

# CSS: Cascading Style Sheets

---

Computer Science and Engineering ■ College of Engineering ■ The Ohio State University

---

## Lecture 15

# Evolution of CSS

- ❑ MIME type: text/css
- ❑ CSS 1 ('96): early recognition of value
- ❑ CSS 2 ('98): improvements in language
  - Adding media types (screen vs print)
  - Inconsistent support by browsers
- ❑ CSS 2.1 ('11)
  - In practice since '04
  - Took forever to standardize
- ❑ CSS 3
  - Breaks standard into many (50?) modules
  - Various modules already adopted & supported

# Key Idea

- Separate content and style
  - Different languages (syntax): HTML vs CSS
  - Different documents
- Goal: Single-point-of-control-over-change
  - Change font of every word in paragraph?
  - Change font of every `<em>` element in document?
  - Change font of every `<em>` element in every document on a site?
  - Change font of every `<em>` element which is part of instructions, but not finalized, on site?

# CSS Syntax

- CSS is *declarative* (not *procedural*)
  - Describe a thing, not how to do compute it
  - Example: RE matching
- CSS = list of *rules* (order can matter)
- Rule = a **location** & the **style** to use there
- Basic syntax of a rule

```
selector {  
    property1: style1;  
    property2: style2;  
    . . .  
}      /* comments always help */
```

# Example CSS

```
h2 {  
  /* draconian OSU visual identity */  
  color: darkred;  
  background: gray;  
  /* additional gratuitous styling */  
  font-style: italic;  
}
```

# Many Available Properties

- ❑ Background
  - `background-color, background-image`
- ❑ Text, font
  - `line-height, text-align, color`
  - `font-family, font-style, font-size`
- ❑ Border, margin, padding
  - `border-left-width, border-bottom-color`
- ❑ Positioning
  - `clear, display, float`
- ❑ Dimension
- ❑ List, table
  - `list-style-type`
  - `border-collapse, caption-side`
- ❑ Generated content and other fancy stuff
- ❑ See: [developer.mozilla.org/Web/CSS/Reference](https://developer.mozilla.org/Web/CSS/Reference)

# Shorthand Properties

- Example: Margins have 4 sides

```
margin-top: 3px;  
margin-right: 5px;  
margin-bottom: 7px;  
margin-left: 9px;
```

- Shorthand property: margin

```
margin: 3px 5px 7px 9px; /* TRBL */  
margin: 7px 9px;         /* TB sides */  
margin: 2px 6px 8px;     /* T sides B */
```

- Mnemonic: always "TRouBLe"

- Missing values filled in with provided value(s)

- Other shorthand properties:

- Padding, border-width, font, border, background...

# Including CSS: Mechanics

- Embed directly in element

```
<p style="color: red; background: gray">
```

- Place in style element in head

```
<head>
```

```
  <style media="screen">
```

```
    p {color: red; background: gray;}
```

```
  </style>
```

```
</head>
```

- Link to separate CSS file in head

```
<head>
```

```
  <link rel="stylesheet"
```

```
    href="3901Style.css" media="screen" />
```

