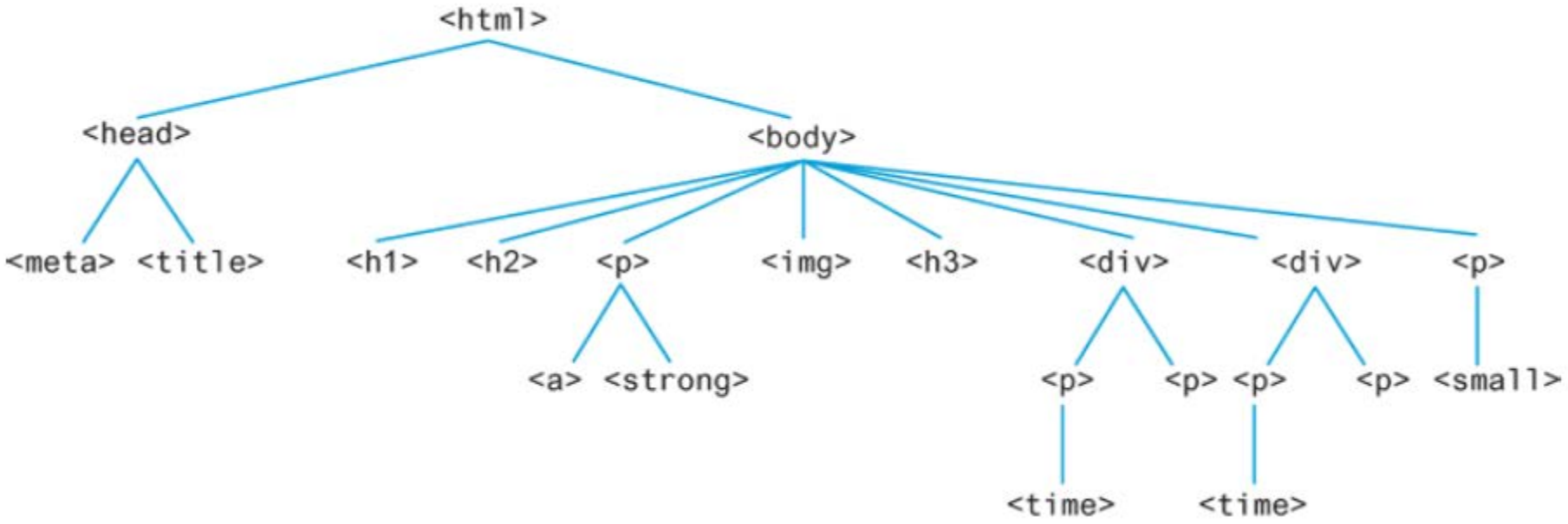
MIS 310 – CSS (cont)

- /* this is a CSS external style sheet comment and it can span multiple lines */
  - Useful for organization
  - Also debugging / hiding rules
- Normalize.css
  - http://necolas.github.io/normalize.css/
- Useful test page
  - https://github.com/cbracco/html5-test-page/blob/master/index.html

Remember the DOM



When devising CSS rules, keeping the DOM in mind will help.

Image from ***Fundamentals of Web Development*** (2nd Edition) by Randy Connolly and Ricardo Hoar

Selectors

- Element selectors

```
* { color: blue; }
p { color: red; }
```

- Class selectors

```
.ri { color: red; font-style: italic; }
<p class="ri">This will be red and in italics</p>
```

  – Adding the same class attribute value to different HTML elements will apply the same style

MDN reference for CSS selectors.

