Web Development

Bsc Applied, Forensics, Entertainment Systems, IOT

Eamonn de Leastar edeleastar@wit.ie)
Dr. Brenda Mullally (bmullally@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

CSS - Part 1

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Agenda

Use of CSS for styling

CSS

- Rules, Selectors, Declarations, Properties, Values
- CSS Binding
- Combining Rules & Selectors
- Html recap: attributes & Span elements
- Classes and class based styling
- Selector rules and the “Cascade” in CSS

Agenda

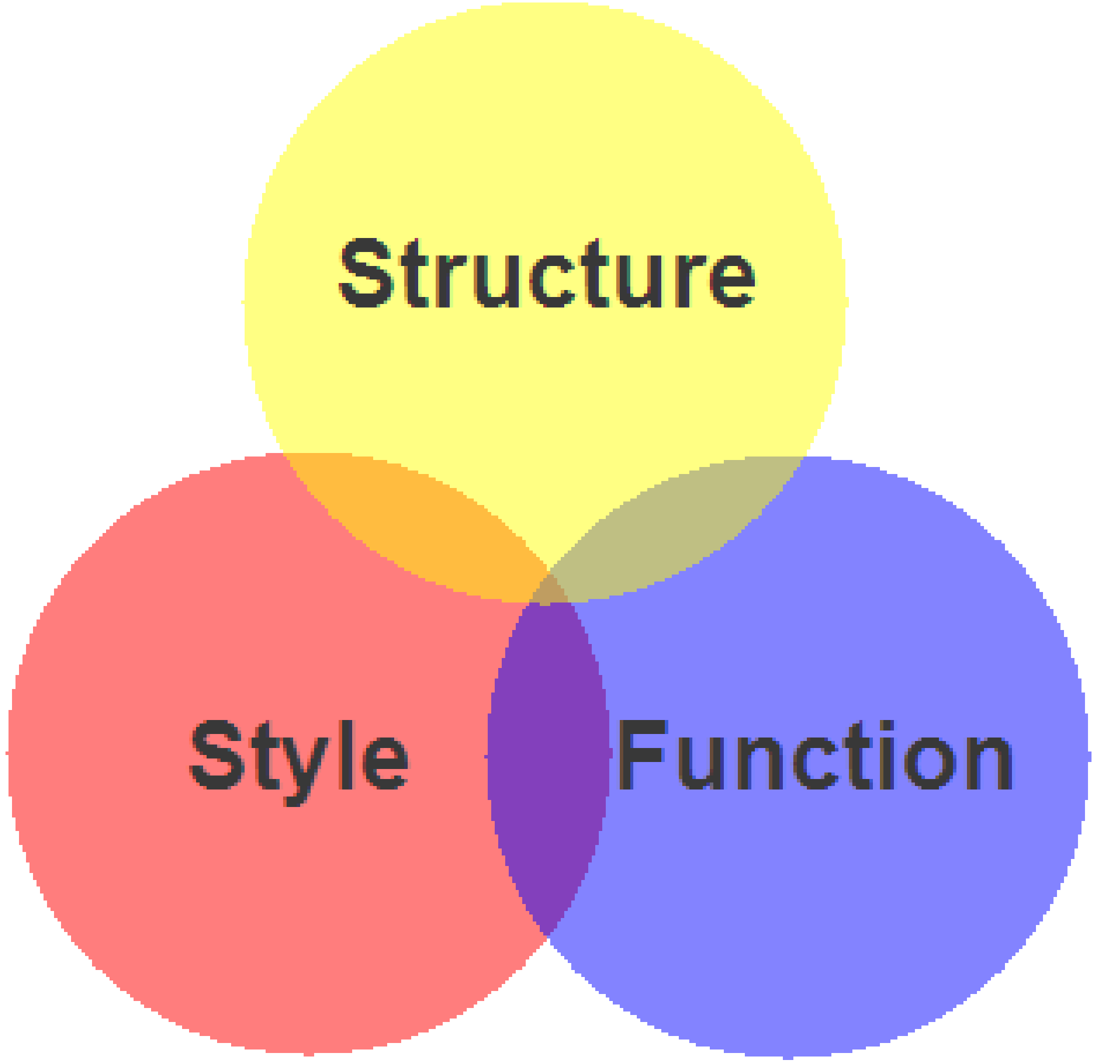
Use of CSS for styling

CSS

- Rules, Selectors, Declarations, Properties, Values
- Combining Rules & Selectors
- Classes and class based styling
- Selector rules and the “Cascade” in CSS

Markup, Style, Function

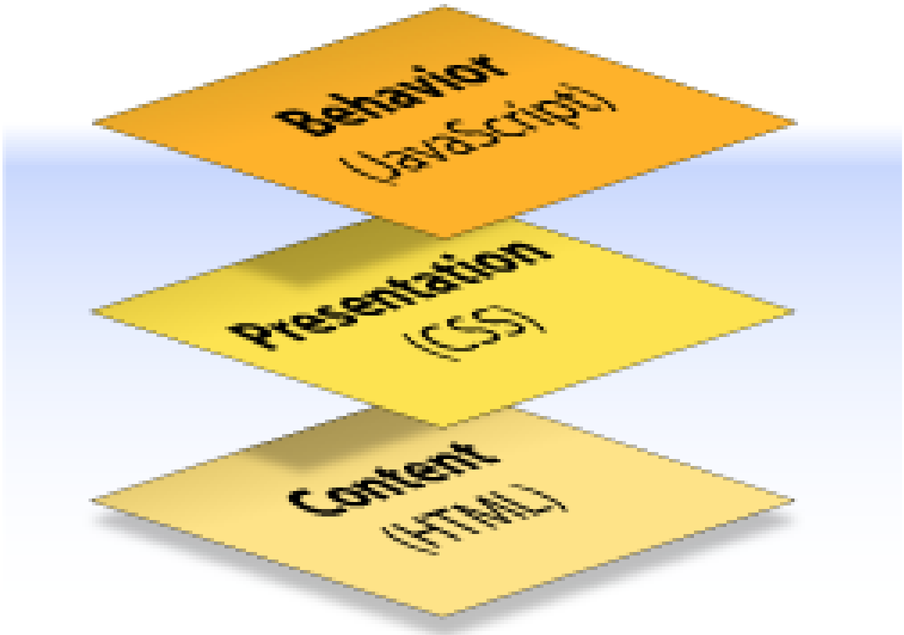
- Markup (XHTML, HTML)
  - Structure
  - Content
- Style (CSS)
  - Style
  - Presentation
  - Appearance
- Function (Javascript)
  - Actions
  - Manipulations



Content, Presentation, Behavior

- **Content** comprises the information the author wishes to convey to his or her audience, and is embedded within HTML or XHTML markup that defines its structure and semantics.

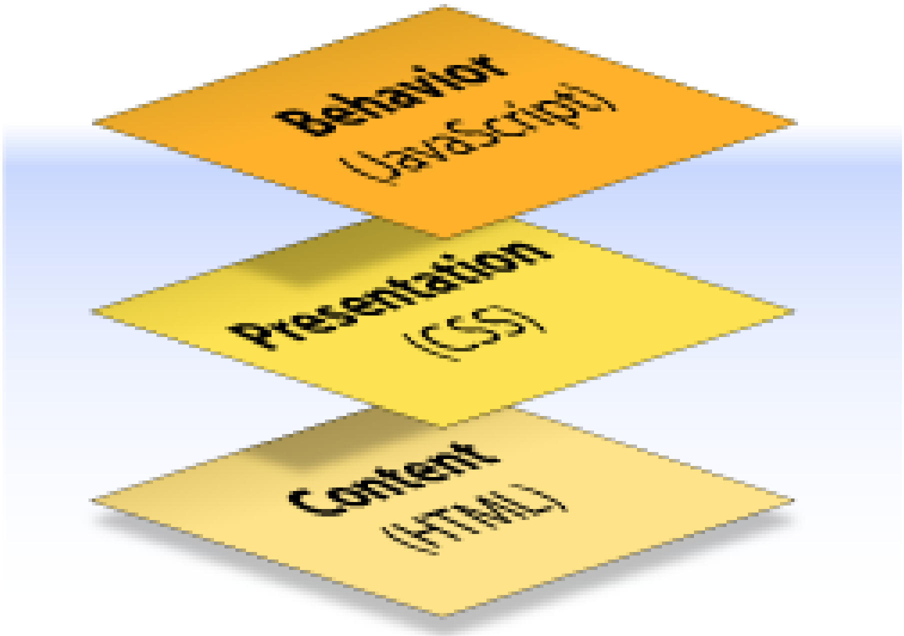
- Most of the content on the Web today is text, but content can also be provided through images, animations, sound, video, and whatever else an author wants to publish



Content, Presentation, Behavior

- **Presentation** defines how the content will appear to a human being who accesses the document in one way or another

- The conventional way to view a web page is with a regular web browser, of course, but that’s only one of many possible access methods. For example, content can also be converted to synthetic speech for users who have impaired vision or reading difficulties

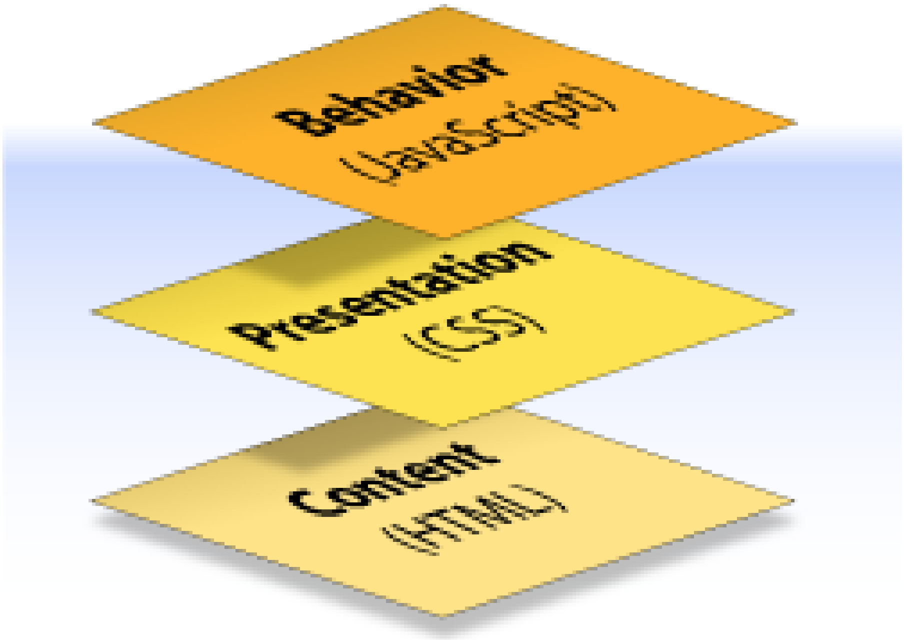


Content, Presentation, Behavior

- **Behavior** layer involves real-time user interaction with the document.

