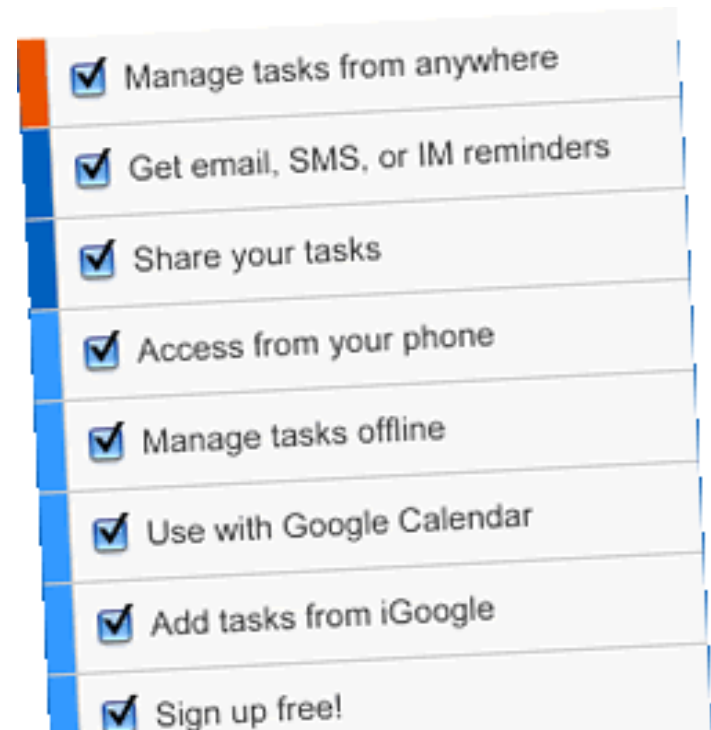
Resources

- Online editors
 - <https://jsfiddle.net/> - demo code, examples





Lists, menus



Elements



- `` - bullets or style with image
- `` - numbered or style with letters or Roman numerals
- `` - individual items

List styles



- list-style-type:
 - unordered - none, disc, circle, square
 - ordered - decimal, decimal-leading-zero, lower-alpha, upper-alpha, lower-roman, upper-roman
 - list-style-image:
 - `ul { list-style-image: url("img/icon.png"); }`
 - `li { margin: 10px 0px 0px 0px; }`
 - list-style-position: outside, inside
 - list-style:
 - style + image + position in any order
- `list-style-position:outside`
`list-style-position:outside`
 - `list-style-position:inside`
`list-style-position:inside`

Horizontal menu



- ``
- `Home`
- `News`
- `Contact`
- `About`
- ``

- What happens when you switch the display to block or inline?

- `a { text-decoration: none; }`
- `ul {`
- `list-style-type: none;`
- `margin: 0; padding: 0;`
- `}`
- `li {`
- `display: inline-block;`
- `width: 7rem;`
- `padding: 0.3rem;`
- `border: 1px grey solid;`
- `background-color: lightgoldenrodyellow;`
- `}`

List styles



- Menu styles without JavaScript
 - Listamatic - <http://css.maxdesign.com.au/listamatic/>
 - Listamatic2 - <http://css.maxdesign.com.au/listamatic2>
 - Listutorial - <http://css.maxdesign.com.au/listutorial/>

Exercise



- Create a menu of 5 items or more for page selection using this CSS

```
nav a {  
    text-decoration: none;  
}
```