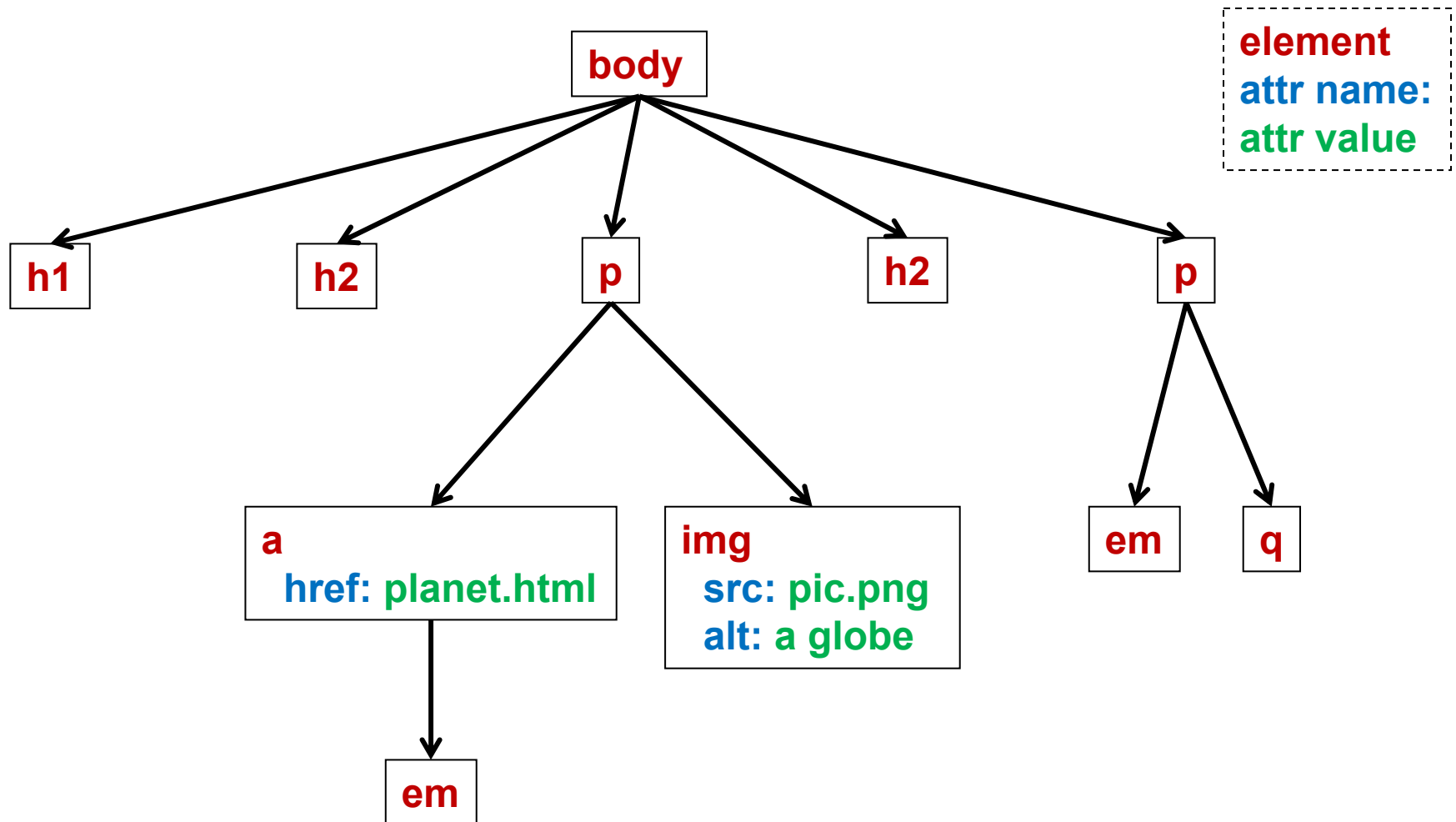
```
</head>
```



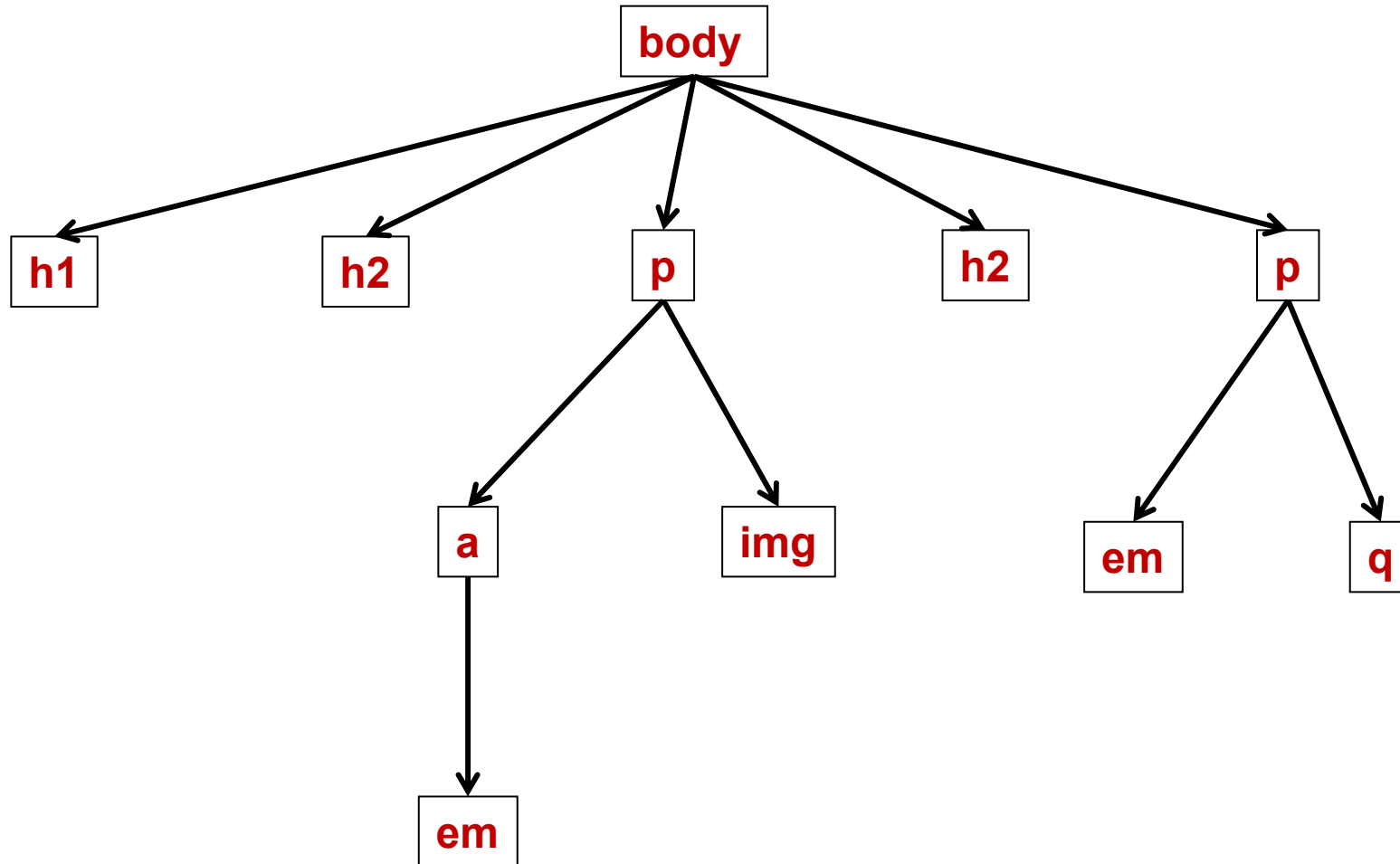
# Example CSS

```
h2 {  
    color: darkred;  
    background: gray;  
    font-style: italic;  
}  
em {  
    font-style: normal;  
    font-weight: bold;  
}
```

