More HTML Basics

Attributes in Elements

- Element: any markup e.g …
- Attribute
  - Adds more meaning and extra data
  - E.g. <img src="" alt="">
  - E.g. …
- Sometimes mandatory, sometime optional
- Example: `<html lang="en">…`

Special resource names – the web

- CSC 170: `start.html` (for lab assignments only)
- Special filename: index *(as in index.html)*
- Web servers configured to automatically recognize
- If present, automatically loads
  - User doesn't (have to) type it in
- Examples:
  - http://www.rochester.edu/college/honesty**/index.html**
    same as…
  - http://www.rochester.edu/college/honesty
  - http://www.facebook.com**/index.php**
    same as…
  - http://www.facebook.com

Links to other Resources

- Examples:
  - <img src=""…
  - <a href=""…

- Link = path to a resource
- path: absolute or relative
- Absolute path:

```
<a href="http://www.rochester.edu/college/honesty/index.html">Academic Honesty</a>
```

- Relative path:

```
<a href="undergraduates.html">Undergraduates</a>
```

Relative paths (links)

- Same directory

```
<a href="index.html">Undergraduates</a>
```

- Child

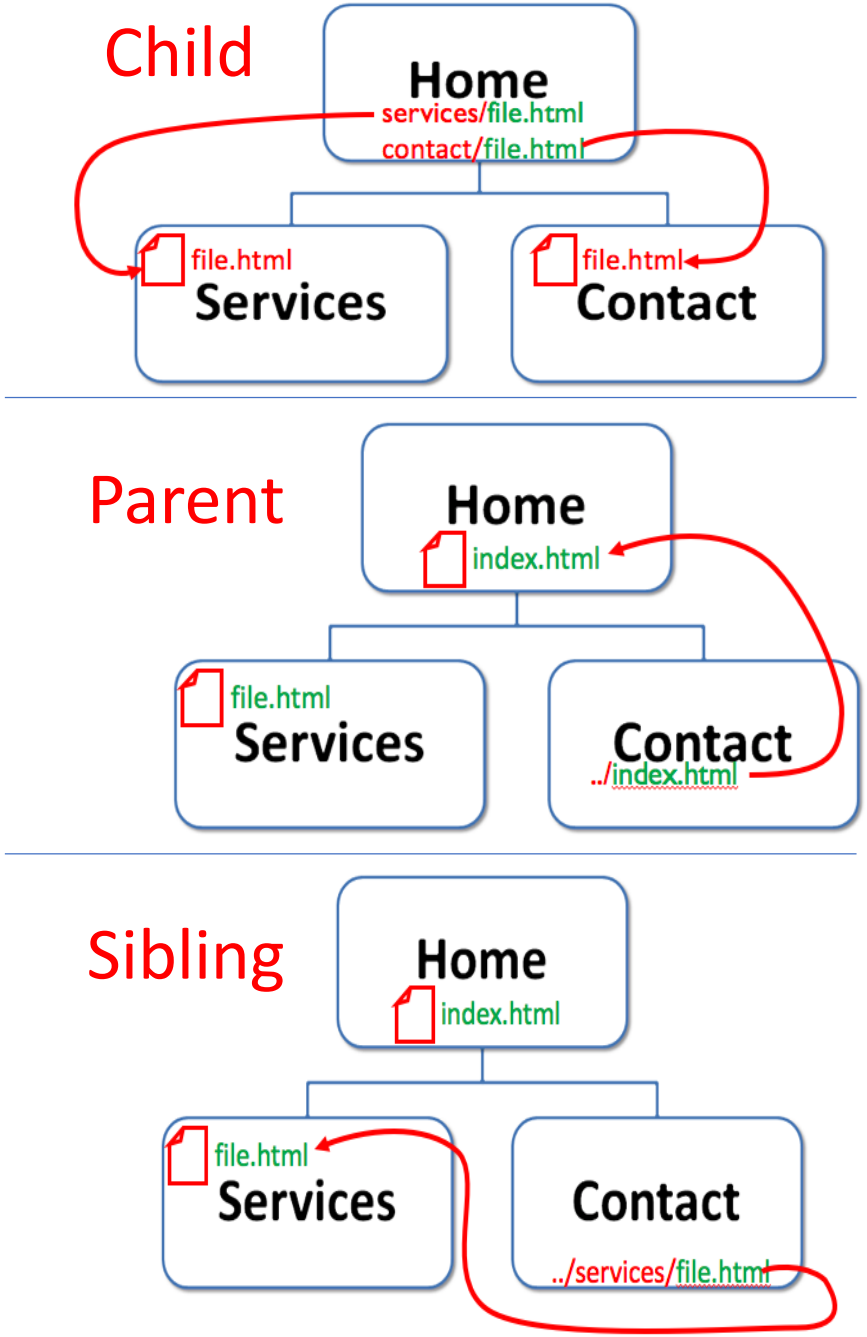
```
<a href="services/index.html">Undergraduates</a>
```

- Parent

```
<a href="../index.html">Home page</a>
```

- Sibling

```
<a href="../services/file.html">Home page</a>
```



External links

For absolute path links, i.e. links to resources on other servers…

- Using attribute: `target="_blank"` …is for off-server links
- Example:

```
< a href="http://www.google.com" target="_blank">Google</a>
```