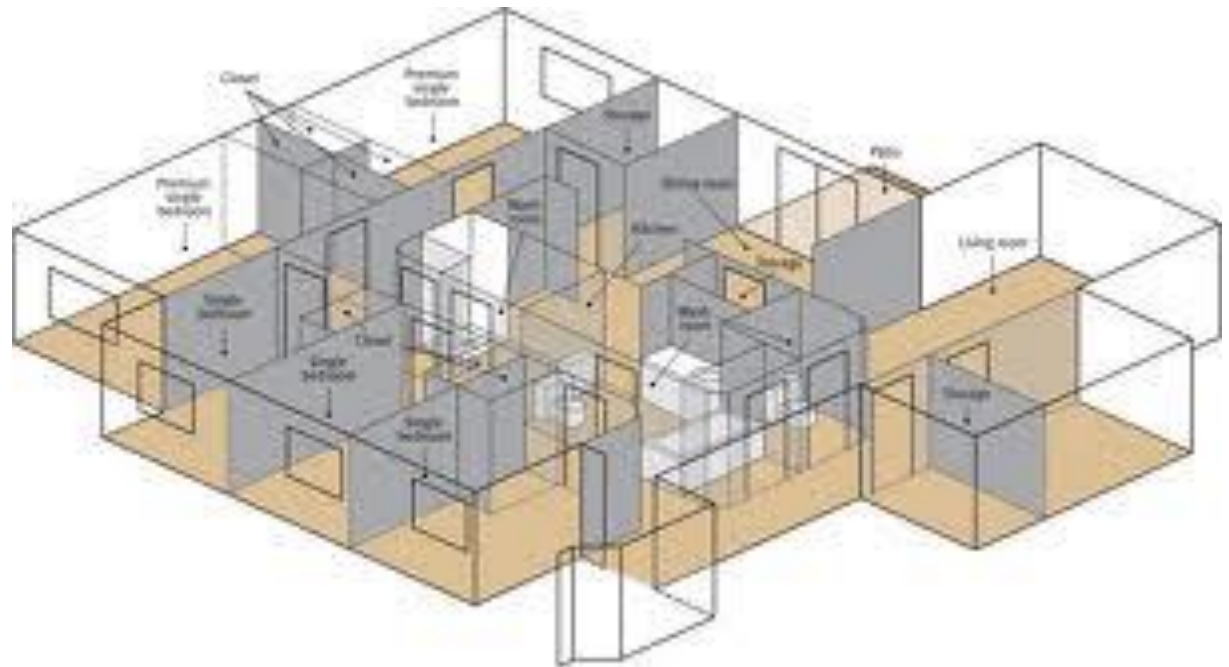
```
nav ul {  
    list-style-type: none;  
}
```

```
nav li {  
    display: inline-block;  
    border: 1px solid grey;  
    padding: 0.3rem 0.6rem;  
    width: 7rem;  
    text-align: center;  
    background-color: darkslateblue;  
}
```

```
nav a {  
    color: white;  
}
```



Layout



CSS - position:



- Positioning elements within block needs rules
 - top:, bottom:, left:, right:
- static - blocks flow to new lines, default
- relative
 - places/moves element to a new position relative to itself
- absolute
 - places/moves element relative to its parent element
- fixed
 - places/moves element in relation to browser window. For navigation panels.

CSS - z-index:



- stacking order – layers
- only works on positioned elements
- what matters is the relative numbers not the actual number
 - higher indicates to move to front/top
 - negative numbers are OK



Exercise

- Put a div with a square image in it on a page after a headline.
 - put a border around it
 - position it static
 - change to relative
 - change to absolute
 - change the opacity
 - change the z-index levels

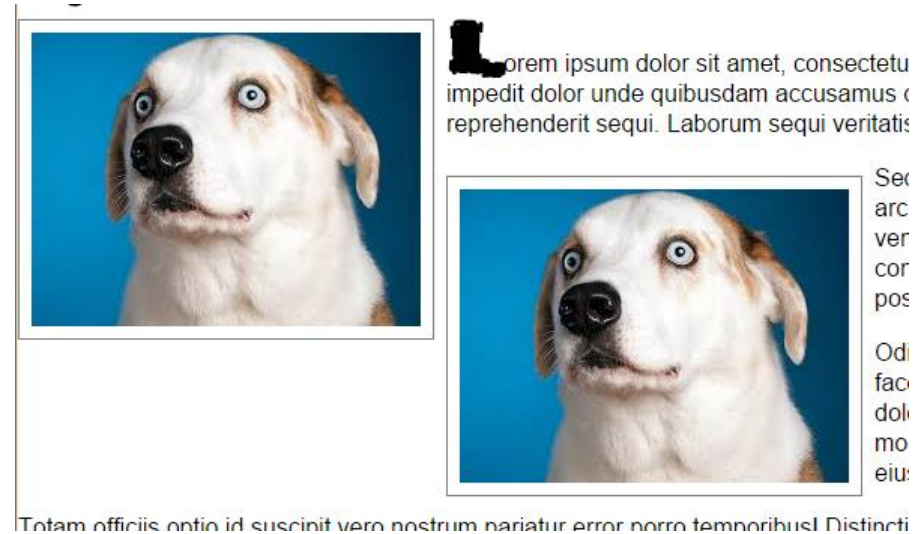
CSS - float:



- left or right
- Put the item floated before the item that it needs to align with. Text will wrap around.
- Use width on floated element to make sure it won't take up too much of the parent element.
- Commonly floated elements
 - img
 - aside
 - figure + figcaption
- Floatutorial - <http://css.maxdesign.com.au/floatutorial>

CSS - clear:

- both, left, right, none
- Element with clear will be placed under any floated element.
- Tips
 - Float images with clear so they don't pile up
 - Add clear to headlines so they always start after a floated element.