- This task is normally handled by JavaScript.

- The interaction can be anything from a trivial validation that ensures a required field is filled in before an order form can be submitted, to sophisticated web applications that work much like ordinary desktop programs.



Separation of Concerns

- It’s possible to embed all three layers within the same document
  - eg <em> and <b> can be used to control the presentation of text, and <hr> will insert a visible rule element.
  - Sometimes called *Presentational Markup*, these types of elements embed presentation-layer information within the content layer, they negate any advantage we may have gained by keeping the layers separate.
- Keeping them separate gives us one valuable advantage:
  - ***We can modify or replace any of the layers without having to change the others.***

What is CSS?

- Cascading Style Sheets is the recommended way to control the presentation layer in a web document.
- The main advantage of CSS over presentational HTML markup is that the styling can be kept entirely separate from the content.
- e.g. it’s possible to store all the presentational styles for a 10,000-page web site in a single CSS file.
- CSS also provides far better control over presentation than do presentational element types in HTML.

Advantages of CSS

- All styling is kept in a limited number of style sheets.

- Saving on bandwidth: The style sheet is cached (kept on client machine) after the first request and can be reused for every page on the site, it doesn’t have to be downloaded with each web page. Could reduce bandwidth usage—by more than 50% in many documented cases.

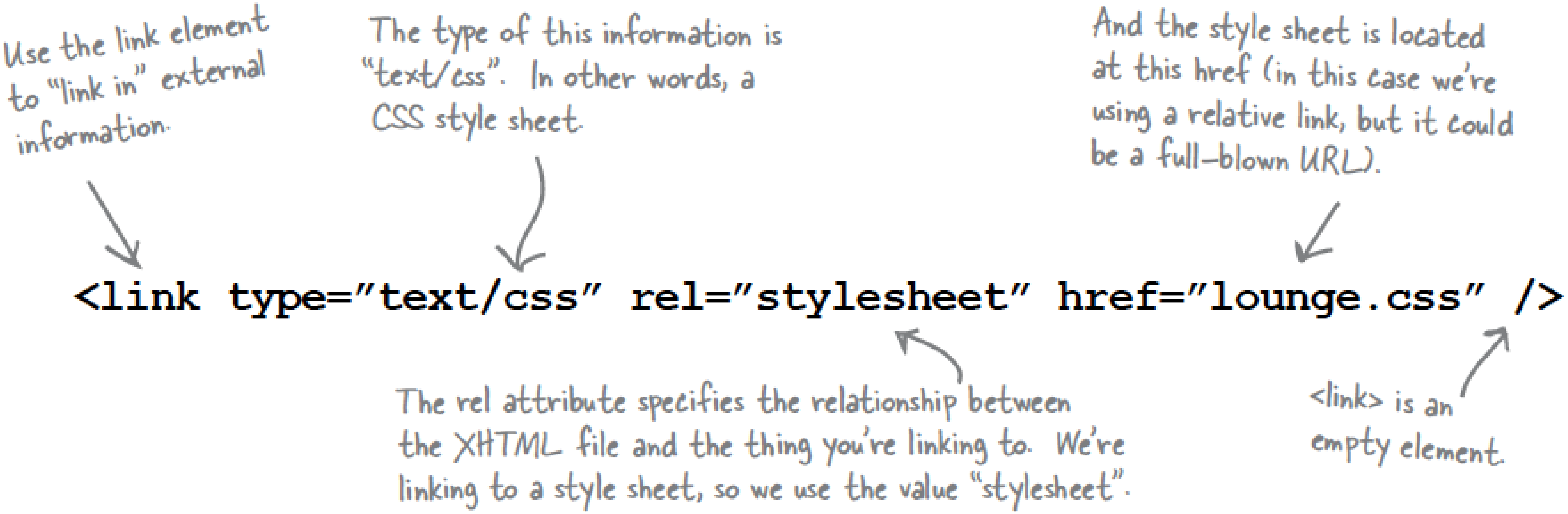
- The separation of content from presentation makes it easier for site owners to reuse the content for other purposes, such as RSS feeds or text-to-speech conversion.

- Separate styling rules can be used for different output media. We no longer need to create a special version of each page for printing—we can simply create a single style sheet that controls how every page on the site will be printed.

Separating Style Sheets from HTML

- Ideally, a site may have a single style sheet for most purposes. It is possible to include the css rules within your html files but this is not advisable.
- In a separate style.css file the rules are written.
- It is then necessary to link from the html file to the style.css file

<link> element



```
<link type="text/css" rel="stylesheet" href="lounge.css" />
```

Use the link element to "link in" external information.

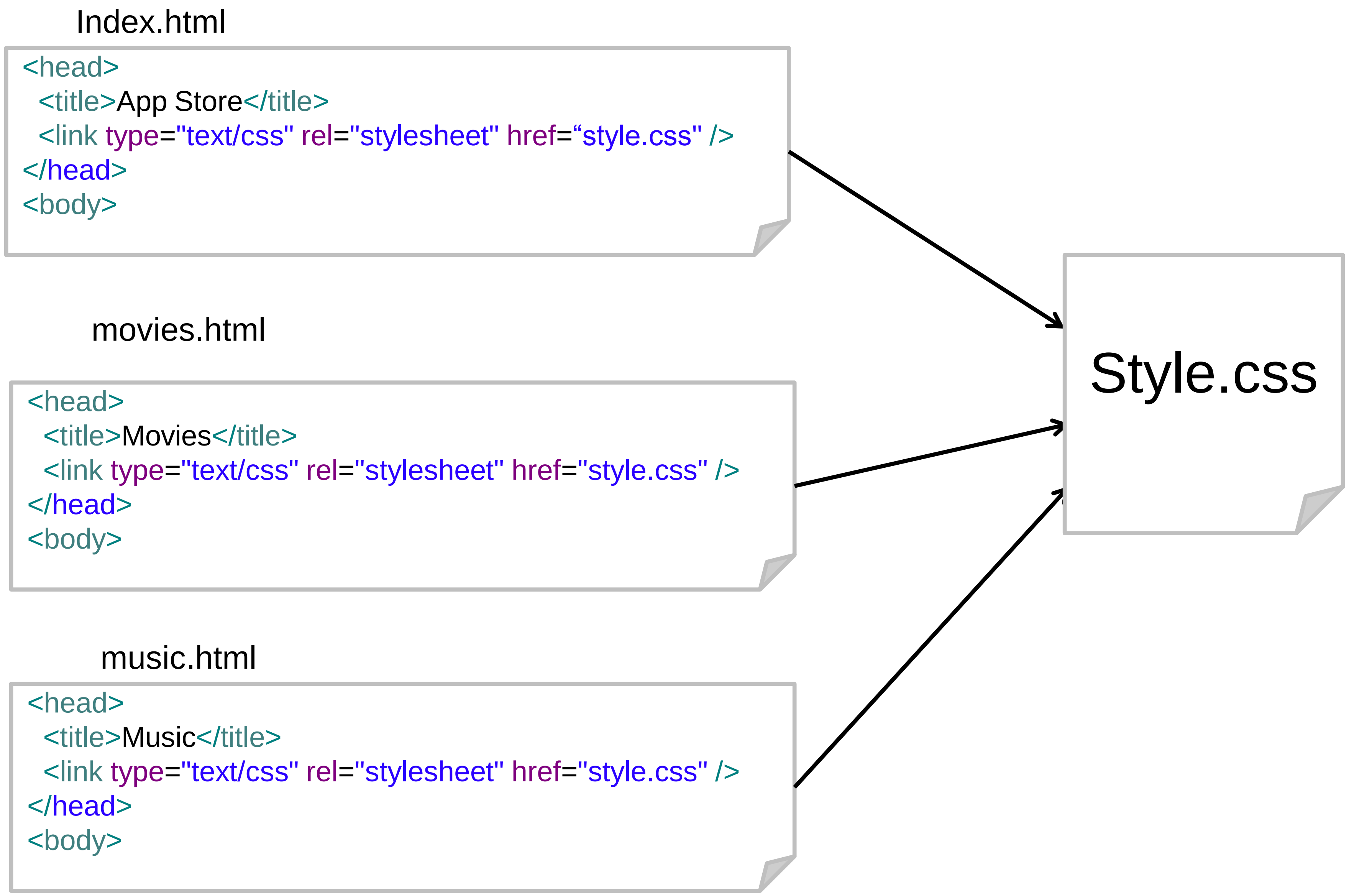
The type of this information is "text/css". In other words, a CSS style sheet.

And the style sheet is located at this href (in this case we're using a relative link, but it could be a full-blown URL).

The rel attribute specifies the relationship between the XHTML file and the thing you're linking to. We're linking to a style sheet, so we use the value "stylesheet".

<link> is an empty element.