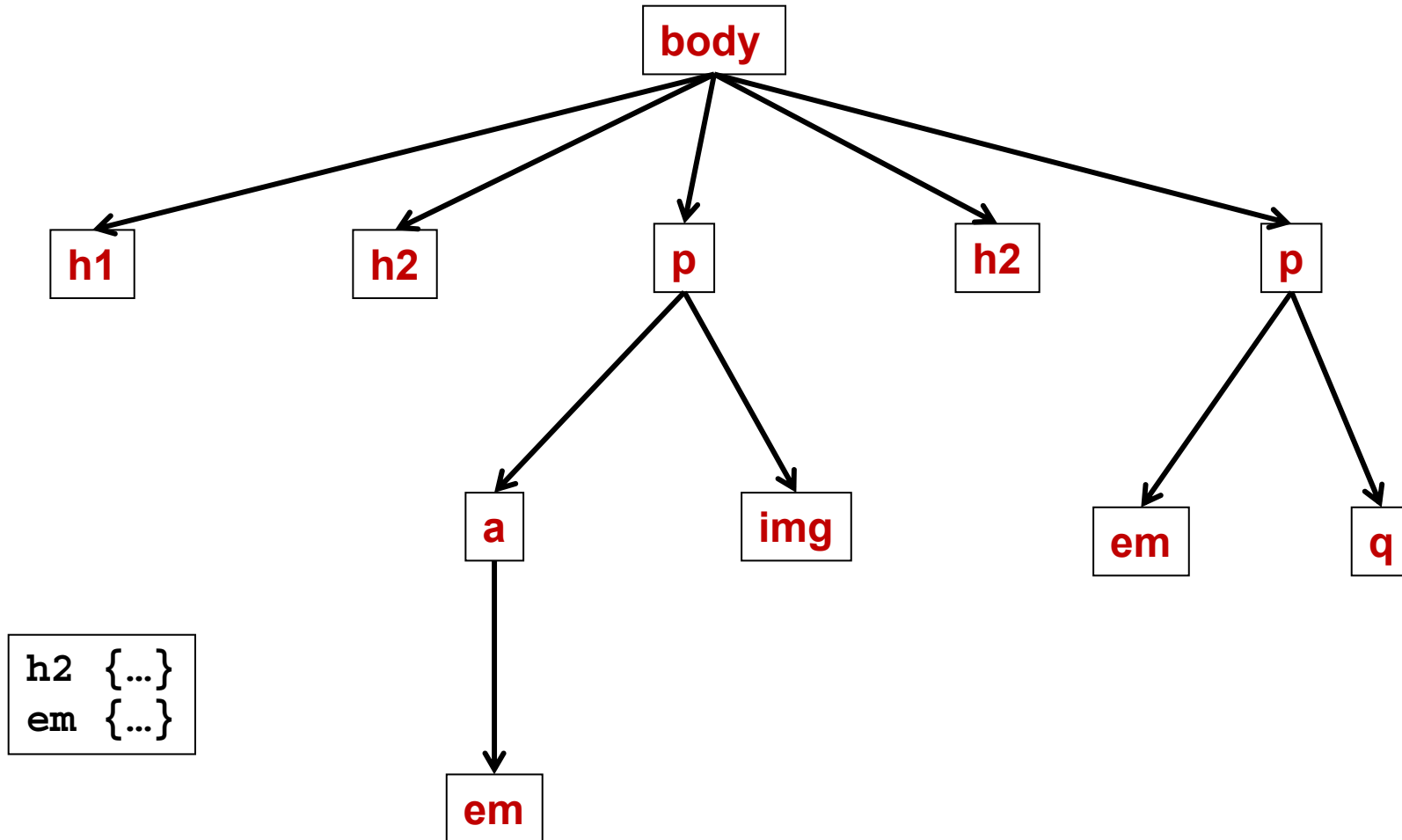
# Tree (Rooted at Body)



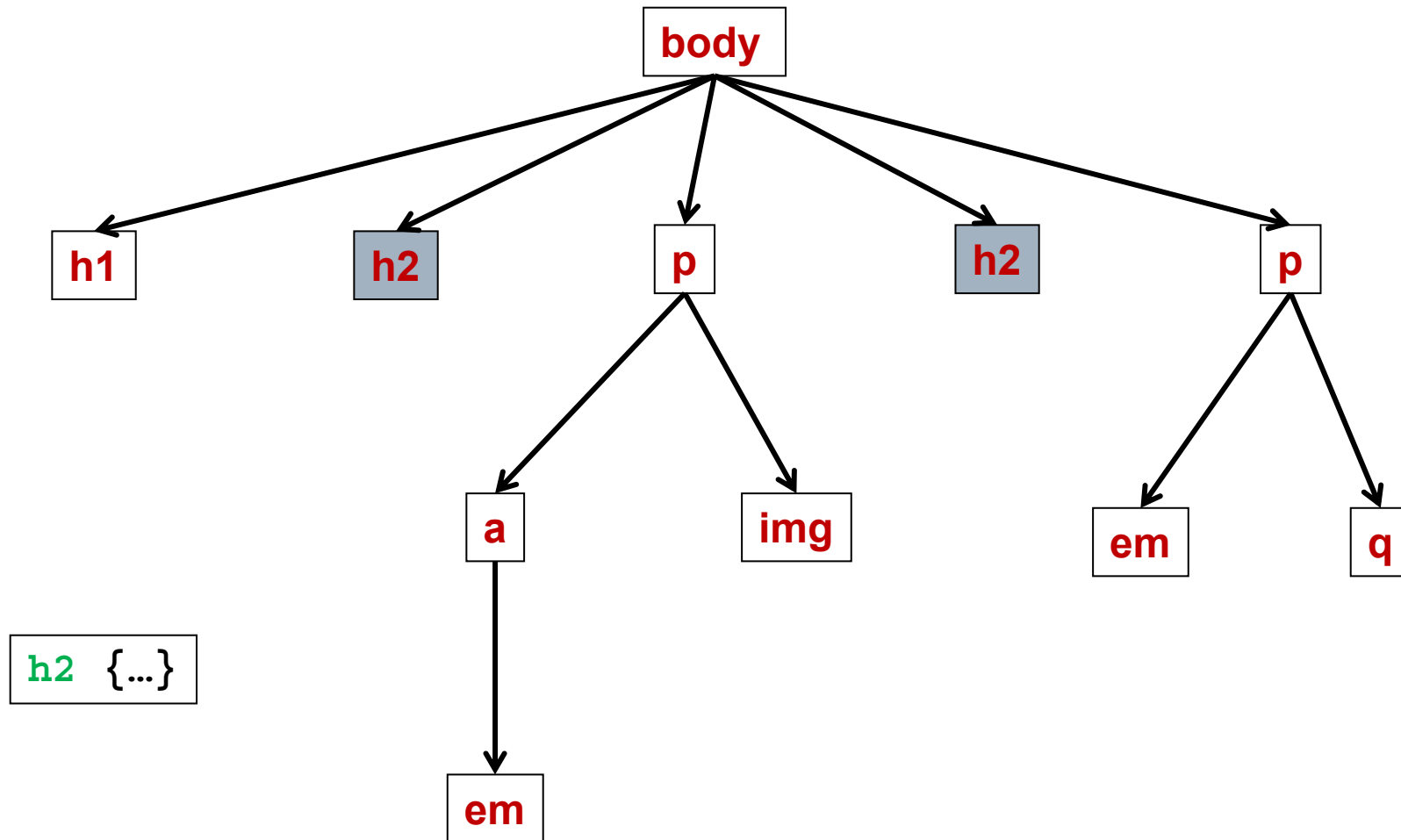
# Tree (sans Attributes)