Exercise

- Image floats with clear
- Try
 - `aside > img + p.caption` – width 15rem, background-color
 - `figure > img + figcaption` – font-size 70%

Horizontal centering



- Centering a container on the page or centering it inside what it sits in
 - set the width of the container (default is 100%)
 - set the left and right margins to **auto**
 - puts equal space on each side
 - Make sure it's a block element
 - `` will need `display:block`
- Tip: turn borders on to check placement.
- prefer using flexbox

Vertical centering



- `.absolute-center {
margin: auto;
position: absolute;
top: 0; left: 0; bottom: 0; right: 0;
}`
- See example <http://codepen.io/strack/pen/dtsuf>
- prefer using flexbox

Fixed width layouts



- You get to control everything
 - The user doesn't get to control much.
 - Gaps can appear.
 - Content is cut off.
 - This is not the web point of view.
- Does not work with mobile (RWD)

Responsive layouts



- Use relative units of measurement and design for flexibility based on screen width
 - No extra space appears
 - Screen size is a problem because lines can be too long or too short.
- Stretching is good within limits for a design.
 - min-width:
 - max-width:

Media queries



- Conditional CSS based on a rule
- In the CSS file, it's a grouping of CSS rules for a specific purpose.
- For smaller screen use
 - @media screen and (max-width:765px) { your CSS rule sets }
- For printing use
 - @media print {
 - your CSS rule sets
 - }

Responsive web design



- best to have smallest last in
- `/* base CSS */`
- `@media only screen and (max-width:2000px){`
- `/* for sumo sized (mac) screens */`
- `}`
- `@media only screen and (max-width:1440px){`
- `/* styles for browsers up to 1440px; */`
- `}`
- `@media only screen and (max-width:960px){`
- `/* styles for browsers up to 960px; */`
- `}`

Exercise

- Create a responsive design that changes on small, medium, or large screens



Misc.



- Long scrolling
 - <http://speckyboy.com/2015/09/17/long-scrolling-web-design-trend/>
- Mega menus
 - <http://geedmo.github.io/yamm3/> - Bootstrap

Resources



- w3schools.com
 - https://www.w3schools.com/css/css_rwd_mediaqueries.asp
 - https://www.w3schools.com/css/css_rwd_templates.asp

