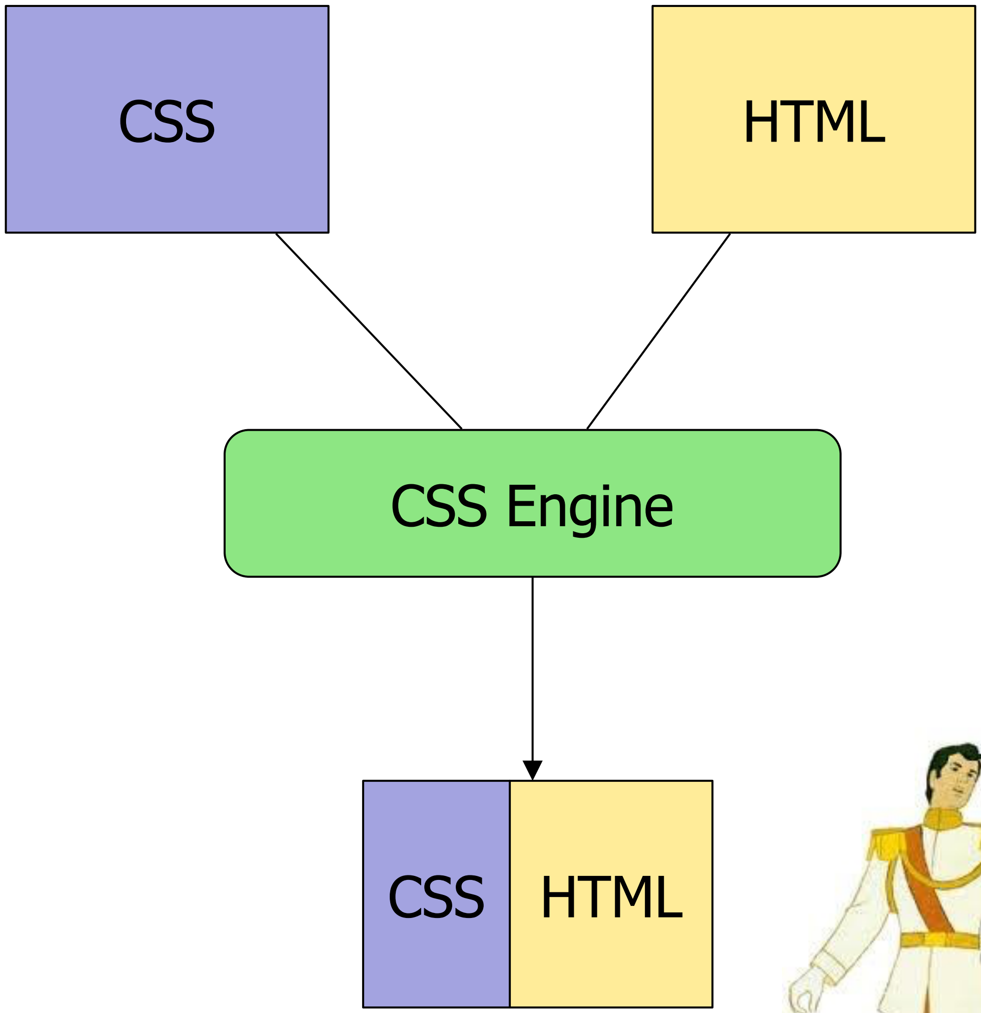
COMPSCI 326
Web Programming

Cascading Style Sheets

Introduction

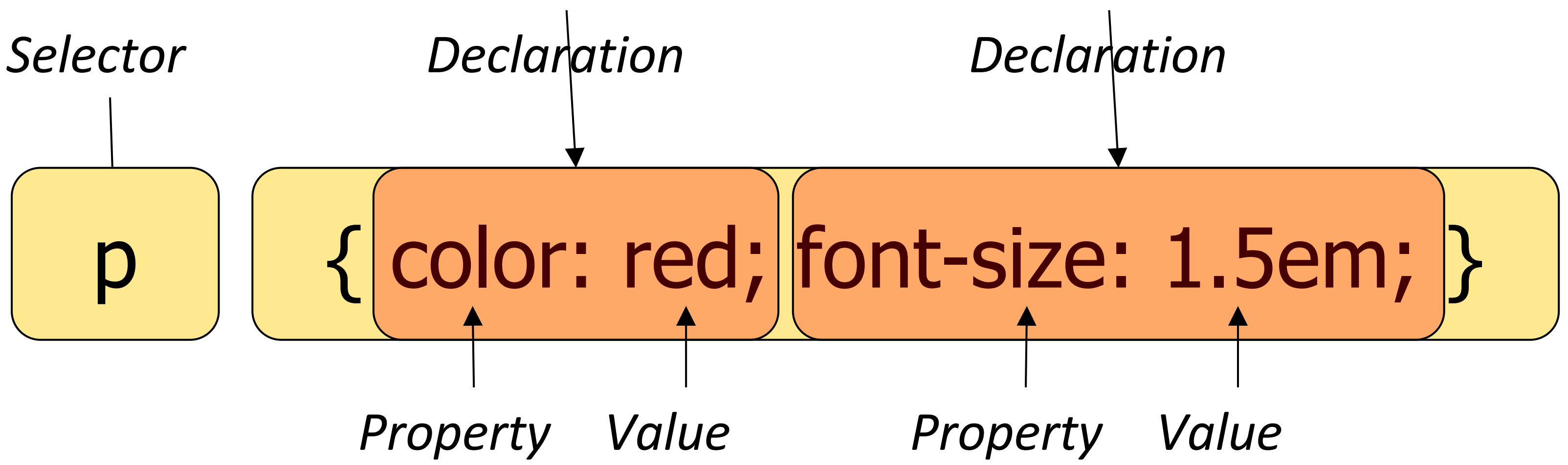
- What is CSS?
  - ✦ Style & Presentation
  - ✦ A Language to Manipulate HTML elements
- Why is it important?
  - ✦ Usability of HTML documents/user interfaces
  - ✦ Separation of concerns
- How is it used?
  - ✦ Internal: within tags or <*style*> element
  - ✦ External: imported with <*link*> element