- Never use for links within the same website

Progressive Enhancement

- Strategy for structured (web) development
- For building webpages in a layered fashion
- Each layer does not need more layers to be whole
- Each layer enhances (provides more value) to the next layer

Progressive Enhancement for Web Development

- Content - foundational layer
  - MS Word (?) …anything

1. Structure
   - HTML - hypertext markup language
   - Proper tags enable the "worldwide database" …big data
2. Presentation
   - CSS - cascading style sheets (next week)
   - formatting and layout
   - E.g. red = danger
3. Behavior
   - JavaScript (and others)
   - User interactions (clicking, tapping - things move around on the screen)

Web Document Structure

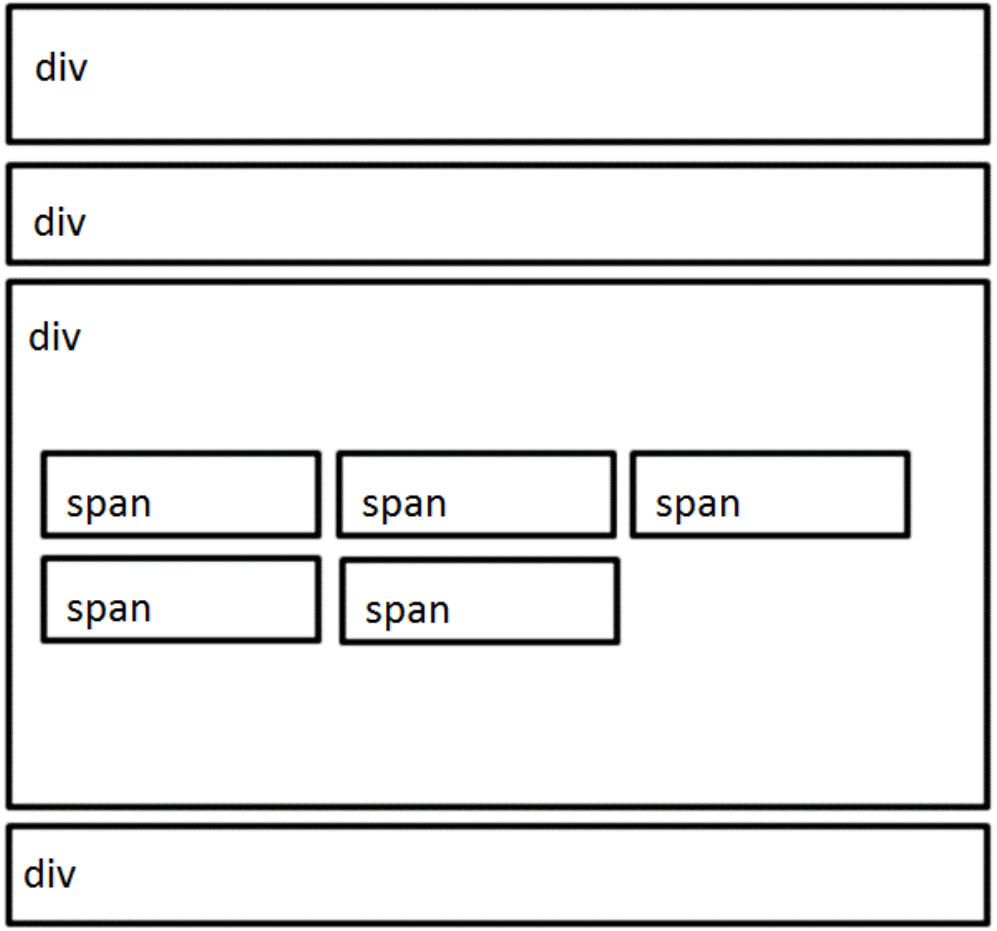
Semantically Correct HTML

The Semantic Web

- RULE: use HTML tags that describe the meaning of the content only (not the appearance)
- Separate: form from content
- See: ***The Machine is Us/ing Us*** **(YouTube)**
- One benefit (among many): find-ability…
  - Google scans webpages and indexes content
  - Google getting correct meaning out of words is hard
  - Tagged content (using the correct HTML tags) makes Google work better
  - YOU (the developer) pick the right HTML tags and your webpages will be found better in Google

Structural Elements

- Block and Inline
  - BLOCK tags: examples: H1, H2, etc., P
    - stack-up top over bottom
    - 100% width
    - as tall as the content needs to be
  - INLINE tags: STRONG, EM, A (hypertext), IMG
    - line-up side-by-side
    - as wide as they need to be
    - as tall as one line
- Non semantic value tags:
  DIV and SPAN (old fashioned)



HTML 5

- The World Wide Web Consortium (http://w3c.org) sets the standards for HTML and its related languages.
- New elements introduced
  - Note: lots dreamed-up by W3C - not all get much action
- Popular (we'll be using)
  - <header>…
  - < nav >…
  - < main >…
  - < article >… = a standalone chunk of content
  - < aside >… = content that can't stand alone; usually placed near an article
  - < footer >…

Lab assignments

- Starting in Lab 4: put your content into "structural" tags
- For the purposes of CSC 170 lab assignments:
  - Use these structural elements…
    - <header>…</header>
    - <article>…</article>
    - <aside>…</aside>
    - <footer>…</footer>
  - …just those, in that order
  - …nothing in between
  - Try to balance content between the ARTICLE and the ASIDE