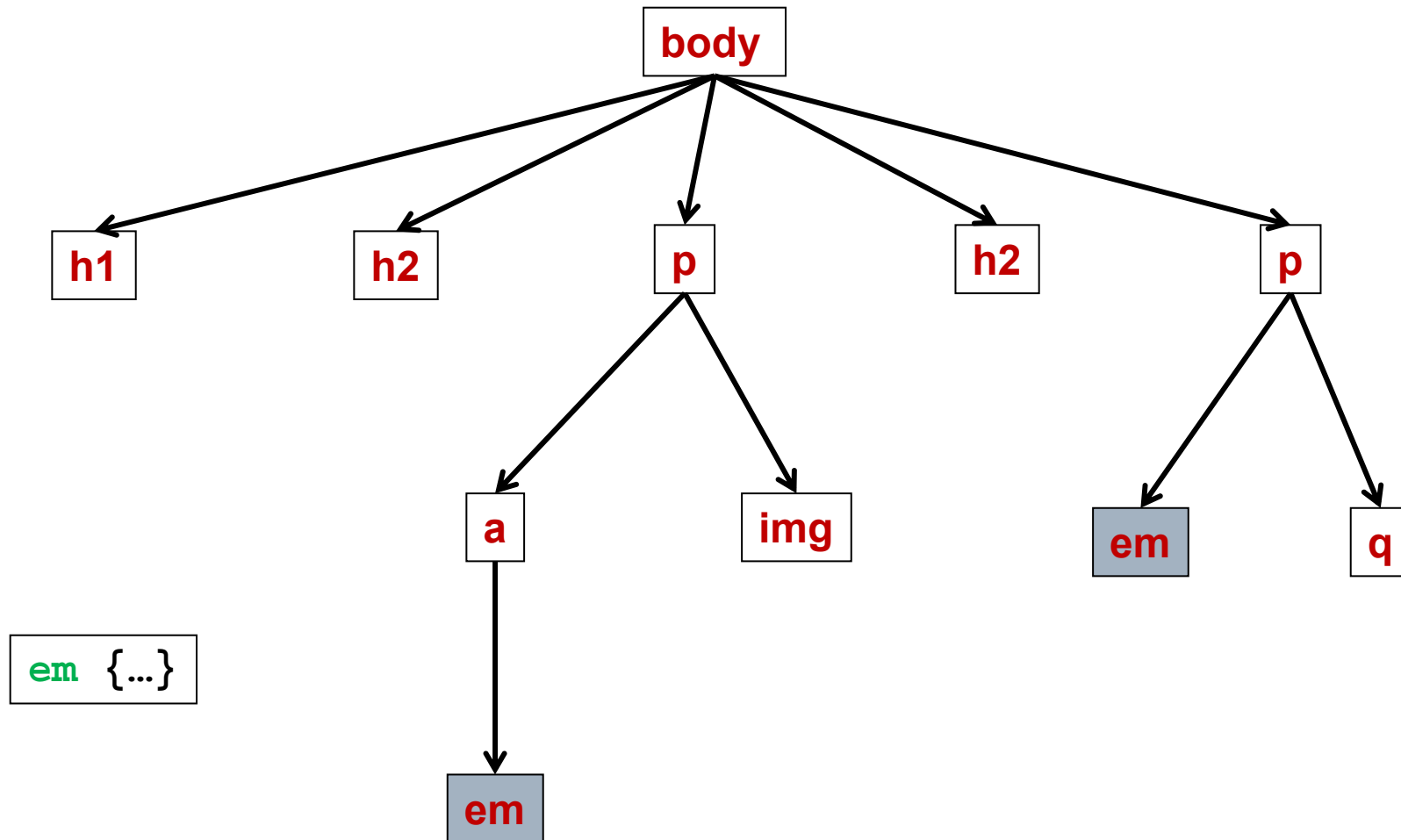
# Tree (sans Attributes)



# Selectors Applied to Tree



# Selectors Applied to Tree



# Multiple Selectors

```
h1 {  
    color: darkred;  
    background: gray;  
    font-style: italic;  
    border-bottom-style: solid;  
}  
h2 {  
    color: darkred;  
    background: gray;  
    font-style: italic;  
}
```

# Multiple Selectors: SPOCOC

```
h1, h2 {  
    color: darkred;  
    background: gray;  
    font-style: italic;  
}
```

```
h1 {  
    border-bottom-style: solid;  
}
```