CSS Usage



Anatomy of a CSS Style Rule

```
p {
    color: red;
    font-size: 1.5em;
}
```



Make all paragraphs have a font color of red and font size of 1.5em

Creating Styles & Style Sheets

- Where do we "put" style rules?
  - Internal
    - We can do this by embedding CSS in a *style attribute* string value or as character data within a *<style> element*.
  - External
    - We can "include" CSS rules using the *src* attribute of a *<link> element*.
    - **This is the better way.**

Internal Style Sheets

- HTML *style* Attribute
  - Syntax: <E style=“CSS declarations”> … </E>

Internal Style Sheets

- HTML *style* Attribute
  - ✦ Syntax: <E style="CSS declarations"> … </E>
- Example:

```
…
<body>
  <p style="color: red; font-size: 1.5em;">
    The rule is, jam tomorrow and jam
    yesterday - but never jam today.
  </p>
</body>
…
```

Internal Style Sheets

- HTML <*style*> Element
    - ✦ Syntax: <style type="text/css"> … </style>
    - ✦ Goes inside the <*head*> element

Internal Style Sheets

- HTML <*style*> Element
  - ✦ Syntax: <style type="text/css"> … </style>
  - ✦ Goes inside the <*head*> element
- Example:

```
…
<head>
  <style type="text/css">
    p { color: red; font-size: 1.5em; }
  </style>
</head>
<body>
  <p>The rule is, jam tomorrow and jam
     yesterday - but never jam today.</p>
</body>
…
```

External Style Sheets

- HTML <*link*> element
  - ✦ Syntax:
    <link rel=“stylesheet” src=“filename.css” type=“text/css” />
  - ✦ Goes inside the <*head*> element

External Style Sheets

- HTML <*link*> element
  - ✦ Syntax:
    <link rel="stylesheet" src="filename.css" type="text/css" />
  - ✦ Goes inside the <*head*> element
- Example:

index.html

```
<head>
  <link rel="stylesheet"
        src="main.css"
        type="text/css">
</head>
<body>
  <p>The rule is, jam tomorrow and jam
     yesterday - but never jam today.</p>
</body>
```

main.css

```
p {
  color: red;
  font-size: 1.5em;
}
```

Selectors

- What is a selector
  - Indicates the *element* or *elements* of a page to style
  - They can be broad
    - Apply to all elements of a particular kind
  - Or they can be specific
    - Apply to an element with a particular name

Identifying What to Style

- Page Wide Styling
  - ✦ CSS rules that apply to every occurance of an HTML element in the document

```
p {
    color: red;
    font-size: 1.5em;
}
```

Identifying What to Style

- Page Wide Styling
  - ✦ CSS rules that apply to every occurance of an HTML element in the document

```
p {
    color: red;
    font-size: 1.5em;
}
```

Perhaps we want to be a little more specific…

Styling Classes of Tags

- Class Selectors
  - ✦ Allow you to style elements that belong to a group or serve some special purpose.

CSS

```
.important {
   color: red;
   font-size: 75px;
}
```

HTML

```
<h2 class="important">
Headlines
</h2>
…
<h2 class="important">
Birthdays
</h2>
…
```

Styling Named Elements

- ID Selectors
  - ✦ Allow you to style elements with a specific name or identifier.
  - ✦ Applies to ***only*** a single element.

CSS

```
#headlines {
   color: red;
   font-size: 75px;
}

#birthdays {
   color: green;
}
```

HTML

```
<h2 id="headlines">
Headlines
</h2>
…
<h2 id="birthdays">
Birthdays
</h2>
…
```

Styling Groups of Tags

- Style across element types
  - ✦ Same style information that you want to apply across many different tags