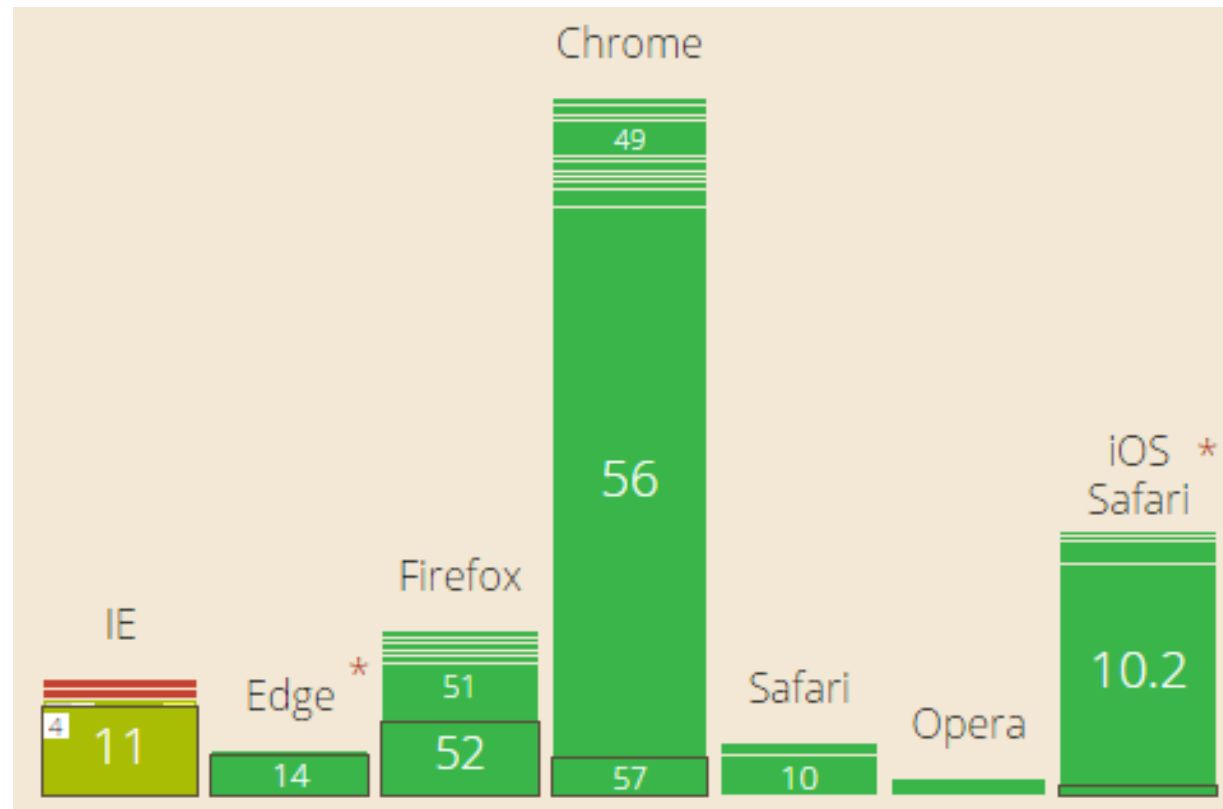


Table replacements

Flexbox, Grid, Multi-column

Flexbox

- 2009 specification
- huge rewrite in 2012 – IE 10
- another rewrite later



Flex container



- Assign one of these to a container
 - **display:** flex
 - **display:** inline-flex
- Children will be flex items which will
 - have the same height as its tallest element, and
 - stack horizontally with no space between the edges of each box

Flex container: direction, wrap

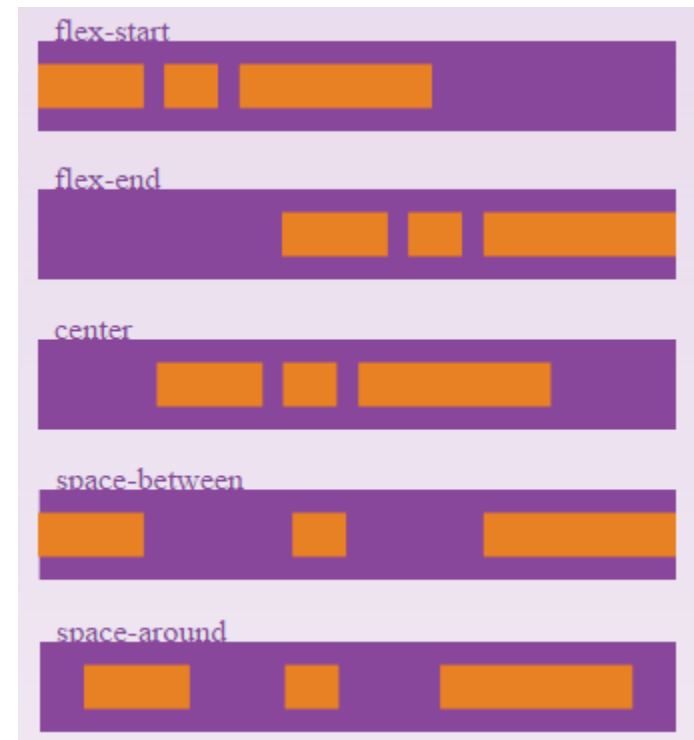


- **flex-direction**: establishes order of placement on axis
 - **row (default)** – left to right
 - row-reverse – right to left
 - column: top to bottom
 - column-reverse: bottom to top
- **flex-wrap**: allow the items to wrap as needed
 - **nowrap (default)**: single-line
 - wrap: multi-line top to bottom
 - wrap-reverse: multi-line bottom to top
- **flex-flow** = flex-wrap + flex-direction shortcut
 - **row nowrap (default)**

Flex container: justify-content



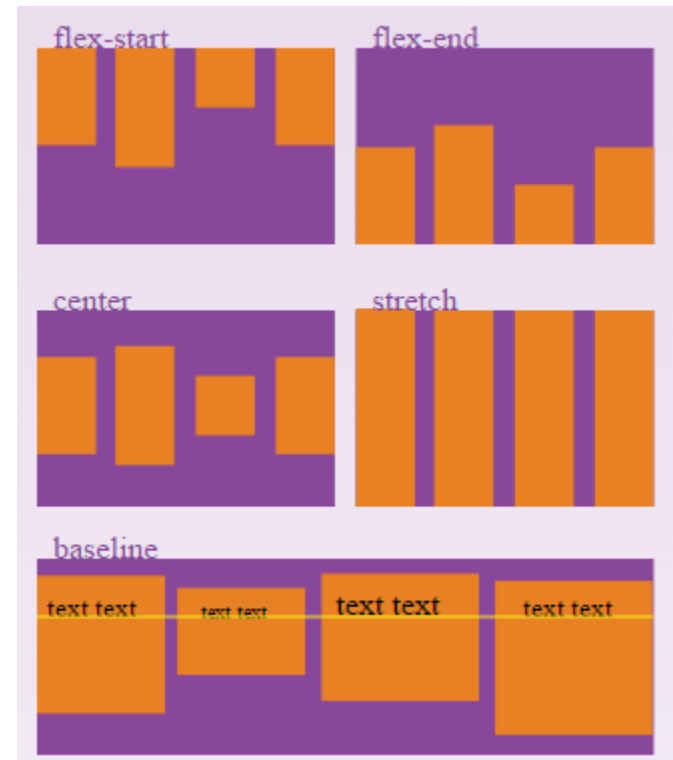
- defines the alignment along the main axis
 - flex-start - default
 - flex-end
 - center
 - space-between:
 - first item is on the start line, last item on the end line
 - space-around