# Inheritance for SPOCOC

- A child inherits many properties from parent by default

- Font weight, color, family, etc
- Can be overridden in child

- Set global styles in root

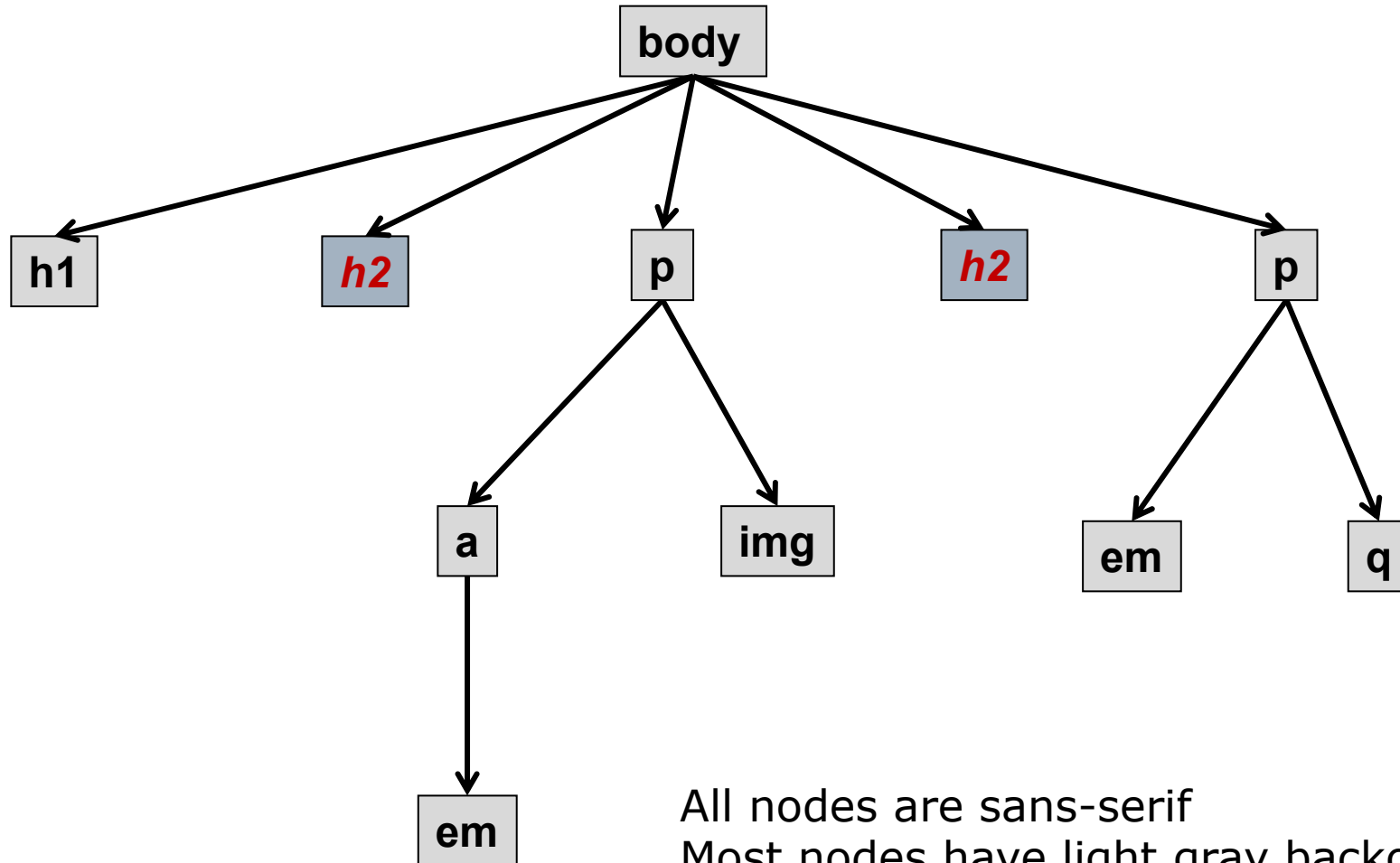
```
body {  
    font-family: sans-serif;  
}
```

- Contrast this with having to set property in all possible elements!
- Generally, text properties (eg color) are inherited, box-related (eg border) are not