One Stylesheet - Multiple html pages



Index.html

```
<head>
  <title>App Store</title>
  <link type="text/css" rel="stylesheet" href=“style.css" />
</head>
<body>
```

movies.html

```
<head>
  <title>Movies</title>
  <link type="text/css" rel="stylesheet" href="style.css" />
</head>
<body>
```

music.html

```
<head>
  <title>Music</title>
  <link type="text/css" rel="stylesheet" href="style.css" />
</head>
<body>
```

Style.css

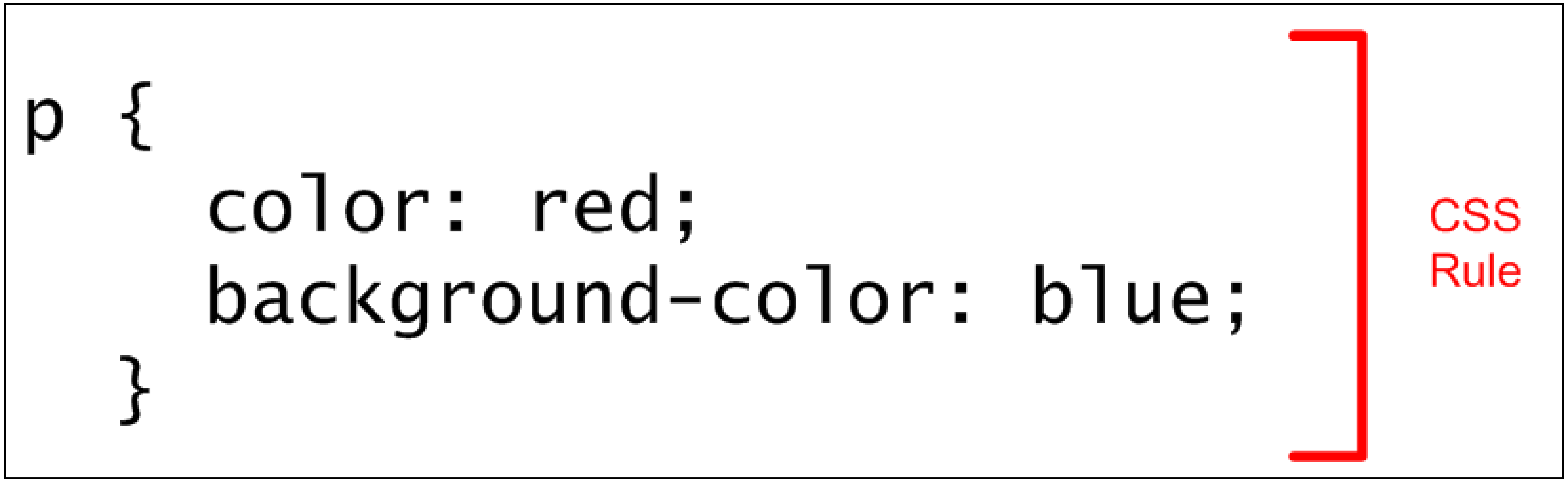
Agenda

Use of CSS for styling

CSS

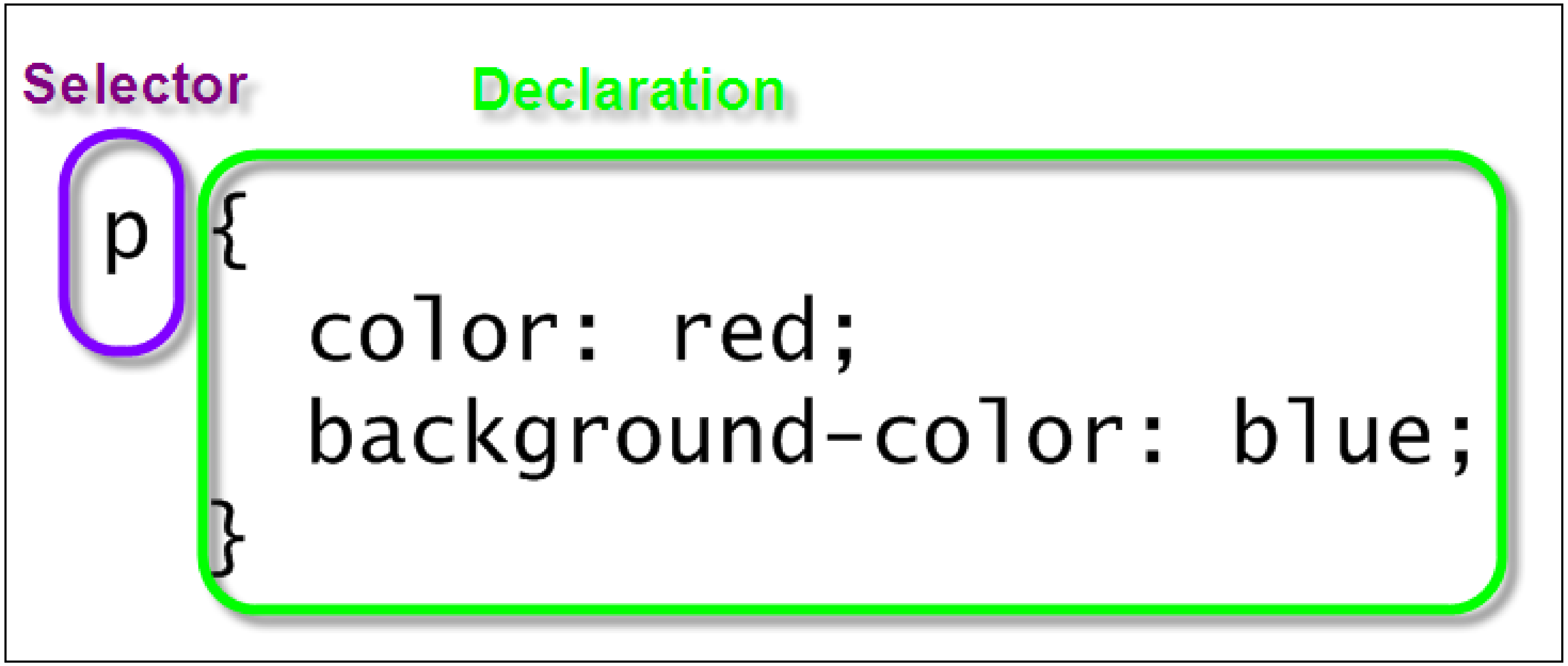
- Rules, Selectors, Declarations, Properties, Values
- Combining Rules & Selectors
- Classes and class based styling
- Selector rules and the “Cascade” in CSS

CSS Rule



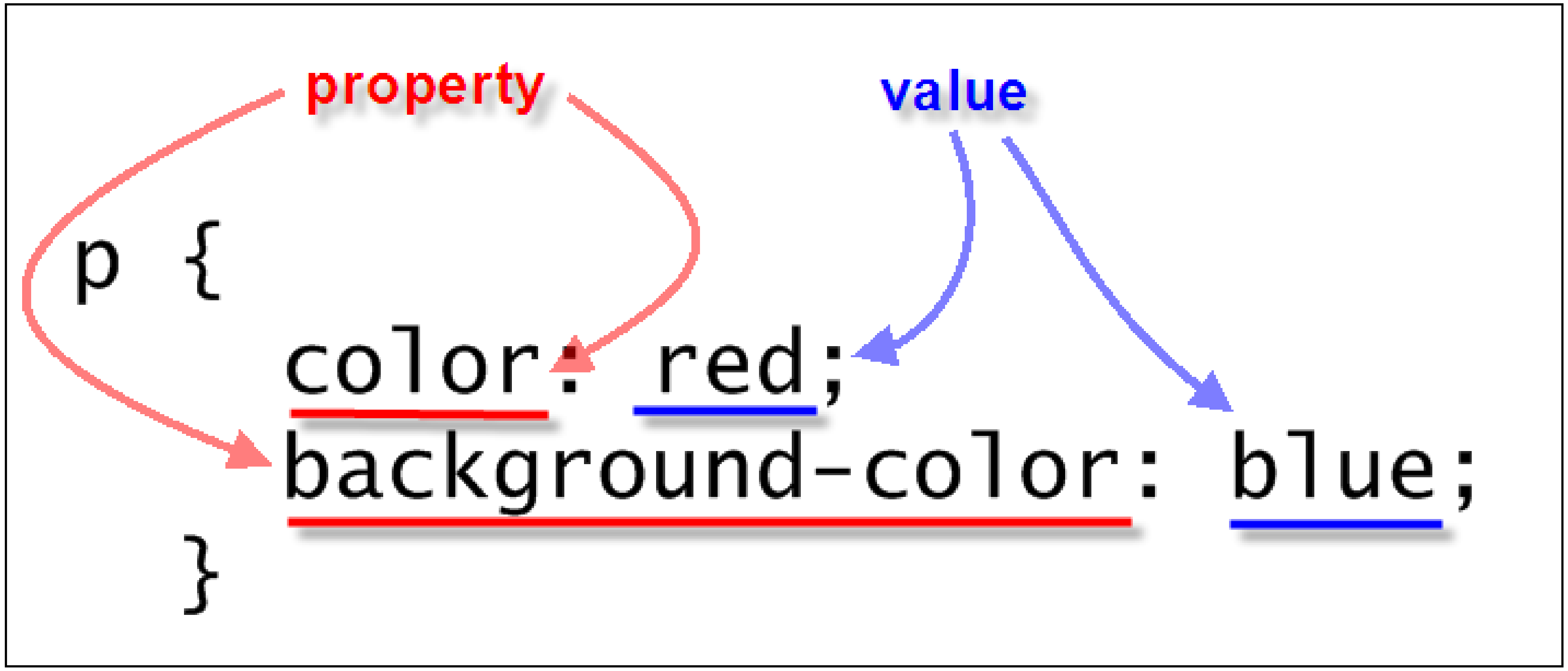
```
p {
    color: red;
    background-color: blue;
  }
```

