CSS

Author: TrungDVQ (Fsoft-Academy)

Fsoft-Academy - 2020

Lesson Objectives

01 Introduction

- ✓ What is CSS?
- ✓ History

02 CSS Basic Concepts

- ✓ Syntax Selector
- ✓ Apply CSS to HTML
- ✓ Cascade and Inheritance

03 CSS Essential

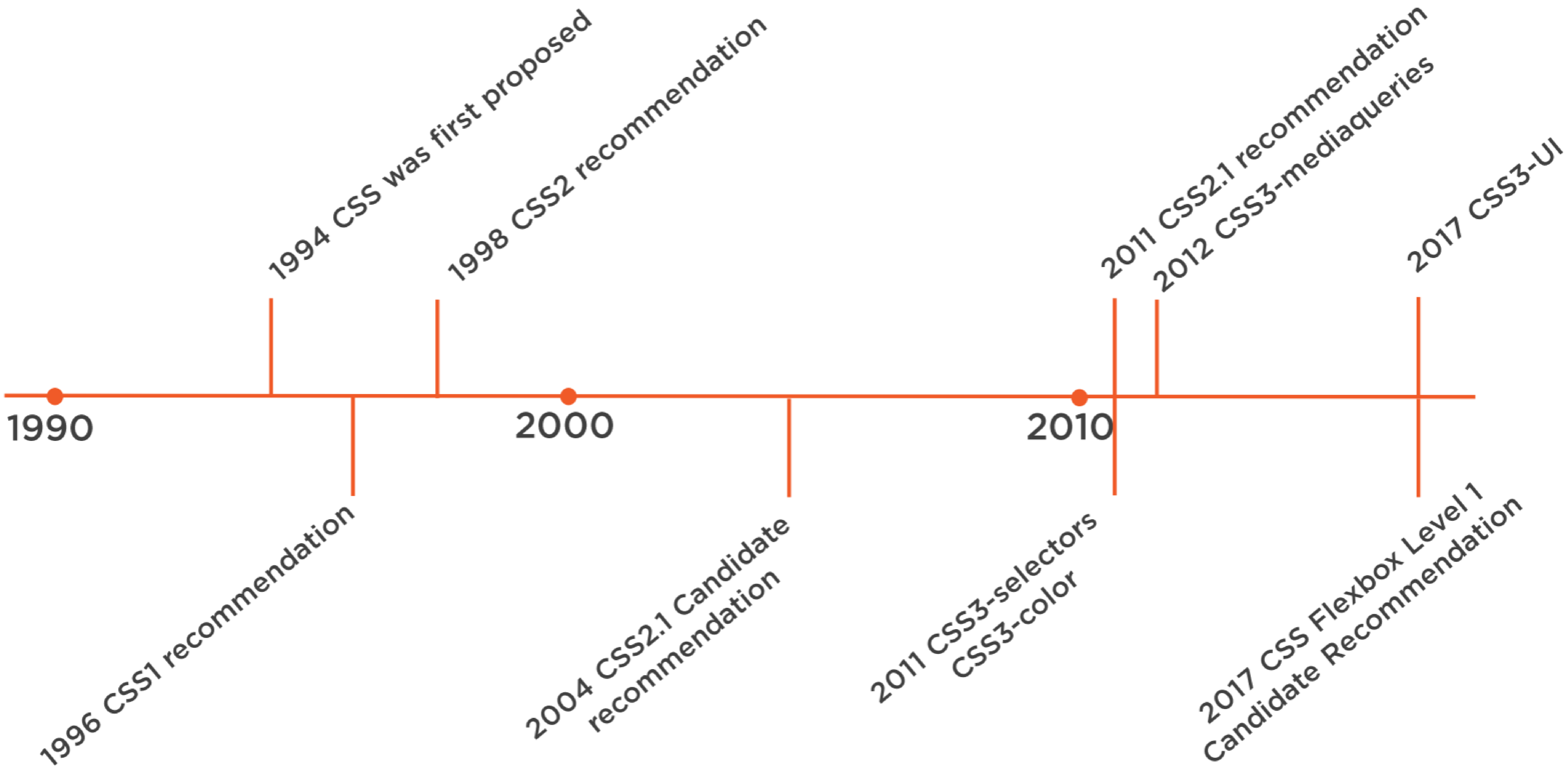
- ✓ Width, height, overflow
- ✓ Borders, background
- ✓ Text style
- ✓ Margins, paddings and float
- ✓ Units

1

Introducing CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language.