Flex container CSS - align-items:

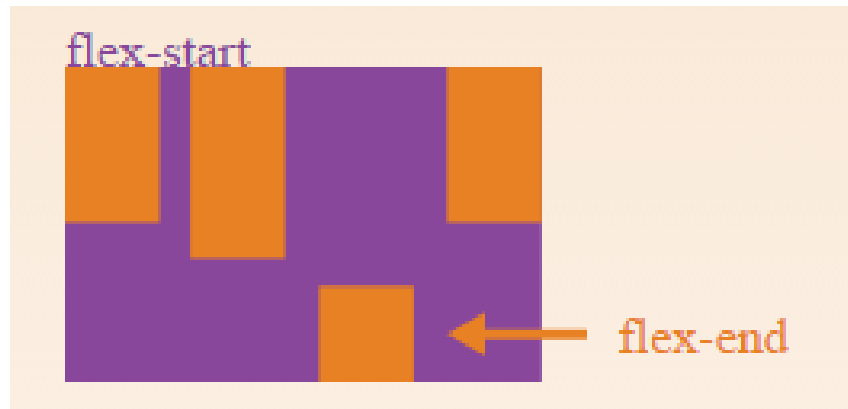


- layout along the cross axis on the current line
 - **flex-start**: cross-start line
 - **flex-end**: cross-end line
 - **center**: cross-axis
 - **stretch** (default): stretch to fill the container (still respect min-width/max-width)
 - **baseline**: baselines align



Flex item CSS

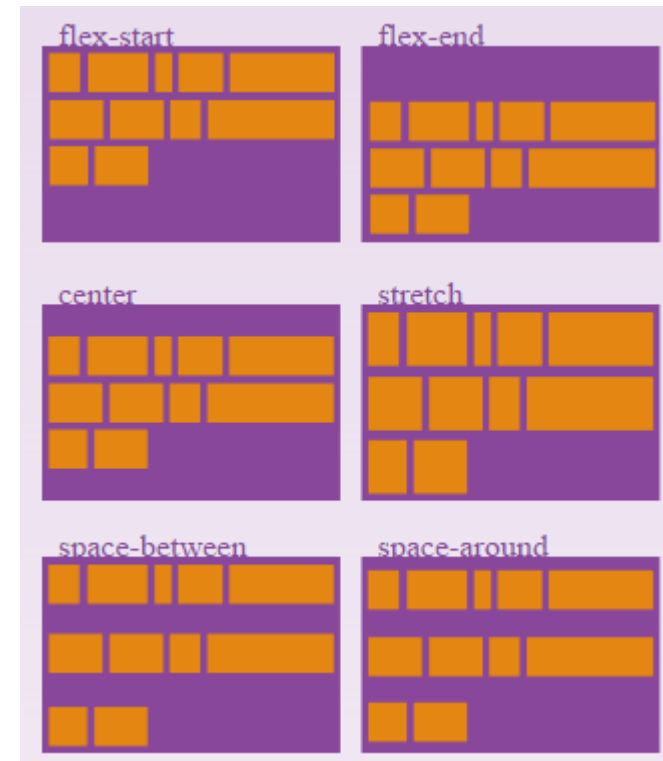
- **align-self: ?**
 - overrides align-items
- use align-items values
- float, clear and vertical-align have no effect on flex items



Flex container CSS - align-content:



- aligns/packs multiple lines when extra space in the cross-axis
 - flex-start
 - flex-end
 - center:
 - stretch (default): lines stretch to take up the remaining space
 - space-between: lines evenly distributed
 - space-around: lines evenly distributed with equal space around each line



Flex item CSS - flex

- = flex-grow, flex-shrink and flex-basis
- default: 0 1 auto
- best for setting other values that work