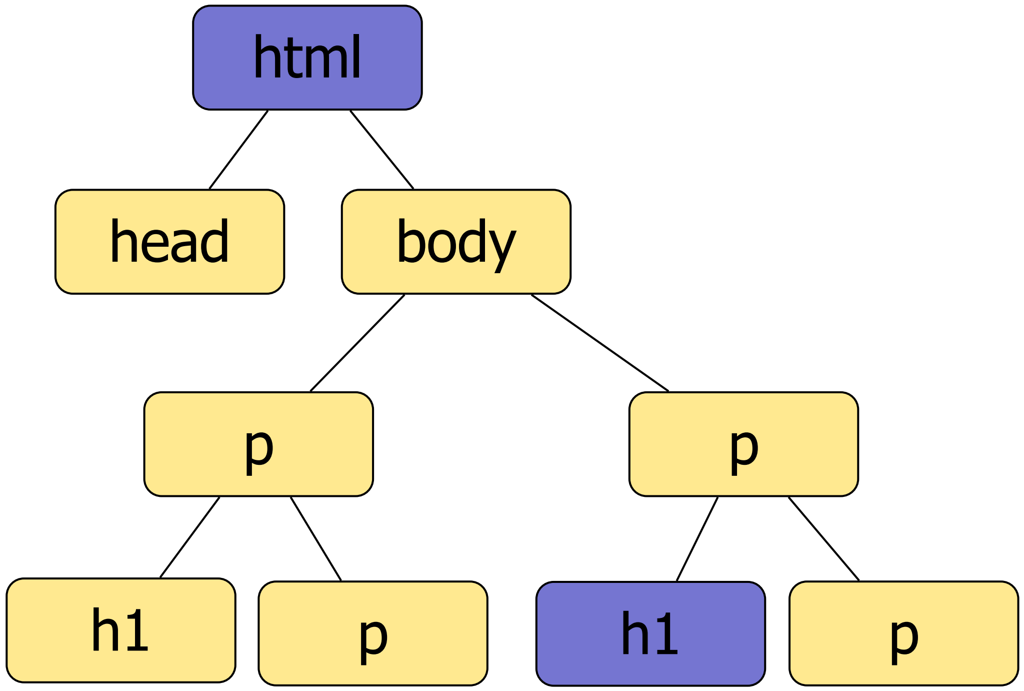
```
h1, h2, p, .hitem, #todo {
    color: #F134AC;
}


* {
    font-weight: bold;
}
```

Styling Tags within Tags

- What if you want to style elements that are *relative* to another element?
- You can use *descendant selectors*.
  - ✦ Ancestor: a tag that wraps another tag
  - ✦ Descendant: a tag inside one or more tags
  - ✦ Parent: the closest ancestor to another tag
  - ✦ Child: tag directly enclosed by another tag
  - ✦ Sibling: children of the same tag are siblings

Styling Tags within Tags

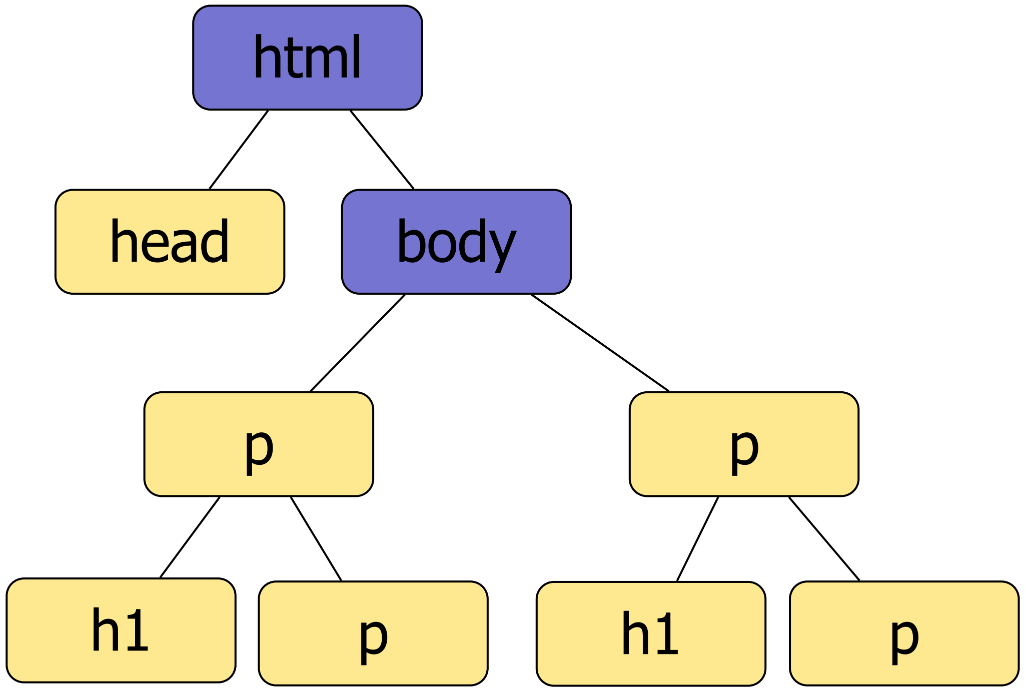


Ancestor Relationship

<html> is an ancestor of <h1>

In fact, <html> is an ancestor of all tags.