Selector and Declarations



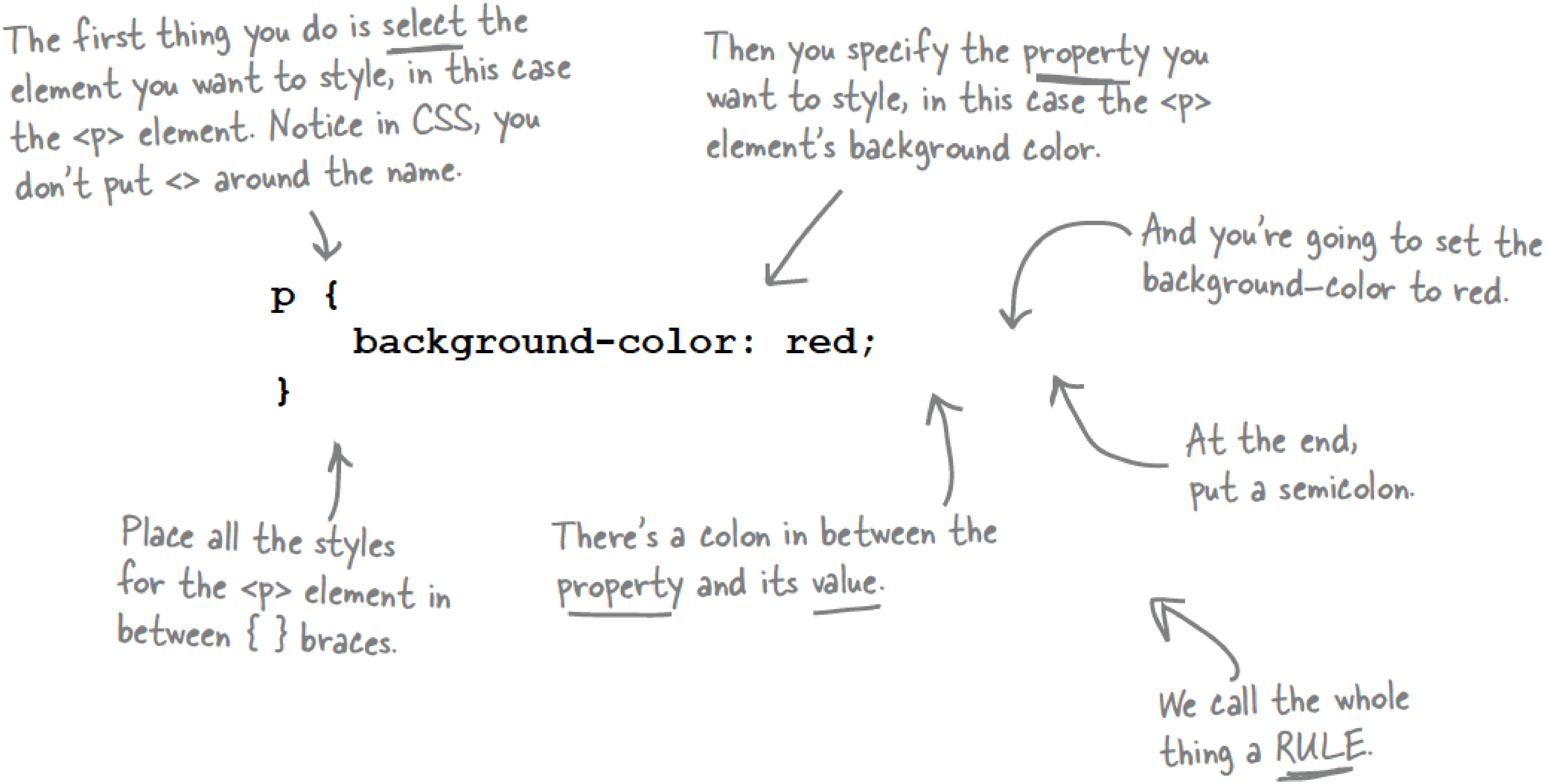
```
p {
    color: red;
    background-color: blue;
  }
```

Properties & Values



```
p {
    color: red;
    background-color: blue;
}
```

A CSS Rule



The first thing you do is select the element you want to style, in this case the <p> element. Notice in CSS, you don't put <> around the name.

Then you specify the property you want to style, in this case the <p> element's background color.

And you're going to set the background-color to red.

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

Place all the styles for the <p> element in between { } braces.

There's a colon in between the property and its value.

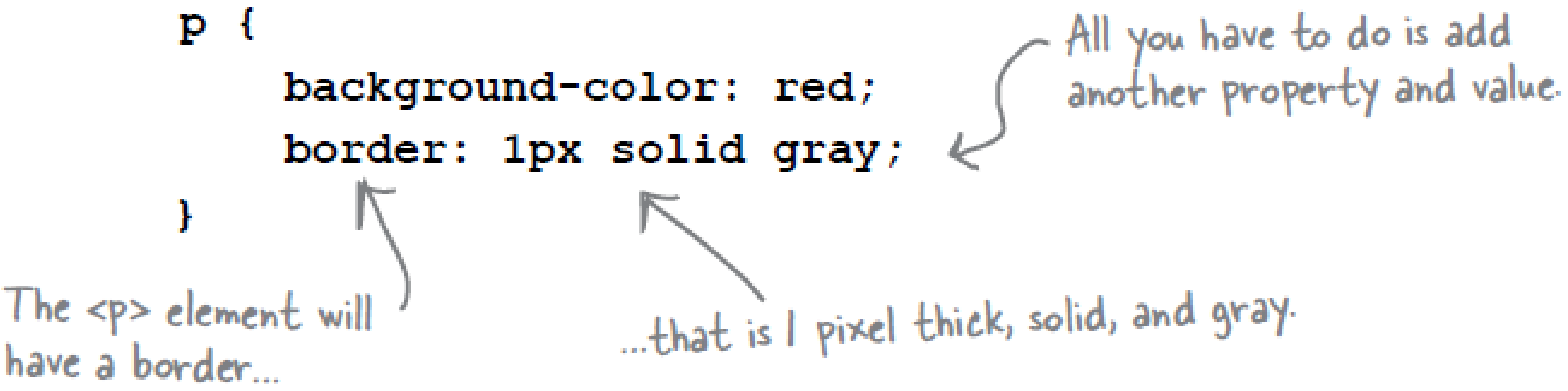
At the end, put a semicolon.

We call the whole thing a RULE.

More Properties...

- You can add as many properties and values as you like in each CSS rule.

- To put a border around your paragraphs:



```
p {
    background-color: red;
    border: 1px solid gray;
}
```

All you have to do is add another property and value.

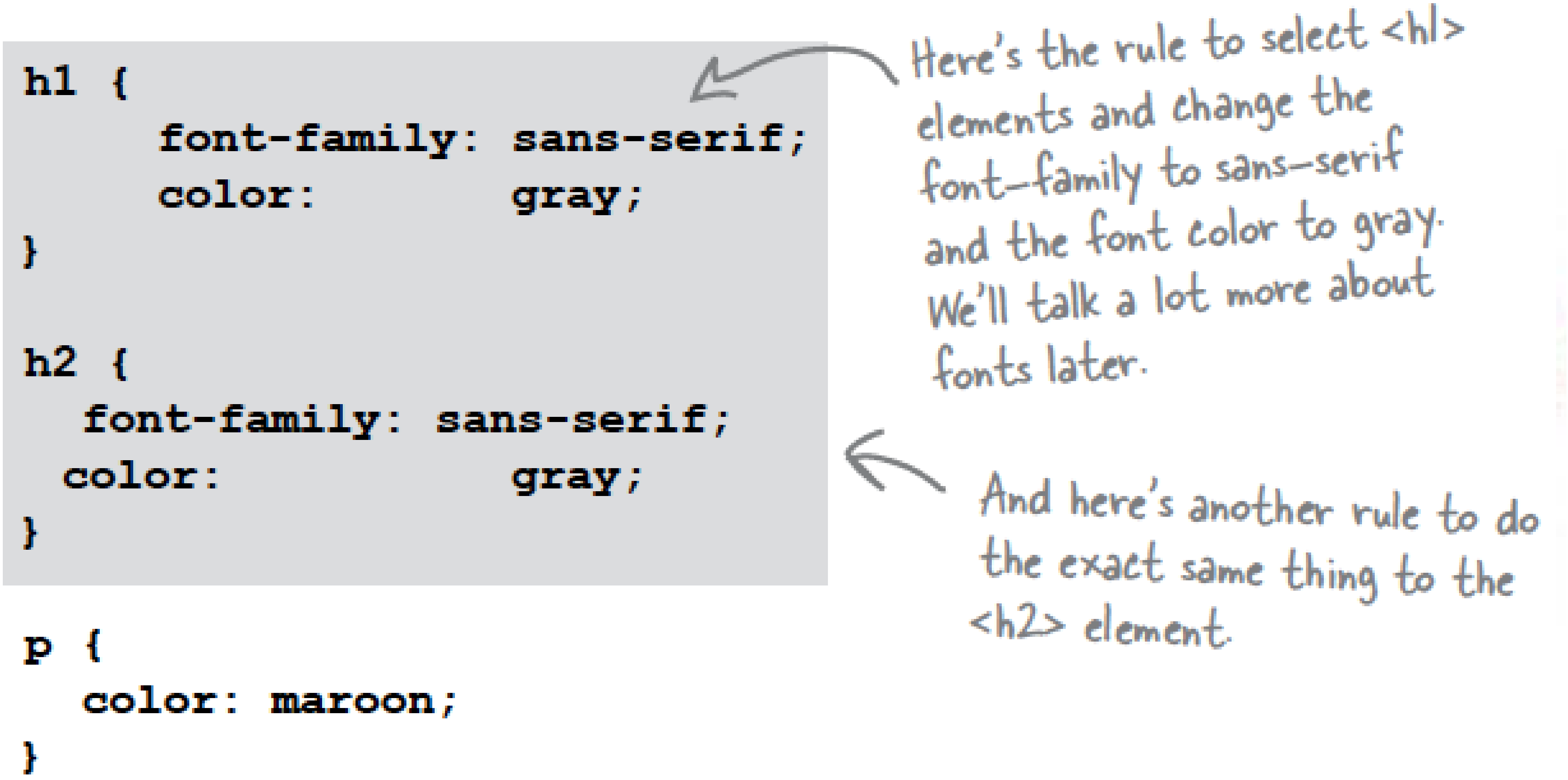
The <p> element will have a border...

...that is 1 pixel thick, solid, and gray.