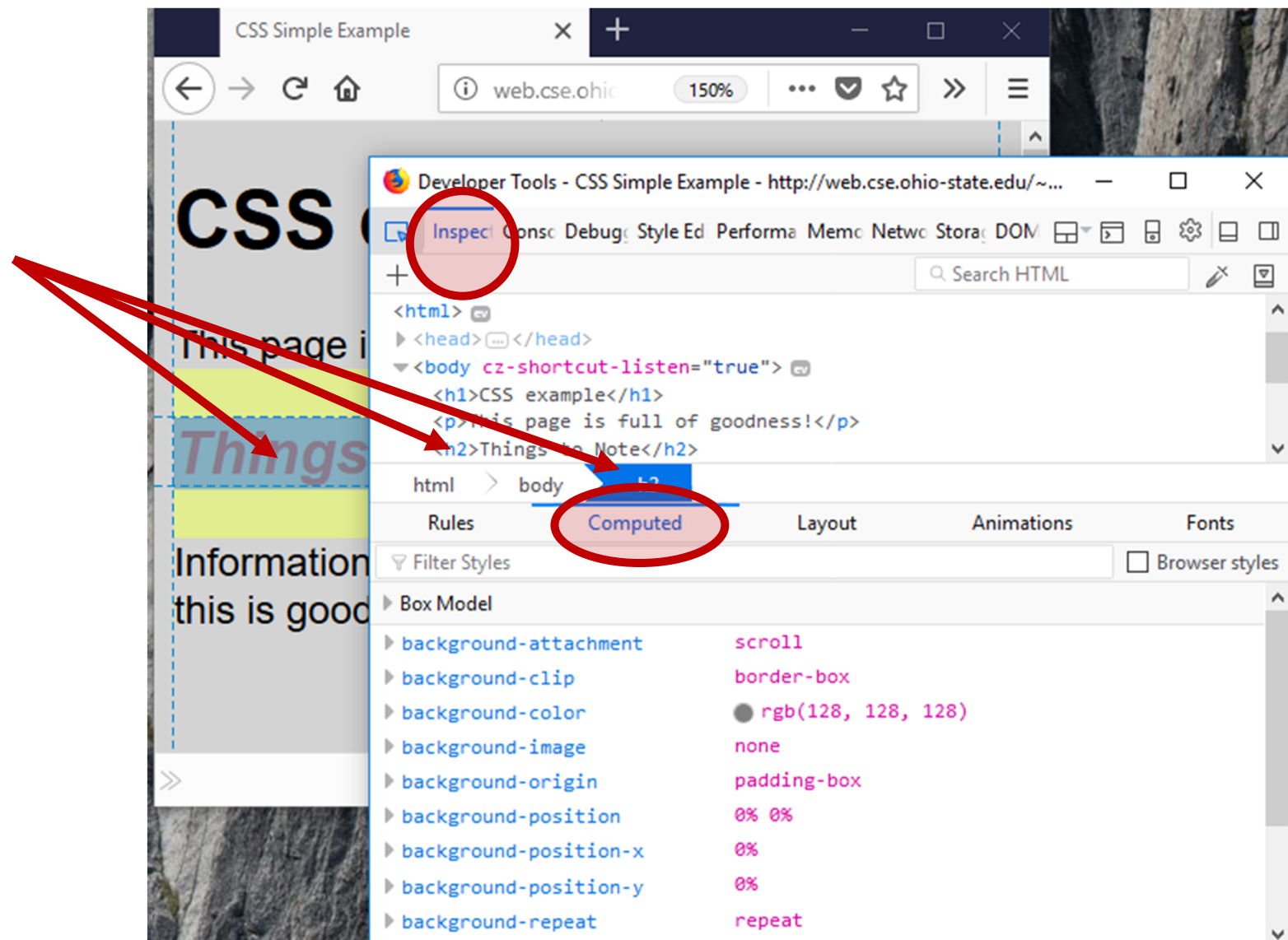
# Example Inheritance

```
body {  
    font-family: sans-serif;  
    background: lightgray;  
}  
h2 {  
    color: darkred;  
    background: gray; /*new backgrnd*/  
    font-style: italic;  
        /* inherits font family */  
}
```