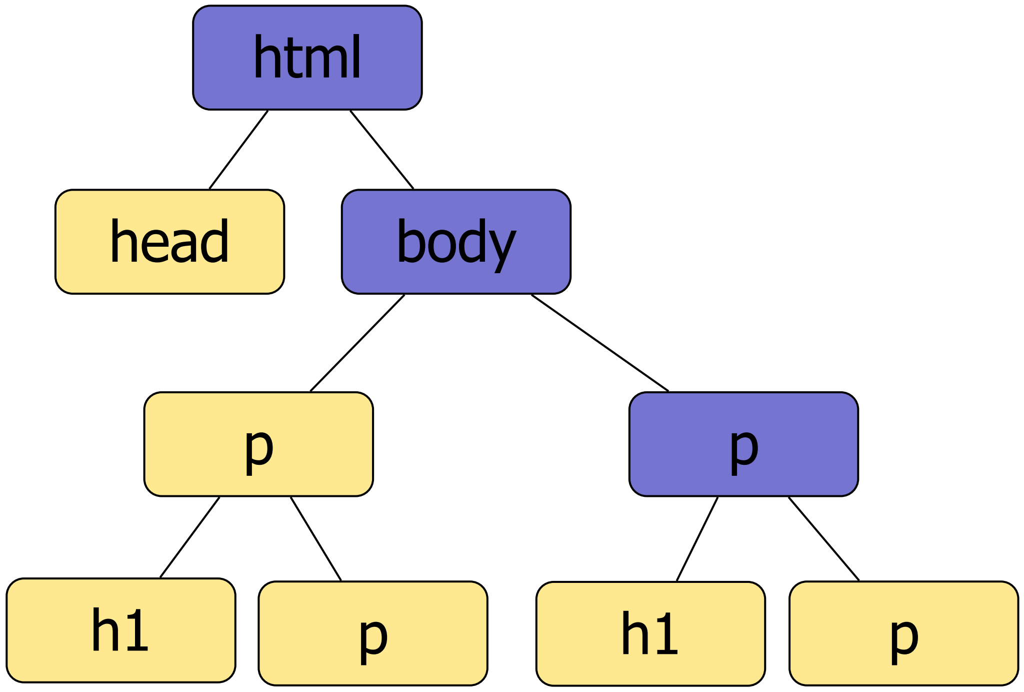
Styling Tags within Tags



Descendant Relationship

The <body> tag is a descendant of the <html> tag.

Styling Tags within Tags

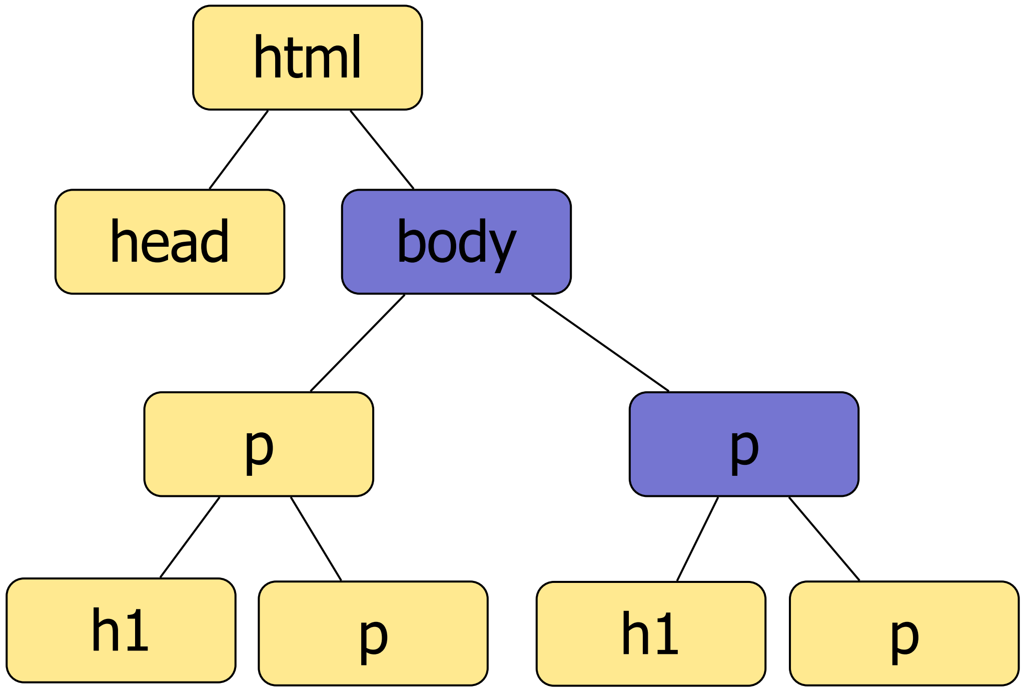


Descendant Relationship

The <body> tag is a descendant of the <html> tag.

The <p> tag is a descendant of both the <body> and the <html> tags.