Agenda

- CSS
  - Rules, Selectors, Declarations, Properties, Values
  - Combining Rules & Selectors
  - Classes and class based styling
  - Selector rules and the “Cascade” in CSS

Multiple Rules



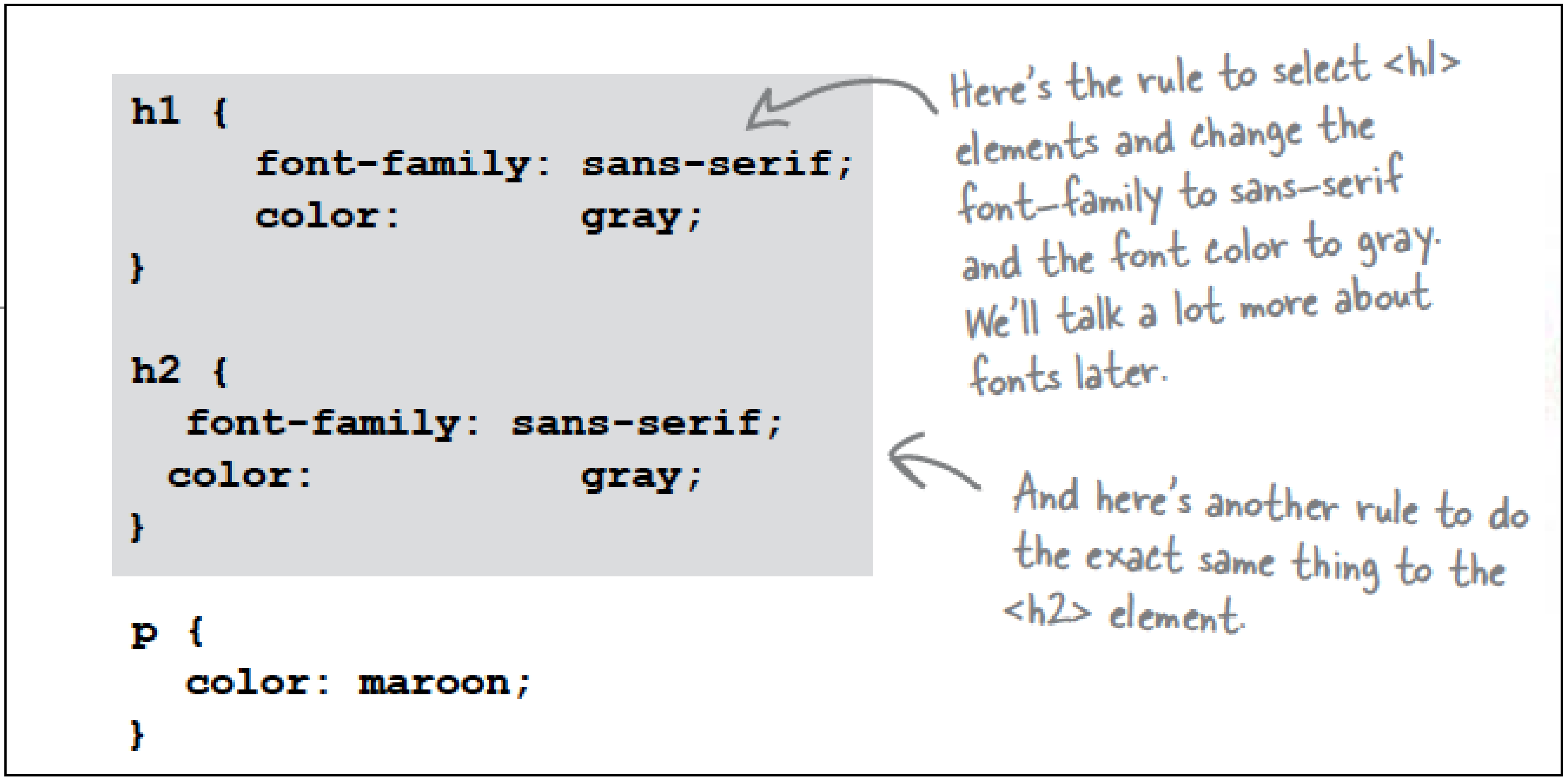
```
h1 {
      font-family: sans-serif;
      color:       gray;
}

h2 {
   font-family: sans-serif;
  color:              gray;
}

p {
   color: maroon;
}
```

Here's the rule to select <h1> elements and change the font-family to sans-serif and the font color to gray. We'll talk a lot more about fonts later.

And here's another rule to do the exact same thing to the <h2> element.

Combining Selectors (1)

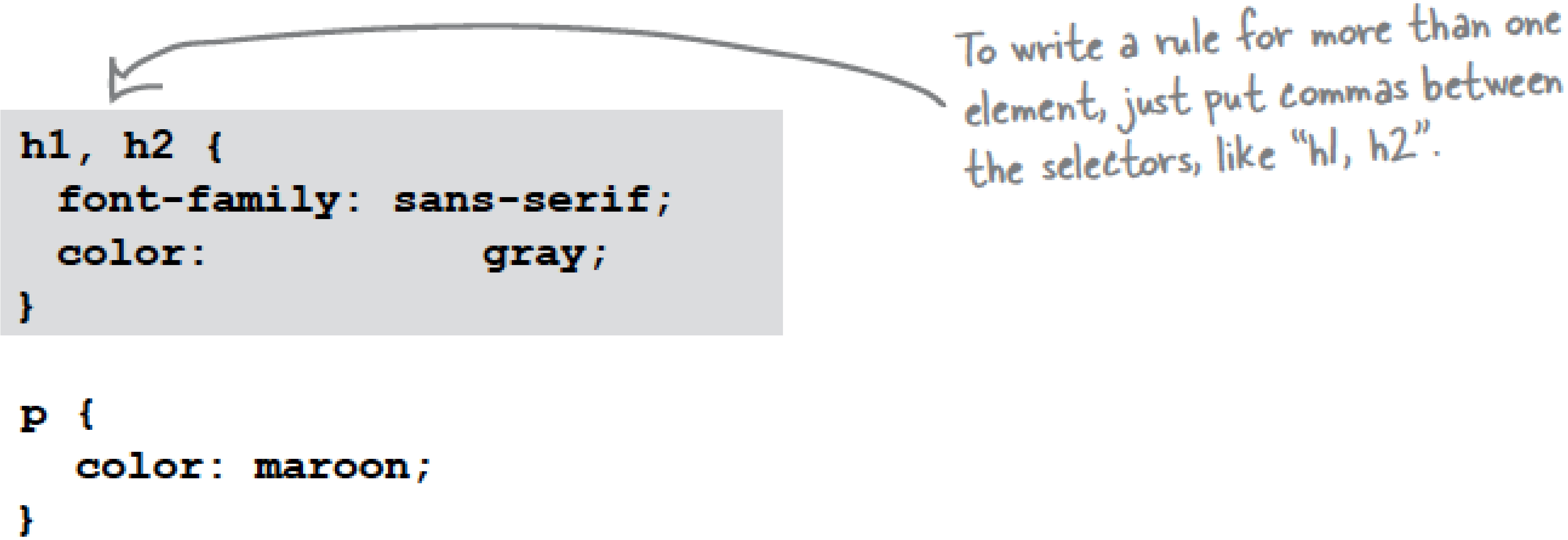
- Rules can be combined if they are identical



```
h1 {
      font-family: sans-serif;
      color:       gray;
}

h2 {
   font-family: sans-serif;
  color:           gray;
}

p {
   color: maroon;
}
```

Here's the rule to select <h1> elements and change the font-family to sans-serif and the font color to gray. We'll talk a lot more about fonts later.

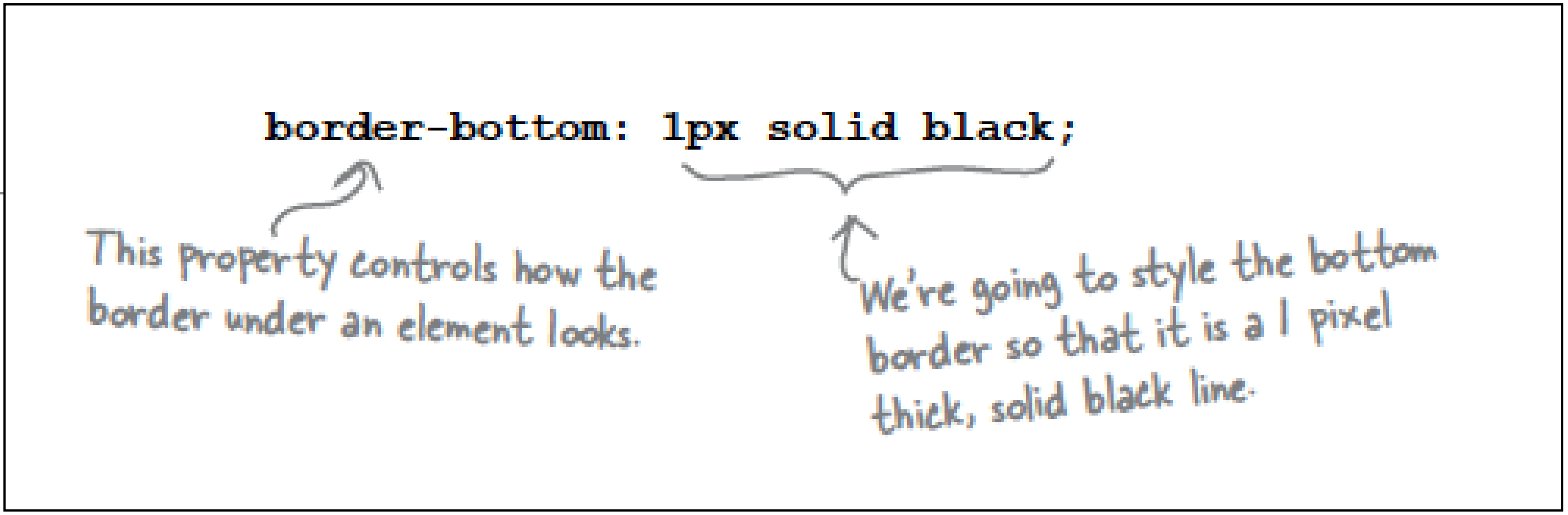
And here's another rule to do the exact same thing to the <h2> element.



```
h1, h2 {
  font-family: sans-serif;
  color:             gray;
}

p {
   color: maroon;
}
```

To write a rule for more than one element, just put commas between the selectors, like "h1, h2".