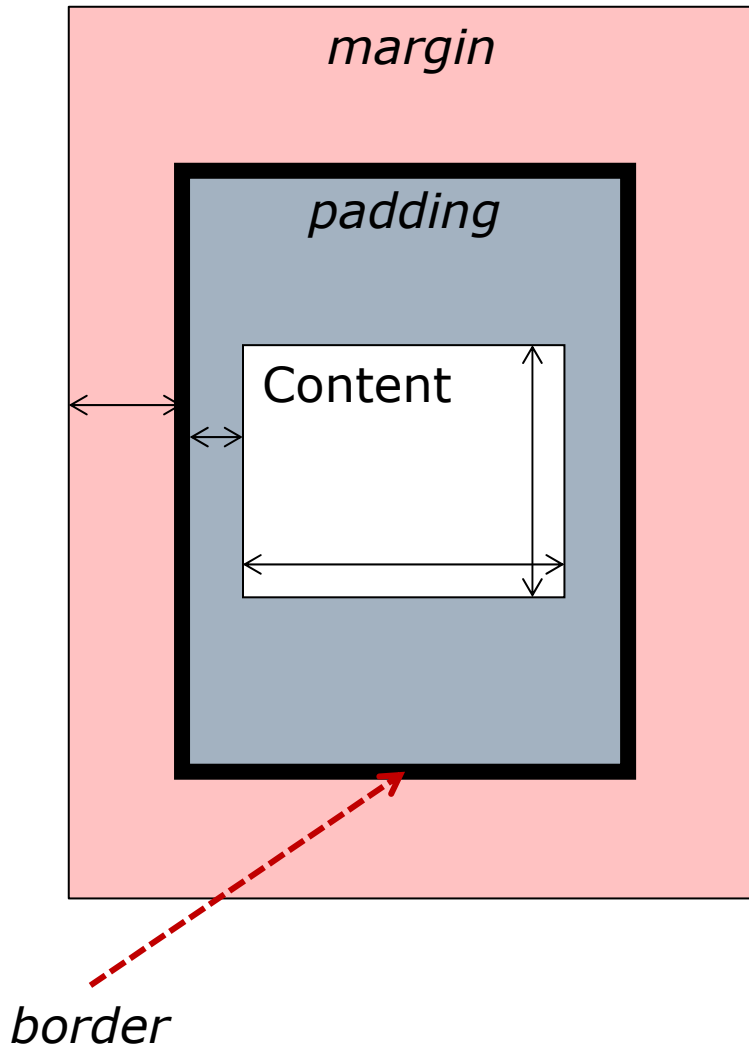
# Inherited Properties



# Demo: Chrome Dev. Inspector

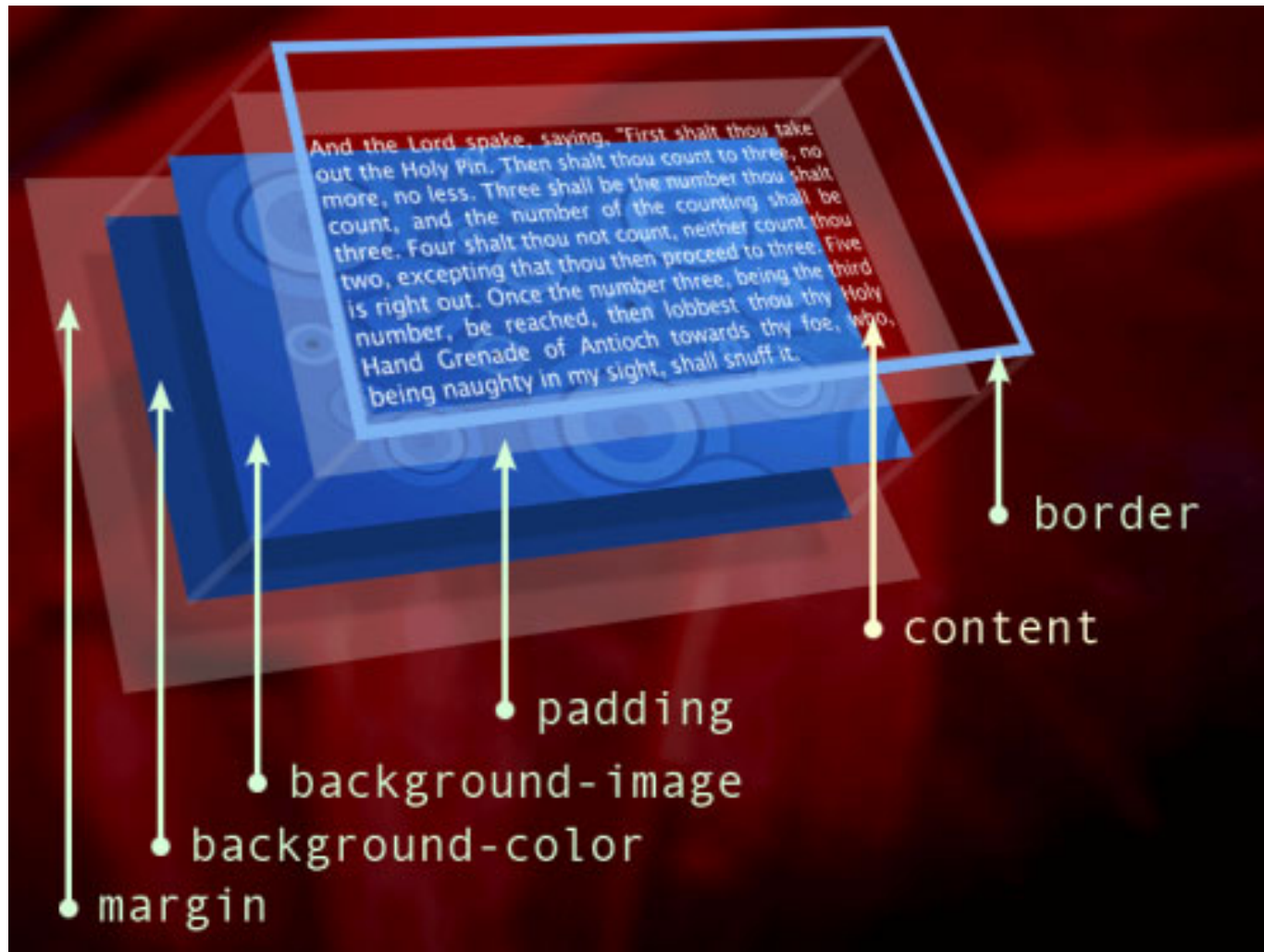


# Layout: The Box Model

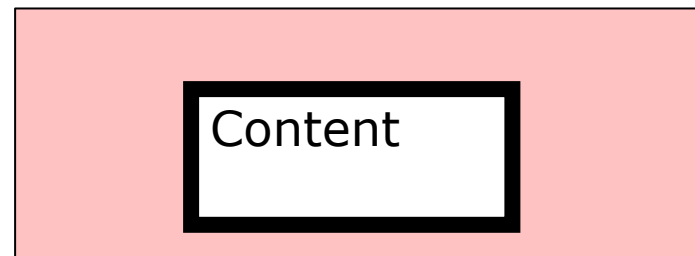
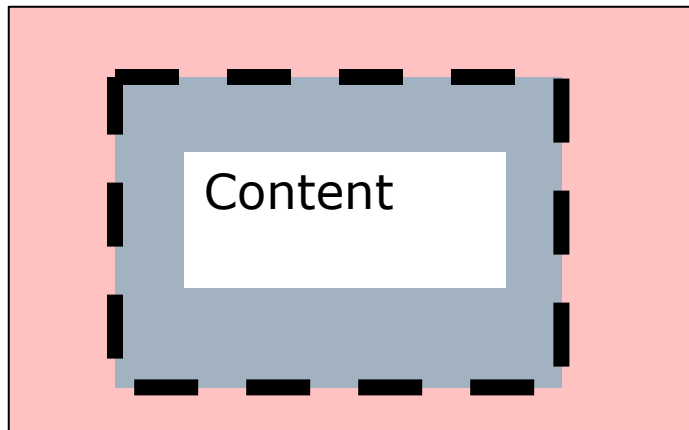
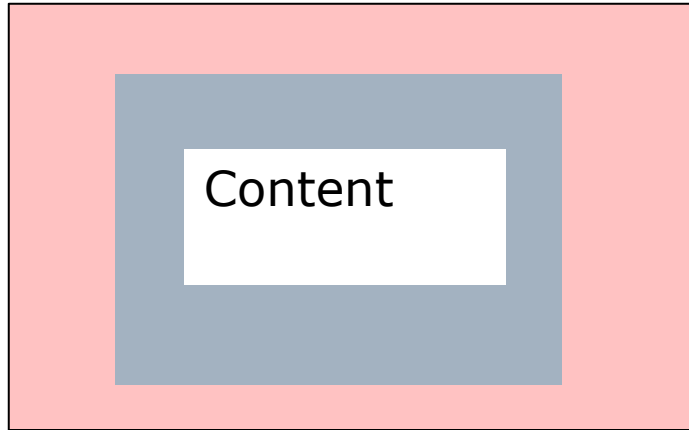


- Both block & inline
  - Minor differences
- Border appearance
  - Style, width, color, radius
- Margins & padding
  - Transparent
  - 4 independent sides
- Padding is *part* of it
  - *Content* background shows through
- Margins gives space
  - Some adjacent margins "collapse"