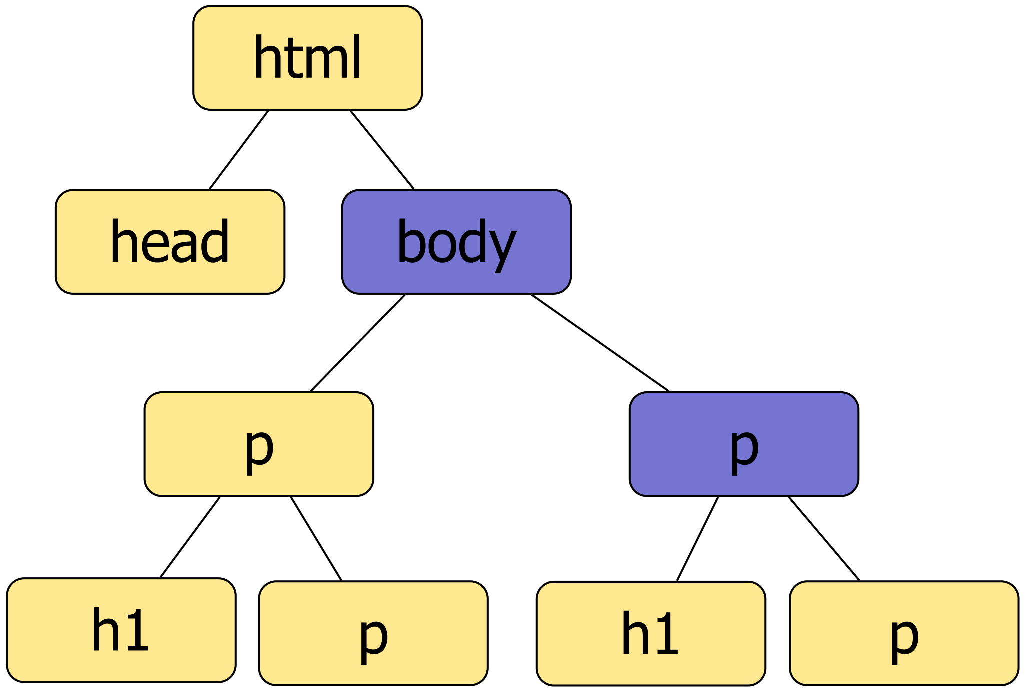
Styling Tags within Tags



Parent Relationship

The <body> tag is the parent of this <p> tag.

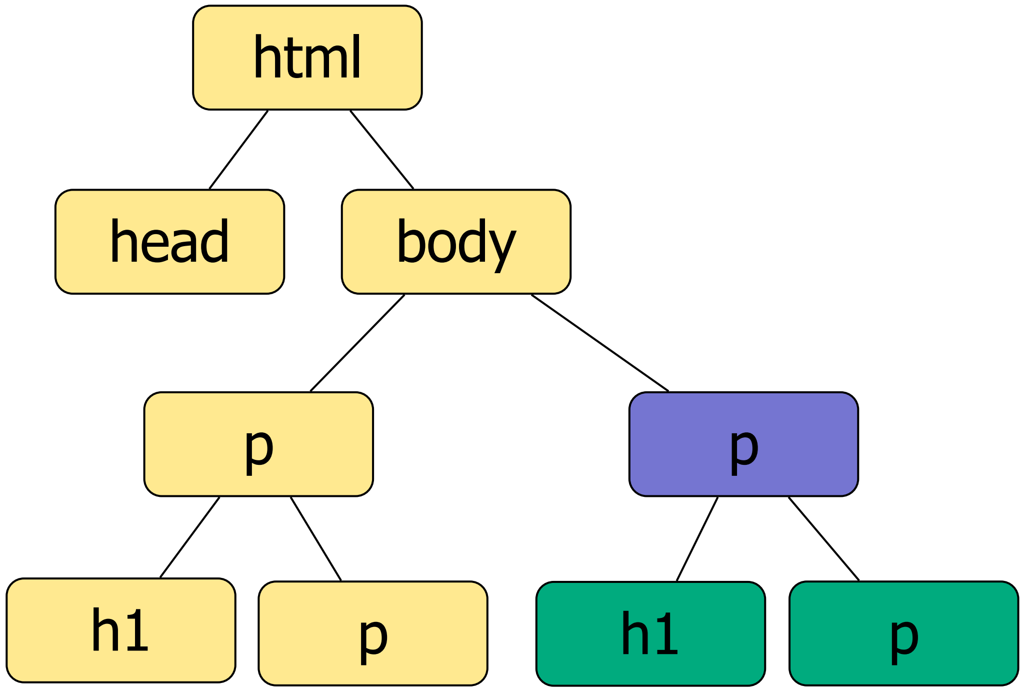
Styling Tags within Tags



Child Relationship

This <p> tag is the child of the <body> tag.

Styling Tags within Tags

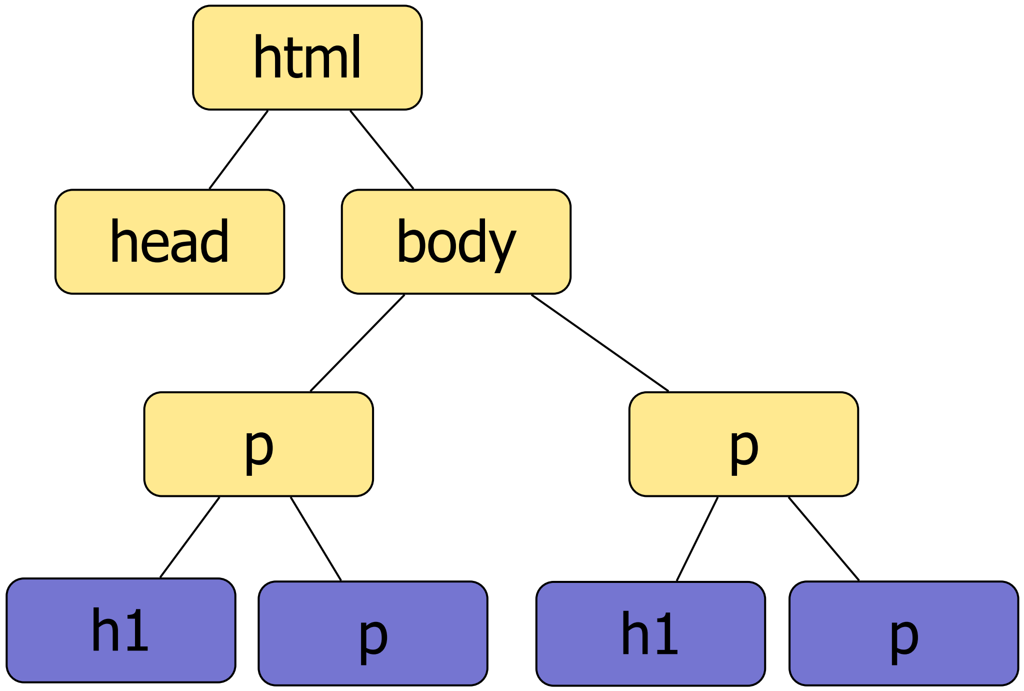


Sibling Relationship

These <h1> and <p> tags are siblings of each other.