Border Styles

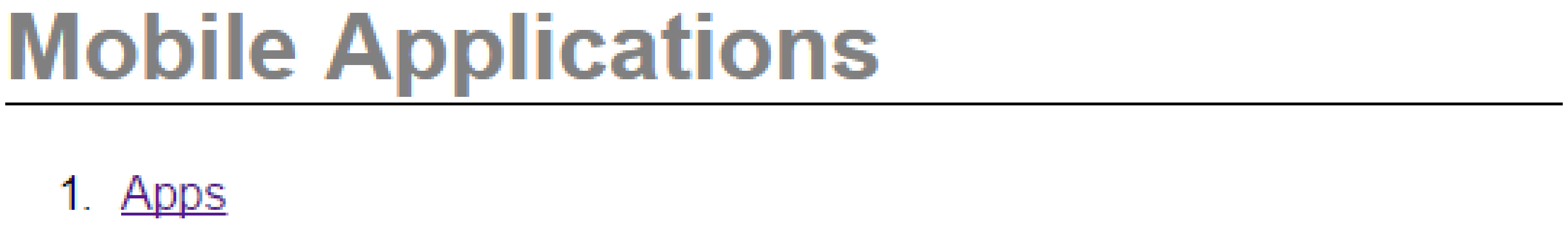


```
border-bottom: 1px solid black;
```

This property controls how the border under an element looks.

We're going to style the bottom border so that it is a 1 pixel thick, solid black line.

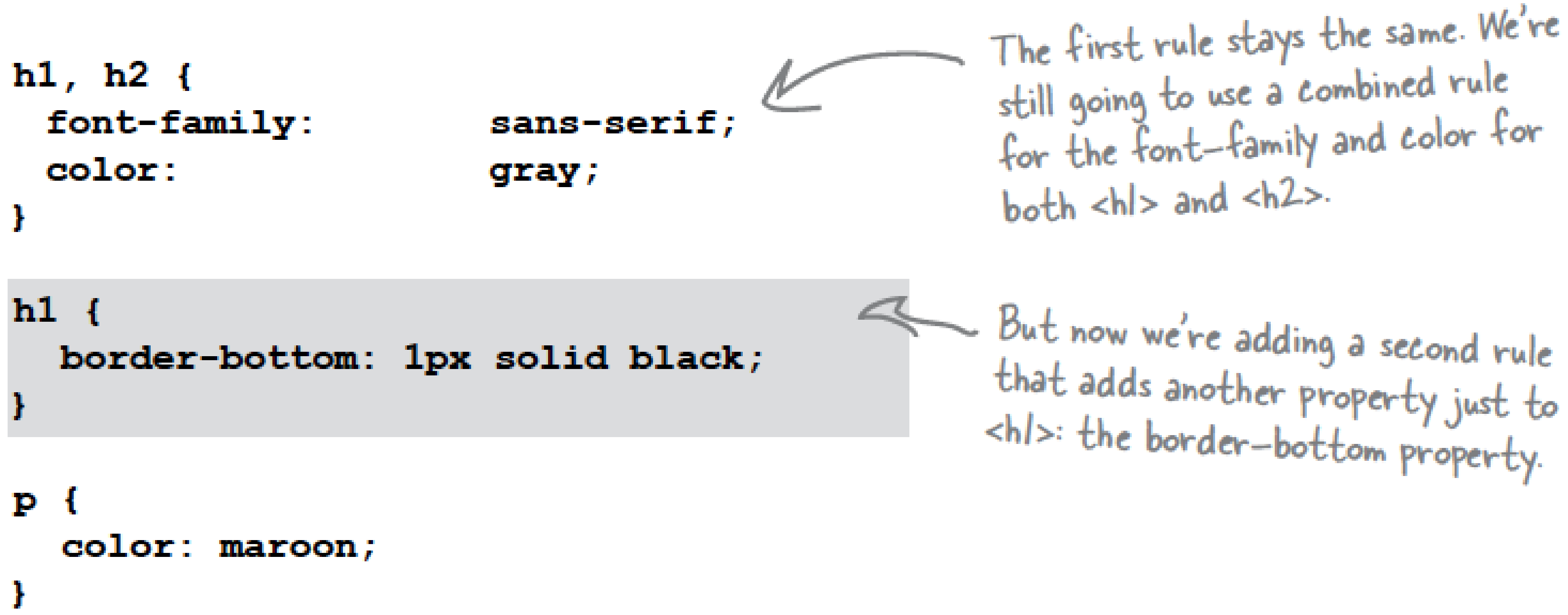
- Placing the above rule associated with h1 “selector”, will draw a line - 1 pixel wide - under the heading in our site (you did this in lab1)



Mobile Applications

1. Apps

Combining Selectors (2)



```
h1, h2 {
  font-family:        sans-serif;
  color:              gray;
}

h1 {
   border-bottom: 1px solid black;
}

p {
   color: maroon;
}
```

The first rule stays the same. We're still going to use a combined rule for the font-family and color for both <h1> and <h2>.

But now we're adding a second rule that adds another property just to <h1>: the border-bottom property.

- Both h1 and h2 share the font-family and color attributes, however only h1 is underlined

Combining Rules

- Rules can be combined. The following two sets of style rules would produce identical results

- Rules can be listed separately:

```
p {color: black;}
p {background-color: teal;}
p {padding: 1em;}
p {margin: 1em;}
p {font-family: helvetica, sans-serif;}
p {text-align: justify;}
```

- Or, rules can be grouped. Property:Value pairs need to be separated by a semicolon.

```
p
{
  color: black;
  background-color: teal;
  padding: 1em;
  margin: 1em;
  font-family: helvetica, sans-serif;
  text-align: justify;
}
```

Combining Selectors

- Selectors can be combined into comma-separated groups.

- We combine the selectors so that a single declaration applies to multiple selectors.

```
h1 { color: maroon; }
h2 { color: maroon; }
h3 { color: maroon; }
h4 { color: maroon; }
h5 { color: maroon; }
h6 { color: maroon; }
```

```
h1, h2, h3, h4, h5, h6 { color: maroon; }
```

Problem

```
<body>
  <p>
    Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci
    nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.
  </p>
  <p>
    Fusce velit. Integer sapien enim, rhoncus vitae, cursus non,
    commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum
    hendrerit turpis.
  </p>
  <p>
    Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras
    sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
    velit at orci.
  </p>
  <p>
    Lorem ipsum dolor sit amet,<span>consectetuer adipiscing elit</span>. Cras
    sollicitudin, orci
    nec acilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.
  </p>
  <p>
    Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras
    sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
    velit at orci.
  </p>
</body>
```

- How to style these paragraphs differently?
- Just using p as the selector will set the style for them all.

```
p
{
  color: black;
  background-color: teal;  ;
}
```

Agenda

- CSS
  - Rules, Selectors, Declarations, Properties, Values
  - CSS Binding
  - Combining Rules & Selectors
  - Classes and class based styling
  - Selector rules and the “Cascade” in CSS

Class of elements

- Using a class of elements for styling

- You can then write a css rule to style any elements that belong to the class.

- Elements can be in more than one class

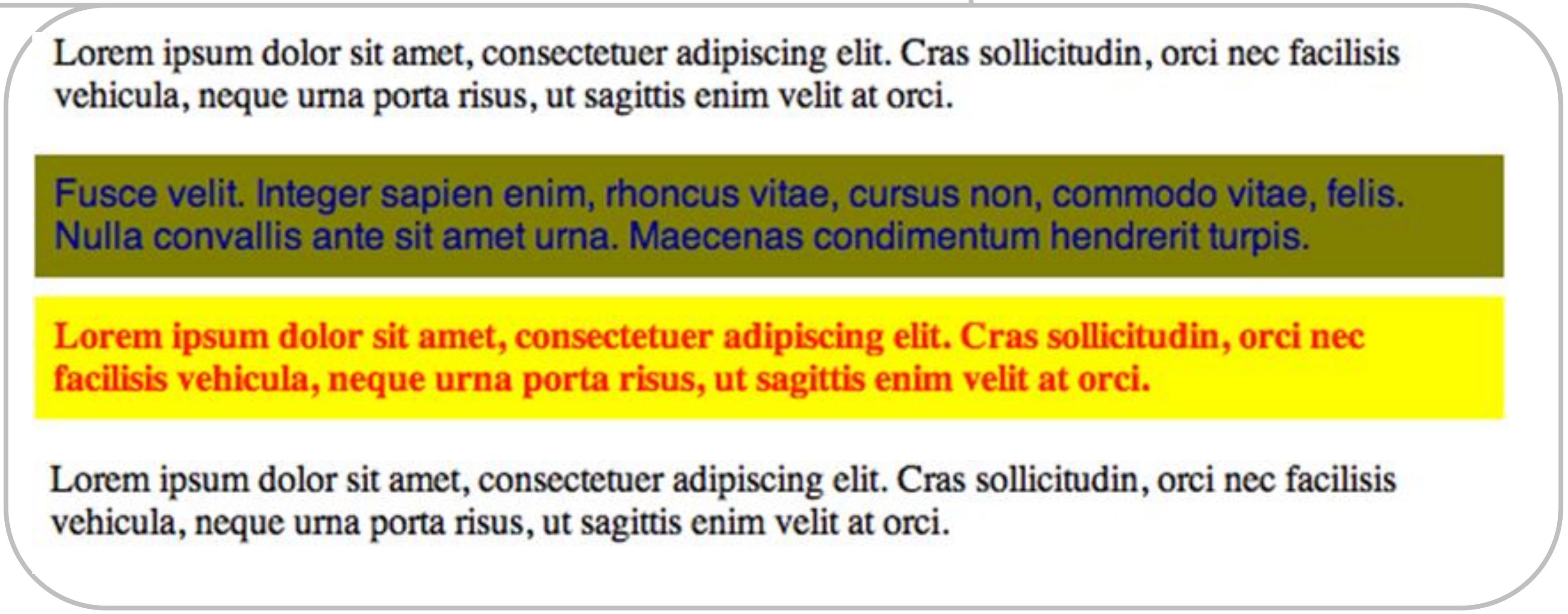
Using **class** to identify elements

```
 <body>
   <p>
     Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci
     nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.
   </p>
   <p class="withstyle">
     Fusce velit. Integer sapien enim, rhoncus vitae, cursus non,
     commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum
     hendrerit turpis.
   </p>
   <p class="warning">
     Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras
     sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
     velit at orci.
   </p>
   <p>
     Lorem ipsum dolor sit amet,<span class="warning">consectetuer adipiscing
elit</span>.
     Cras sollicitudin, orci nec acilisis vehicula, neque urna porta risus, ut sagittis
     enim velit at orci.
   </p>
</body>
```

- To indicate that an element is a member of a class we use the class attribute.