Flex container CSS

- **flex-wrap**
- flex-flow = flex-direction + flex-wrap



Flex item CSS

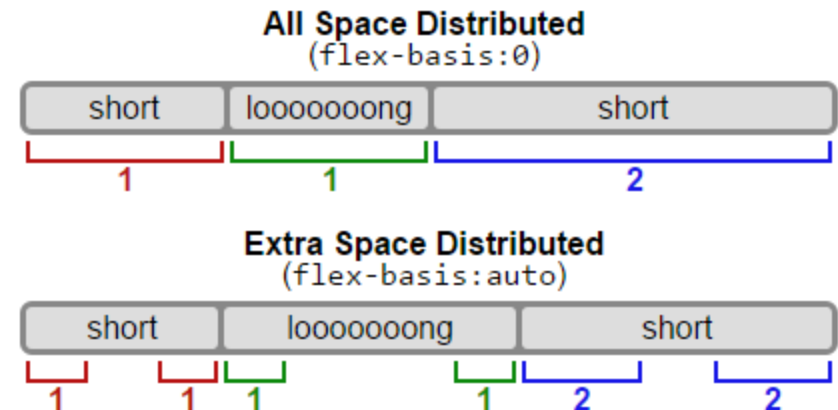


- **flex-grow: #**
 - uses a number as a proportion to show how much space to expand to (1, 2, 4, 2, 4, 6)
- **flex-shrink: #**
 - uses a number as a proportion to show how much space to collapse to (1, 2, 4, 2, 4, 6)
- **flex-basis:**
- **order: #**
 - controls layout order (1,1,1,4,6), (4,8,100, 200), (-5, -1, 0, 555)

Flex item CSS - flex-basis



- default size before the remaining space is distributed.
- 20%, 5rem
- 0 – ignore extra space
- auto - use width, height, flex-grow
- content – use item's content
 - max-content, min-content, and fit-content



Centering



- .flex-container {
 - display: flex;
 - flex-flow: row wrap;
 - justify-content: space-around; }
- RWD layout - <http://codepen.io/team/css-tricks/pen/jqzNZq>

Flexbox resources

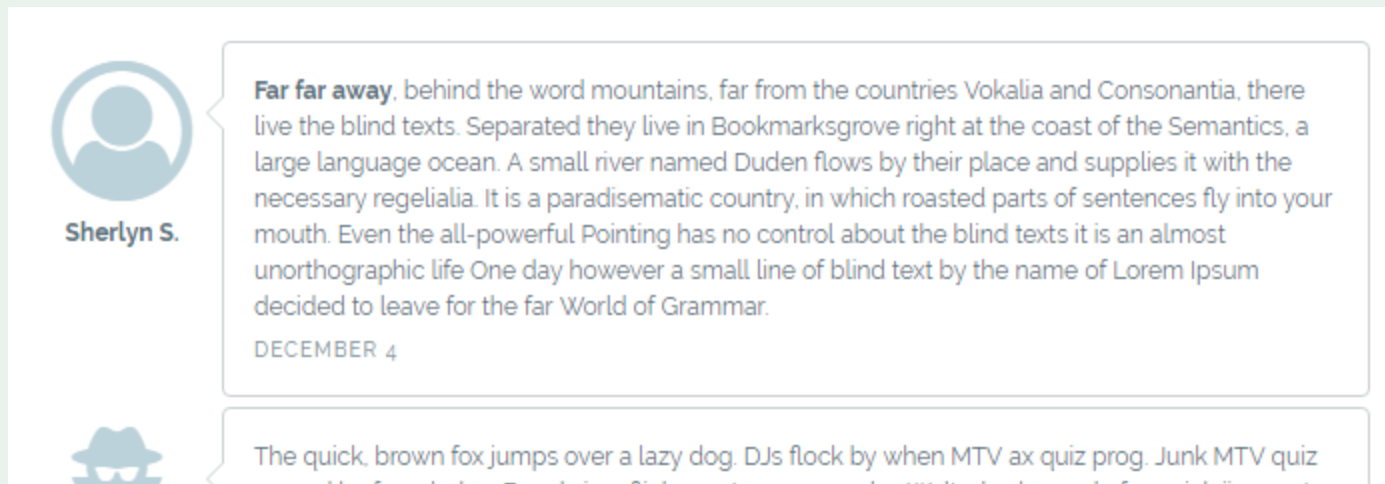


- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <http://www.flexboxpatterns.com>
- <http://maxsteenbergen.com/fibonacci/> - makes flexbox layouts
- <http://bennettfeely.com/flexplorer/> - change flexbox settings