# The Box Model As Layers

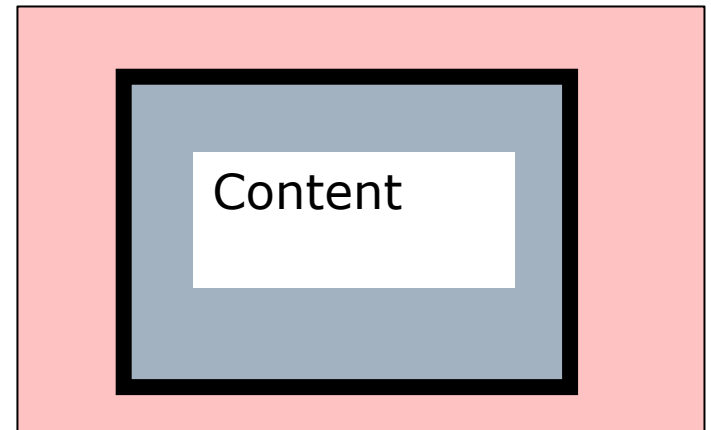


# Examples



# Box Sizing

```
p {  
  margin: 10px 100px 10px 10px;  
  border-width: 5px 1px 5px;  
  width: 200px;  
  padding: 2px;  
}
```



- ❑ Total width = ?
- ❑ CSS3 adds box-sizing
  - content-box (width sizes content only)
  - border-box (width includes border & padding)

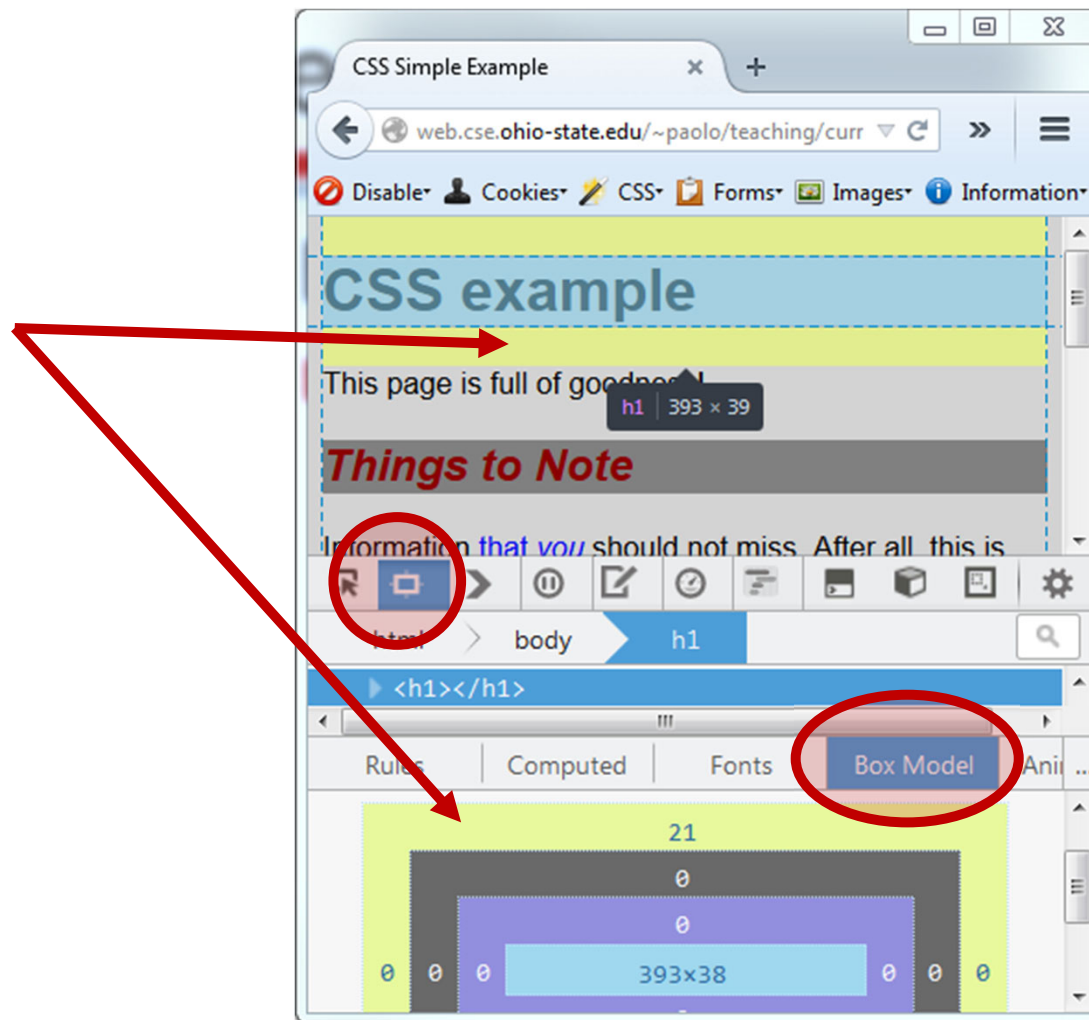








# Demo: Chrome Dev. Tools, Box



# Inheritance and Box Sizing

- Generally, text properties (eg color) are inherited, while box-related (eg border) are not
- (Content) width set by parent, child “fits” inside
  - Relative, absolute
- (Content) height set by child, parent “fits” around
  - Relative, absolute
- Parent and child's (vertical) margins collapse (if they touch)

# Summary

- CSS separates style from structure
  - Syntax: Rules with selectors, properties
  - Link to CSS file from HTML document
- Selectors for picking elements in tree
- Box Model
  - Content, padding, border, margin
  - Margins can collapse when overlapping
- Inheritance
  - Parent passes (font) properties to child
  - Box-related properties aren't inherited