Selectors (cont)

- ID selectors

  #best { color: red; font-style: italic; }

  This will be red and in italics

  – ID selectors are used once per page; Class selectors can be used multiple times

- Attribute selectors

  [title] { border: 10px solid red; }

  <img src="pics/happy.jpg" alt="happy kids" title="happy“>

  – Can make use of additional symbols: =, ~, ^, $

Selectors (cont)

- Attribute selectors (cont)

```
a[href$="pdf"] { color: red; }
<p><a href="doc1.pdf">1pdf</a></p>
<p><a href="doc1.doc">1doc</a></p>
```

- Pseudo-Element selector

```
p::first-letter { font-size: 130%; }
```

- Pseudo-Class selector

```
img:hover { transform: scale(1.1); }
```

Selectors (cont)

- Contextual selectors / Combinators
  - Adjacent siblings: immediately follows

```
h1 + p { color: red; }
```

  - General siblings: follows

```
h1 ~ p { color: red; }
```

  - Child: direct descendants

```
h1 > p { color: red; }
```

  - Descendant: any level

```
h1 p { color: red; }
```

Cascading

- Inheritance – many (but not all) CSS properties affect descendants
  - E.g. font, color, text (not layout, sizing, borders)
- Selector specificity

  `body { color: red; }` is overridden by `p { color: blue; }`
- Location – last rule wins
- !important