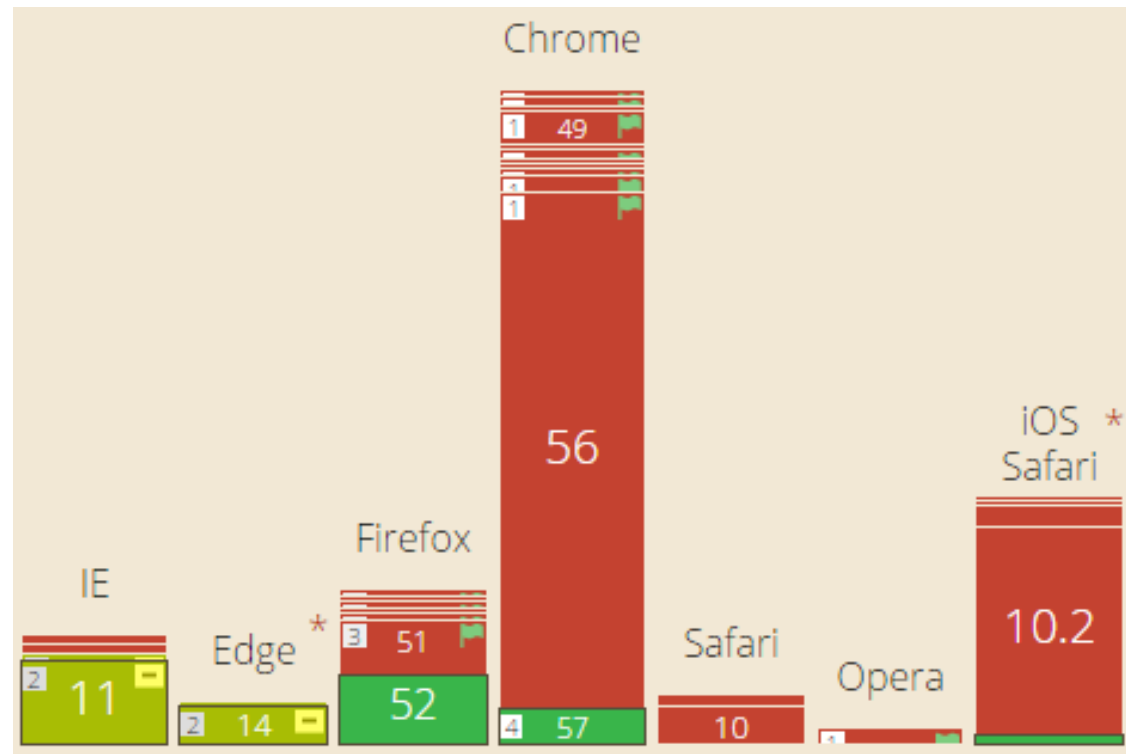
Exercise

- set up a grid for an icon and text message with flexbox



Grid Layout

- Emerging support
- Will be best for page layouts





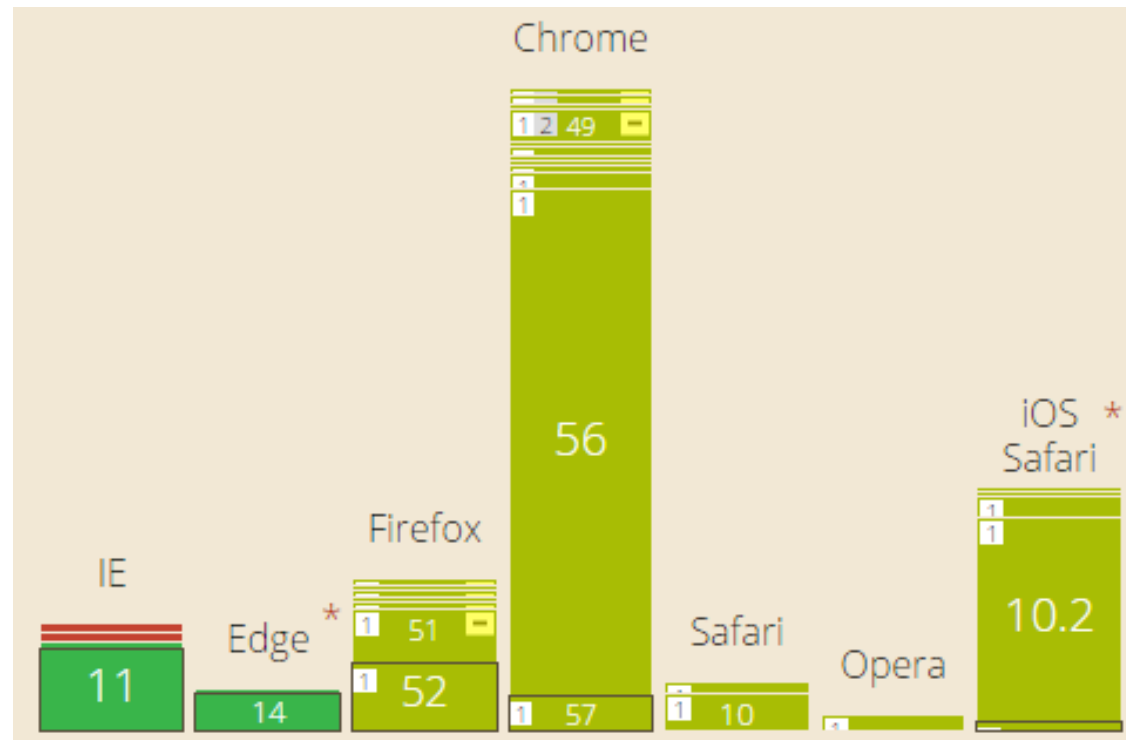
Exercise

- Investigate the layout of the grid by changing the browser width.

Multi-column Layout



- Text can't flow from one div to another
- Columns are usually divs that float left or right
 - column-count
 - column-width
 - column-gap
 - column-rule
 - column-span



Multi-column Layout



- Using LESS is easier
 - install extension in Code
 - <http://lesselements.com/>
 - create LESS file with classes using mixins
 - `.columns(250px, 3, 50px, #EEE, solid, 1px);`
 - (col width, count, gap, border color, border style, border width)



Exercise

- Set up serpentine columns of text like a newspaper article that flow from one to the next.
 - Use LESS with the elements mixins to help

Resources



- Grid layout
 - <https://css-tricks.com/snippets/css/complete-guide-grid/> or
 - <http://chris.house/blog/a-complete-guide-css-grid-layout/>
 - <http://gridbyexample.com/examples/>
- Multi-column
 - <https://css-tricks.com/guide-responsive-friendly-css-columns/>
 - <http://aaronlumsden.com/multicol/> - change multi-column settings



Resources

Templates



- Libraries of CSS & JavaScript
- Qualities you want
 - HTML, CSS, very little JavaScript
 - Easily modified
- Web sites
 - <http://templated.co/> , <http://www.csstemplatesfree.org/> (new)
 - [TemplateMonster](#), [ThemeForest](#), [DreamTemplate](#),
[TemplatesBox](#), [FreeWebTemplates](#), [TemplateWorld](#), [OSWD](#),
[4Templates](#)

Layout frameworks



- Grid systems