- While the name of an element specifies its *type*, the class attribute lets you assign to it one or more *subtypes*.

```
 <body>
   <p>
     Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci
     nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.
   </p>
   <p class="withstyle">
     Fusce velit. Integer sapien enim, rhoncus vitae, cursus non,
     commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas
condimentum
     hendrerit turpis.
   </p>
   <p class="warning">
     Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras
     sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim
     velit at orci.
   </p>
   <p>
     Lorem ipsum dolor sit amet,<span class="warning">consectetuer adipiscing
elit</span>.
     Cras sollicitudin, orci nec acilisis vehicula, neque urna porta risus, ut sagittis
     enim velit at orci.
   </p>
</body>
```



Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

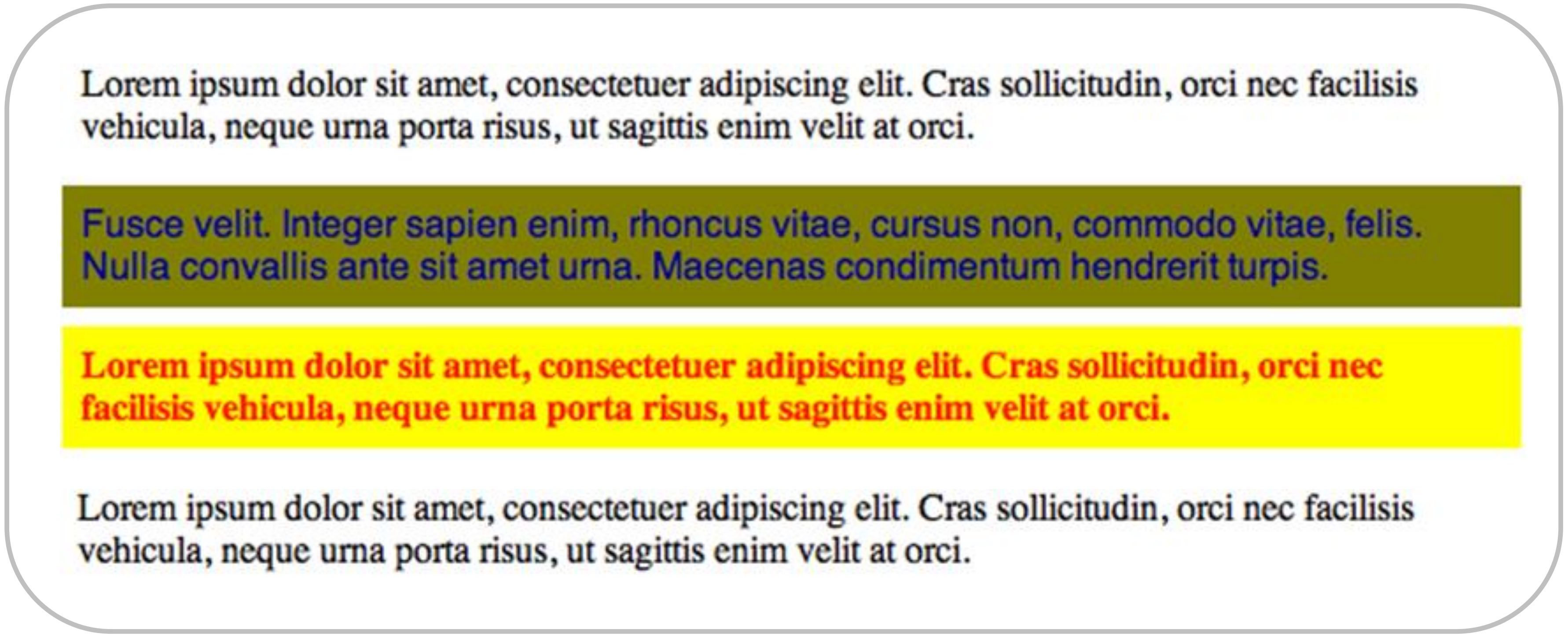
Fusce velit. Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum hendrerit turpis.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Using Classes in CSS

- Class names are referenced in CSS as
  - element.classname



Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Fusce velit. Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum hendrerit turpis.

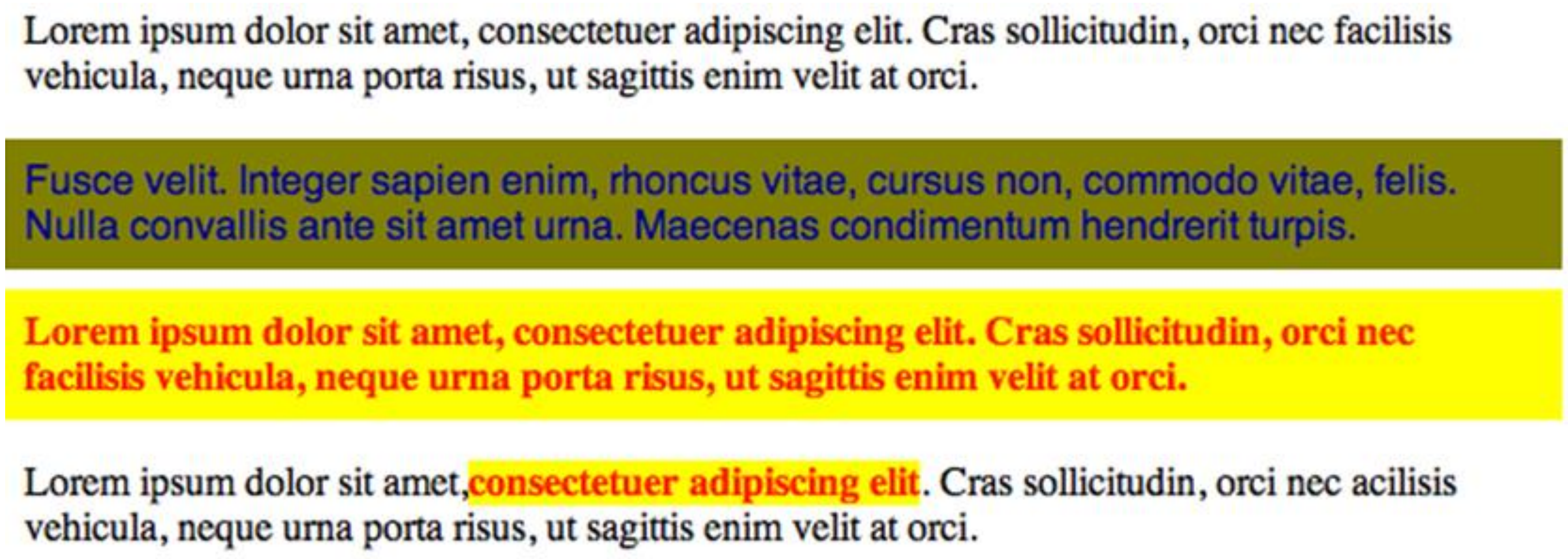
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

```
p
{
  background-color: white;
  color: black;
  font-family: times;
  margin: 0.5em;
  padding: 0.5em;
}

p.withstyle
{
  background-color: olive;
  color: navy;
  font-family: sans-serif;
  margin: 0.5em;
  padding: 0.5em;
}

p.warning
{
  background-color: yellow;
  color: red;
  font-weight: bold;
}
```

Classes Independent of Elements

- May not have an element name preceding the period:
    - .classname
- Selector now matches *any* element of the given class



Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Fusce velit. Integer sapien enim, rhoncus vitae, cursus non, commodo vitae, felis. Nulla convallis ante sit amet urna. Maecenas condimentum hendrerit turpis.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras sollicitudin, orci nec facilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

Lorem ipsum dolor sit amet,**consectetuer adipiscing elit**. Cras sollicitudin, orci nec acilisis vehicula, neque urna porta risus, ut sagittis enim velit at orci.

```
p
{
  background-color: white;
  color: black;
  font-family: times;
  margin: 0.5em;
  padding: 0.5em;
}

.withstyle
{
  background-color: olive;
  color: navy;
  font-family: sans-serif;
  margin: 0.5em;
  padding: 0.5em;
}

.warning
{
  background-color: yellow;
  color: red;
  font-weight: bold;
}
```

Agenda

- CSS
  - Rules, Selectors, Declarations, Properties, Values
  - CSS Binding
  - Combining Rules & Selectors
  - Html recap: attributes & Span elements
  - Classes and class based styling
  - Selector rules and the “Cascade” in CSS

Rules, Classes, Elements - which ones get selected?

(1) *Explicit Match:* Do any selectors select your element?

- Examine CSS rules for *explicit match* for element.

(2) *Inheritance Match*: What if no rules match the element:

- Rely on *inheritance*.
- Look at the element’s parents, and parents’ parents, and so on, until you find the property defined.

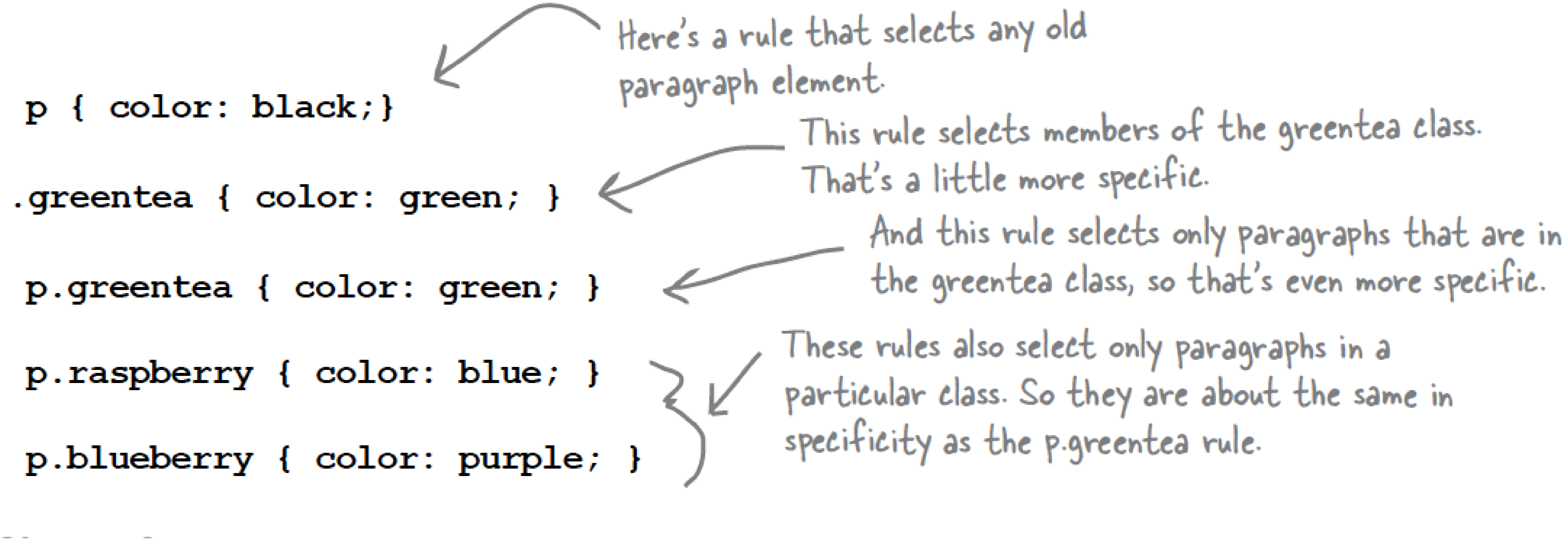
(3) *Default Match:* still no explicit or inherited match

- use the *default* value defined by the browser

(4) *Most Specific Match*: What if more that one match (of different rules)?