They are both *children* of the containing <p> tag.

Styling Tags within Tags



Building Descendant Selectors

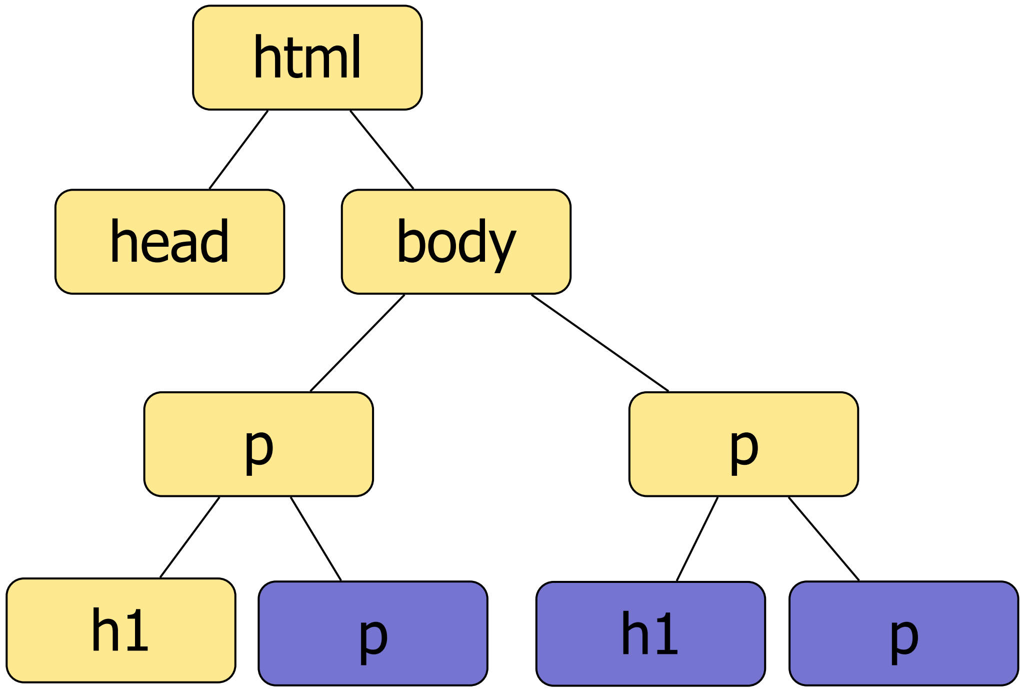
Make all paragraphs that are descendants of a <p> tag red.

Make all <h1> tags inside a <p> tag green.

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

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

Styling Tags within Tags



Building Descendant Selectors

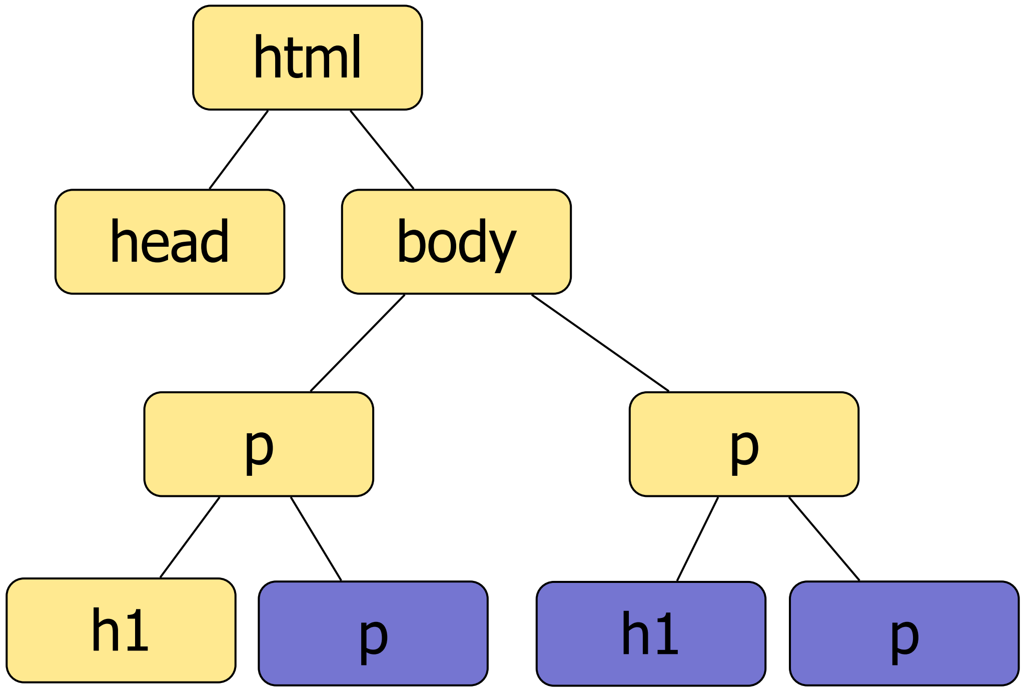
Make all paragraphs that are descendants of a <p> tag red.