 - used originally in the newspaper industry
 - ads are sold by grid
- Zurb Foundation
 - <http://foundation.zurb.com/>
- Twitter Bootstrap
 - <http://twitter.github.com/bootstrap/>
 - <https://www.patternfly.org/>
- Inuit CSS
 - <http://csswizardry.com/inuitcss/>

CSS libraries



- Pure.css
 - <http://purecss.io/>
- Topcoat
 - <http://topcoat.io/>
- CSSDB.co – curated repository
 - <http://cssdb.co/>

CSS design



- Case study for Salesforce
 - <https://medium.com/salesforce-ux/building-an-enterprise-framework-is-hard-1e8d8b33e082> Sep 2015
- Design using the cascade
 - <http://simurai.com/blog/2015/09/09/back-to-the-roots/> Sep 2015
- ITCSS, BEM
 - <http://thomasbyttebier.be/blog/less-css-mess>

Material design



- Google's design philosophy
 - <https://www.google.com/design/spec/material-design/introduction.html>
 - <https://www.google.com/design/>
- Material Design Lite – Google's implementation 7/15
 - <http://www.getmdl.io/>
 - Templates, components, styles
- Others – Materialize, Material Bootstrap

CSS management

- Mobile first
- Class reusability
- Consistent naming
- Use SASS or LESS



Animation



- <https://css-tricks.com/comparison-animation-technologies/>
 - CSS – 3 or less motions
 - JavaScript – higher complexity, use GreenSock
- Shift.css – JavaScript driven but declarative
 - <http://shift.octavector.co.uk/>

Other



- <https://css-tricks.com/>
 - technique snippets
- <http://tympanus.net/codrops/>
 - great demos, CSS only, CSS & JavaScript
- <https://html5boilerplate.com/>
 - a standard starting point for best practices
- <http://webfieldmanual.com/>
 - list of great resources
- <http://caniuse.com>
 - compatibility