  `.myParagraph { color: blue !important; }`

The Box Model

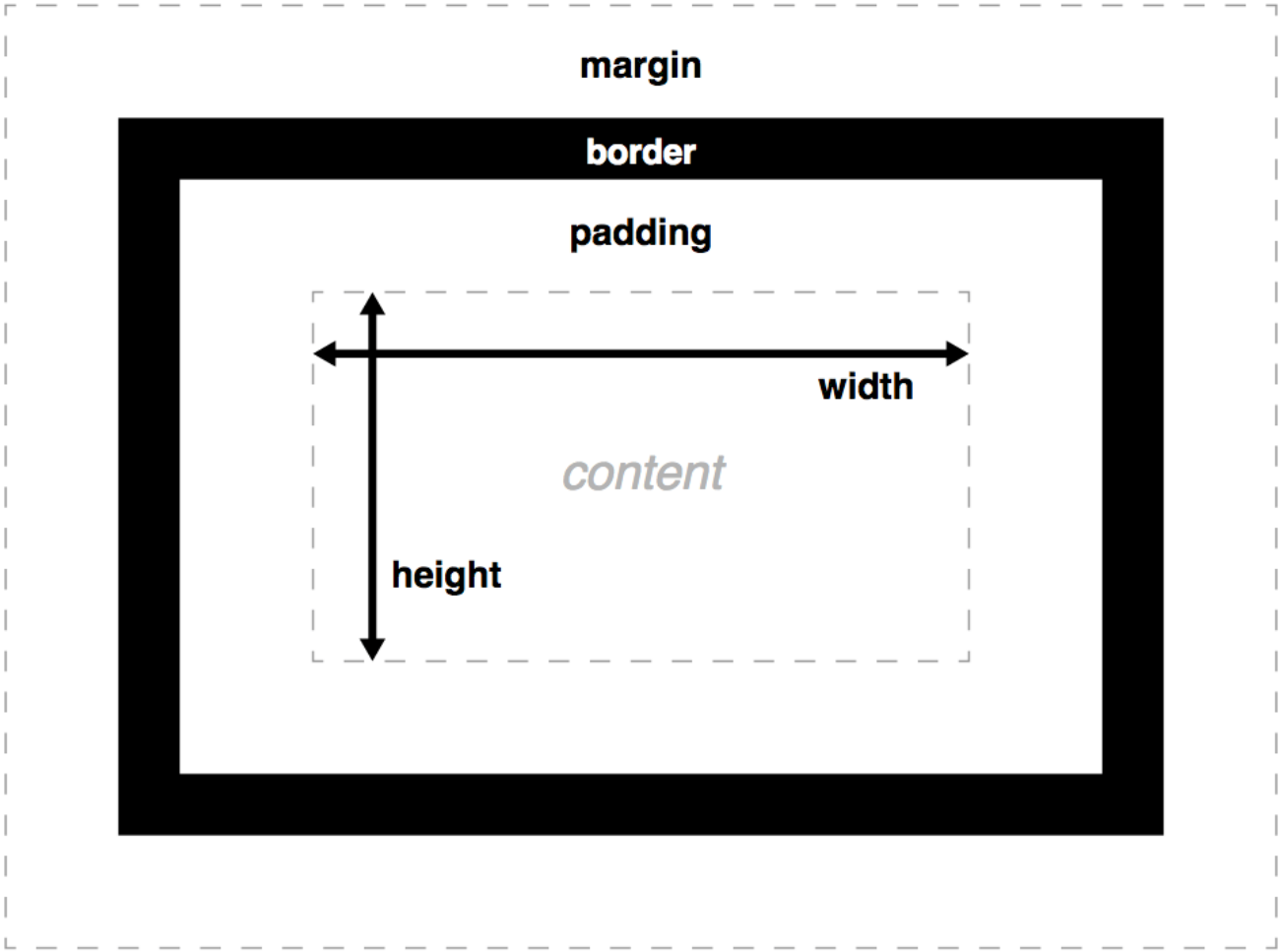


Image from MDN

Border

- Border [width] [style] [color];

  border: 2px dashed black;

  border-top: 2px solid black;

  border-color: red;

  border-color: red green orange blue; /* top-right-bottom-left */

Margin / Padding

- Examples:

  margin: -3px; /* All four sides */

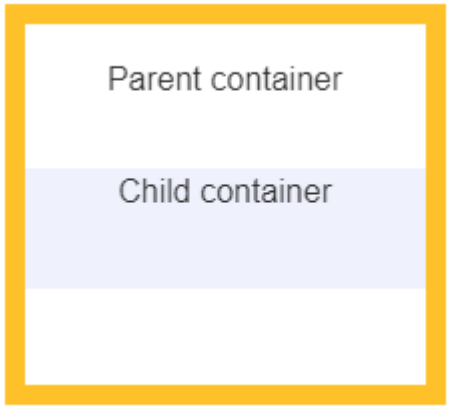
  margin: 5% auto; /* vertical | horizontal */

  margin: 1em auto 2em; /* top | horizontal | bottom */

  margin: 2px 1em 0 auto; /* top | right | bottom | left */

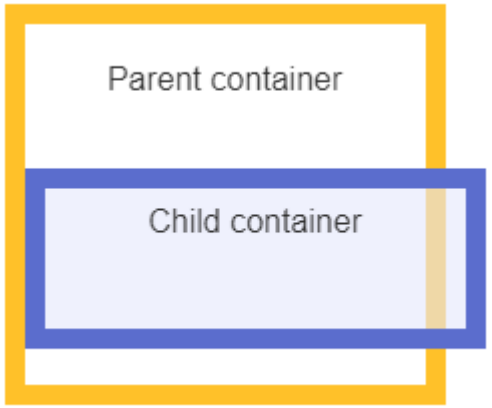
- Same approach for padding
- MDN demo: margin | padding

Sizing Concerns



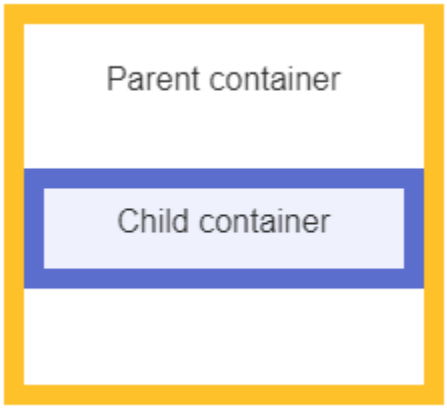
width: 100%;

Border/padding might cause a child element to extend beyond the limits of the parent element.



width: 100%;
border: solid #5B6DCD 10px;
padding: 5px;

Default box-sizing behavior is with respect to the content-box. Changing it to border-box will take padding/border into account.



box-sizing: border-box;
width: 100%;
border: solid #5B6DCD 10px;
padding: 5px;

Image from MDN