Make all <h1> tags inside a <p> tag green. Only <h1> tags that are of the class "emphasize".

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

```
p h1.emphasize {
    color: green;
}
```

Styling Tags within Tags



Building Descendant Selectors

Make all paragraphs that are descendants of a <p> tag red.

Make all <h1> tags inside a <p> tag green. Only <h1> tags that are of the class "emphasize".

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

```
p h1.emphasize {
    color: green;
}
```

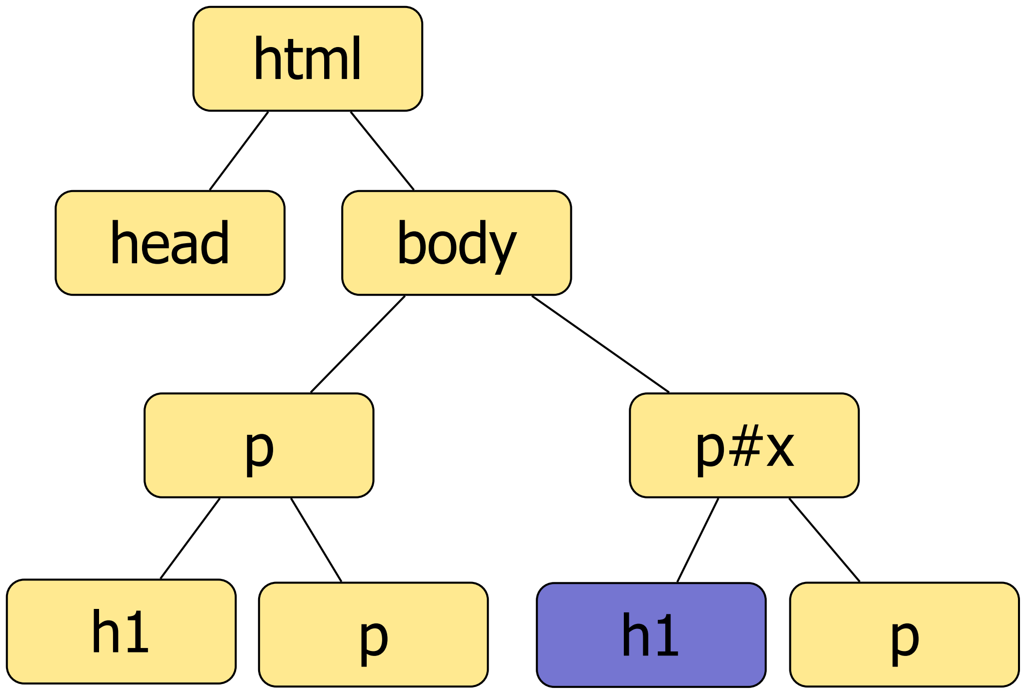
```
p h1 .emphasize {
    color: green;
}
```

What would this do?

Child Selectors

- What if we want to style a *direct* descendant of an element?
  - ✦ Use child selectors

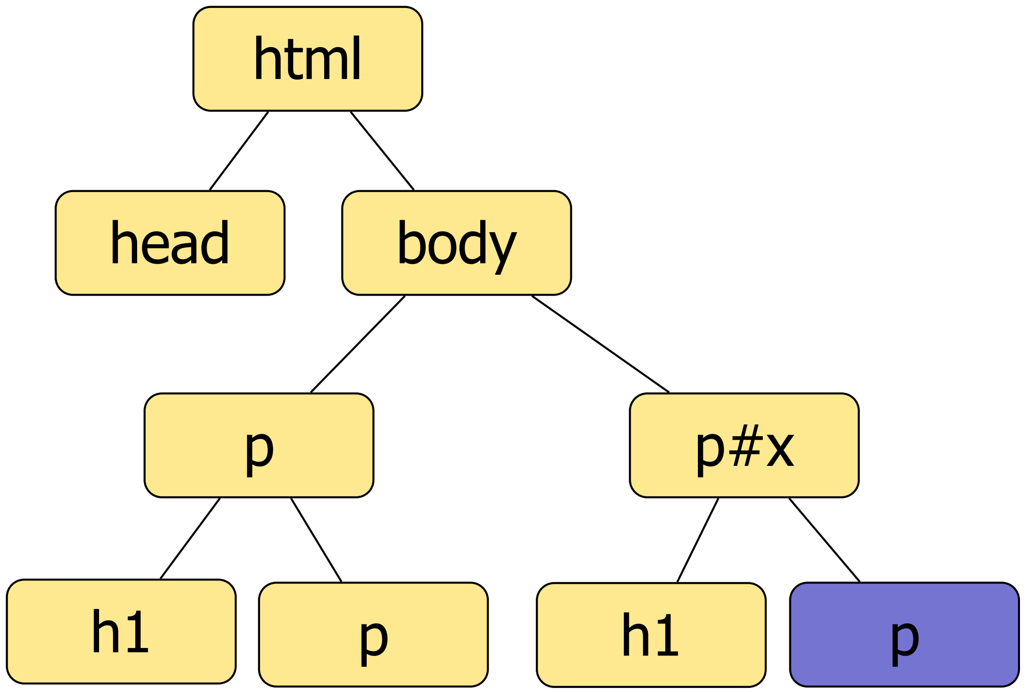
```
body > p#x > h1 {
   color: red;
}
```