- select rule that is the *most specific*

Examples



```
p { color: black;}

.greentea { color: green; }

p.greentea { color: green; }

p.raspberry { color: blue; }

p.blueberry { color: purple; }
```

Here's a rule that selects any old paragraph element.

This rule selects members of the greentea class. That's a little more specific.

And this rule selects only paragraphs that are in the greentea class, so that's even more specific.

These rules also select only paragraphs in a particular class. So they are about the same in specificity as the p.greentea rule.

Specific & Ordering Match

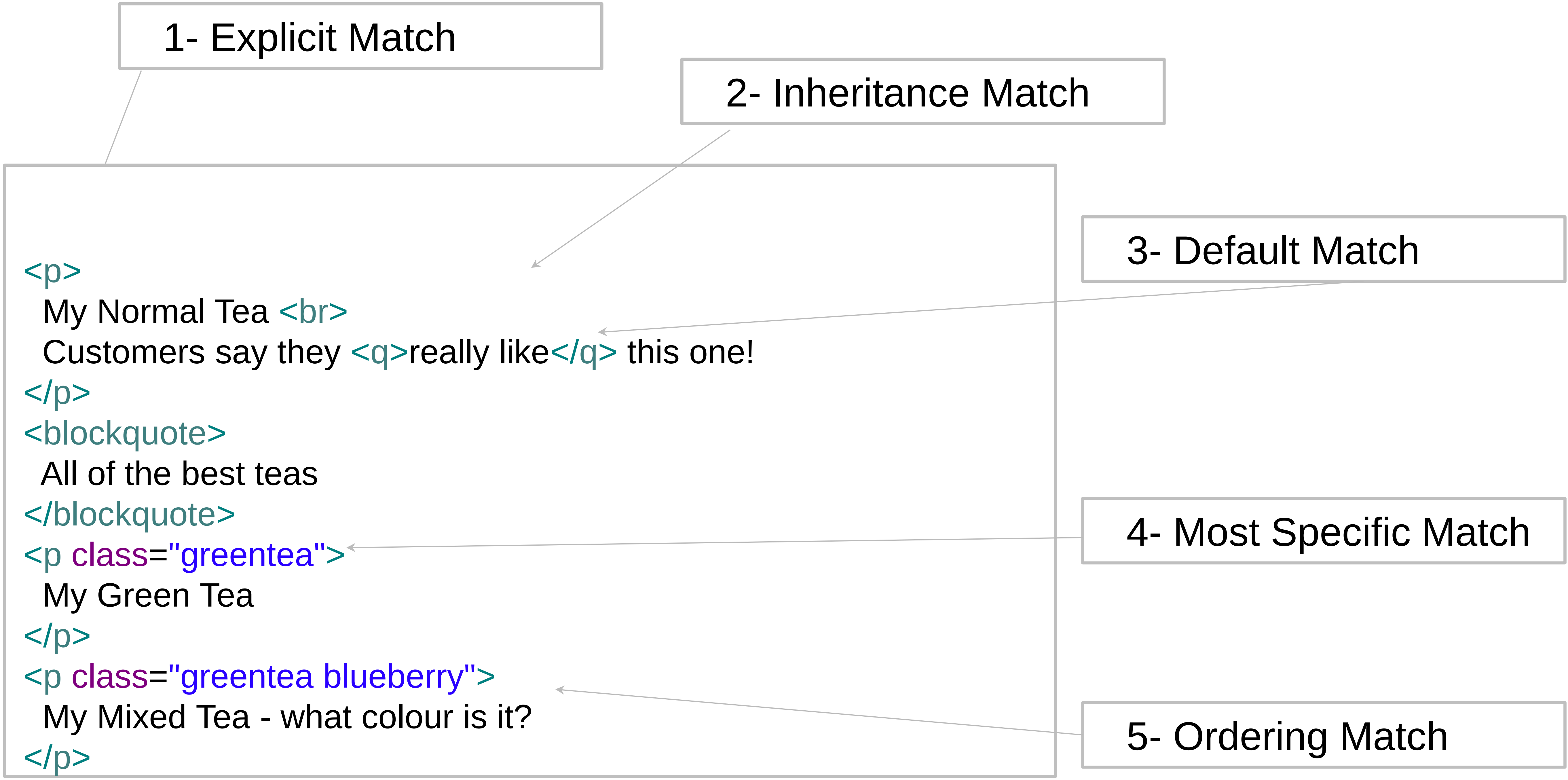
- If we had an element that belonged only to the greentea class
  - there would be an obvious winner:
  - the p.greentea selector is the most specific, so the text would be green.
- But you have an element that belongs to all three classes: greentea, raspberry, and blueberry.
  - So, p.greentea, p.raspberry, and p.blueberry all select the element, and are of equal specificity. What do you do now?
  - You choose the one that is listed last in the CSS file.

(5): *Ordering Match:* - If you can’t resolve a conflict because two selectors are equally specific

- use the ordering of the rules in your style sheet file. That is, you use the rule listed last in the CSS file (nearest the bottom). And in this case, that would be the p.blueberry rule.

Example

```
p           { color: black;  }
.greentea   { color: green;  }
p.greentea  { color: green;  }
p.raspberry { color: blue;   }
p.blueberry { color: purple; }
```



1- Explicit Match

2- Inheritance Match

3- Default Match

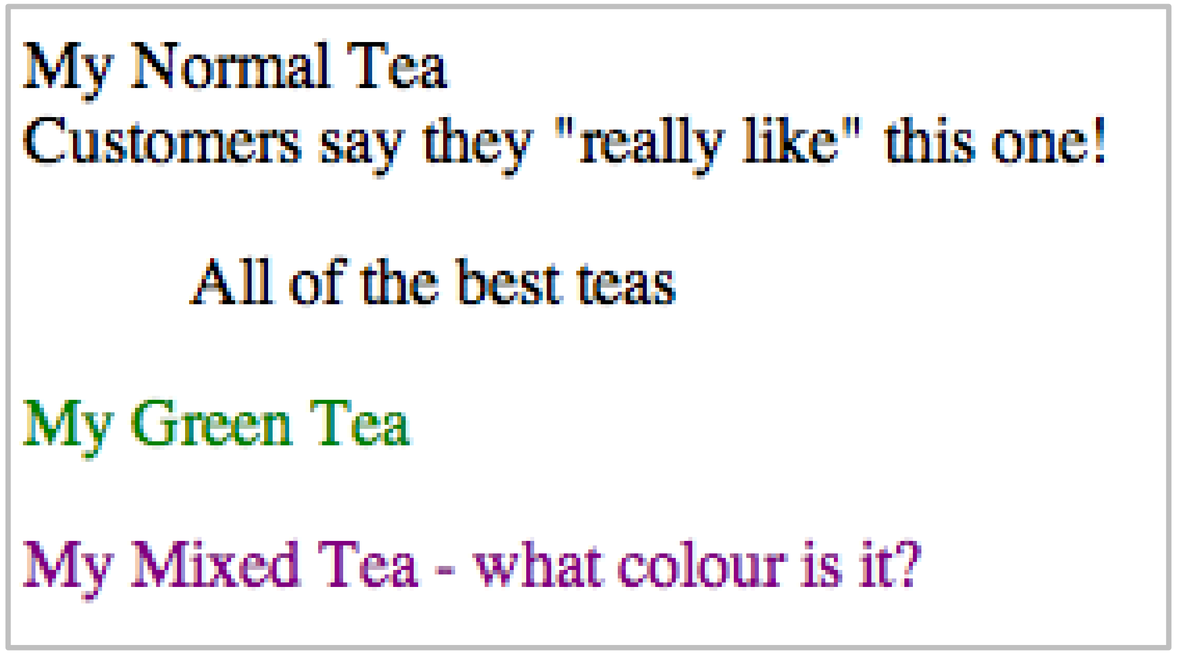
4- Most Specific Match

5- Ordering Match

```
<p>
  My Normal Tea <br>
  Customers say they <q>really like</q> this one!
</p>
<blockquote>
  All of the best teas
</blockquote>
<p class="greentea">
  My Green Tea
</p>
<p class="greentea blueberry">
  My Mixed Tea - what colour is it?
</p>
```

```
p           { color: black;  }
.greentea   { color: green;  }
p.greentea  { color: green;  }
p.raspberry { color: blue;   }
p.blueberry { color: purple; }
```

```
<p>
  My Normal Tea <br>
  Customers say they <q>really like</q> this one!
</p>
<blockquote>
  All of the best teas
</blockquote>
<p class="greentea">
  My Green Tea
</p>
<p class="greentea blueberry">
  My Mixed Tea - what colour is it?
</p>
```



My Normal Tea
Customers say they "really like" this one!

> All of the best teas

My Green Tea

My Mixed Tea - what colour is it?

Learning Outcomes

- On completion of this class you should:
  - Understand the motivations for using CSS in web site development
  - Be able to compose simple CSS rules and incorporate them into a separate CSS file for a site.