Sibling Selectors

- What if we want to style a *sibling* of an element?
  - ✦ Use sibling selectors

```
body > p#x > h1 + p {
   color: red;
}
```



Attribute Selectors

- Perhaps we care about elements with particular attributes.
  - img[title]
    - images with a title attribute
  - a[href="http://google.com"]
    - links to google
  - input[type="text"]
    - text input boxes

Attribute Selectors

- Perhaps we care about elements with particular attributes that match the beginning of some text value:
  - a[href^="http://"]
    - Links to external sites
  - a[href^="http://"], a[href^="https://"]
    - Regular and secure links to external sites

Attribute Selectors

- Perhaps we care about elements with particular attributes that match the end of some text value:
  - a[href$=“.pptx”]
    - Links to external sites

```
a[href$=“.docx”] {
  background-image: url(docx.png) no-repeat;
  padding-left: 15px;
}
```

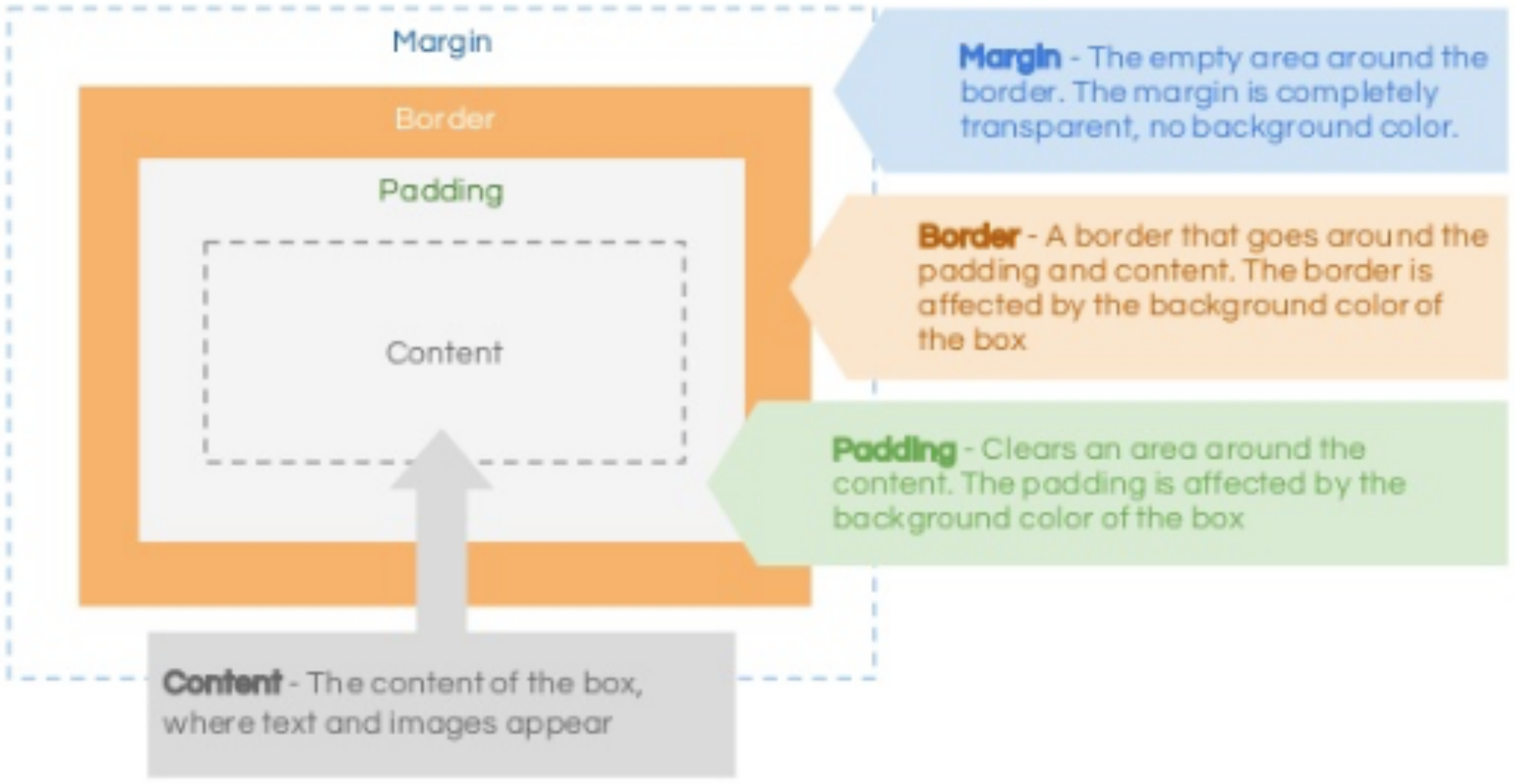
Attribute Selectors

- Perhaps we care about elements with particular attributes that match any part of some text value:
    - img[src*=“face”]
        - Any image containing “face” in its src value

Styling Content

- There are lots of ways to style content!
  - The size, color, and shape of text
  - The position of text
  - The margins and padding of text
  - The font family
- To many possibilities to cover in class
  - See: Mozilla Developer Network

CSS Box Model


Margin
Border
Padding
Content
Margin - The empty area around the border. The margin is completely transparent, no background color.
Border - A border that goes around the padding and content. The border is affected by the background color of the box
Padding - Clears an area around the content. The padding is affected by the background color of the box
Content - The content of the box, where text and images appear

Examples

HTML and CSS