History of CSS



1990

1994 CSS was first proposed

1996 CSS1 recommendation

1998 CSS2 recommendation

2000

2004 CSS2.1 Candidate recommendation

2010

2011 CSS2.1 recommendation

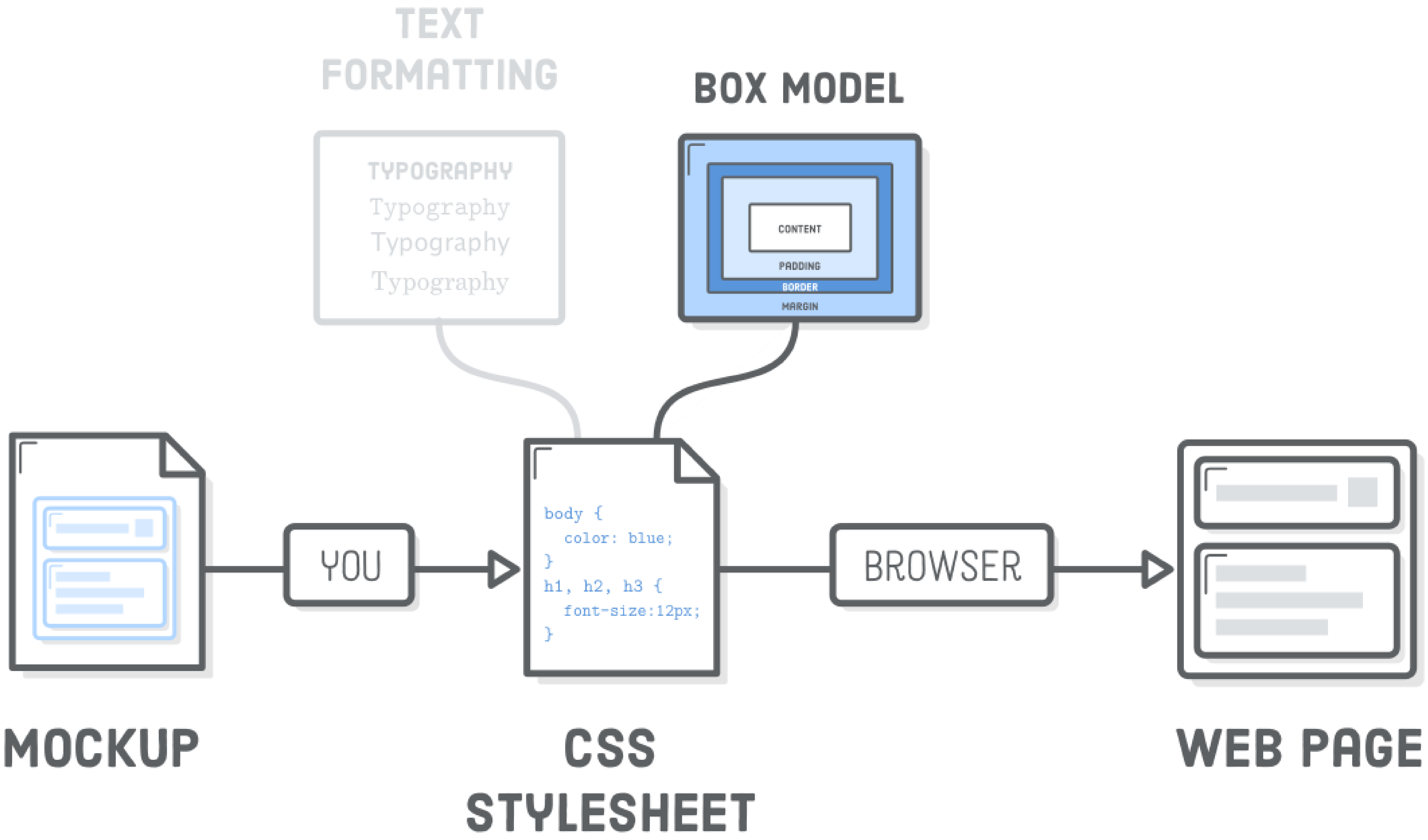
2011 CSS3-selectors CSS3-color

2012 CSS3-mediaqueries

2017 CSS3-UI

2017 CSS Flexbox Level 1 Candidate Recommendation

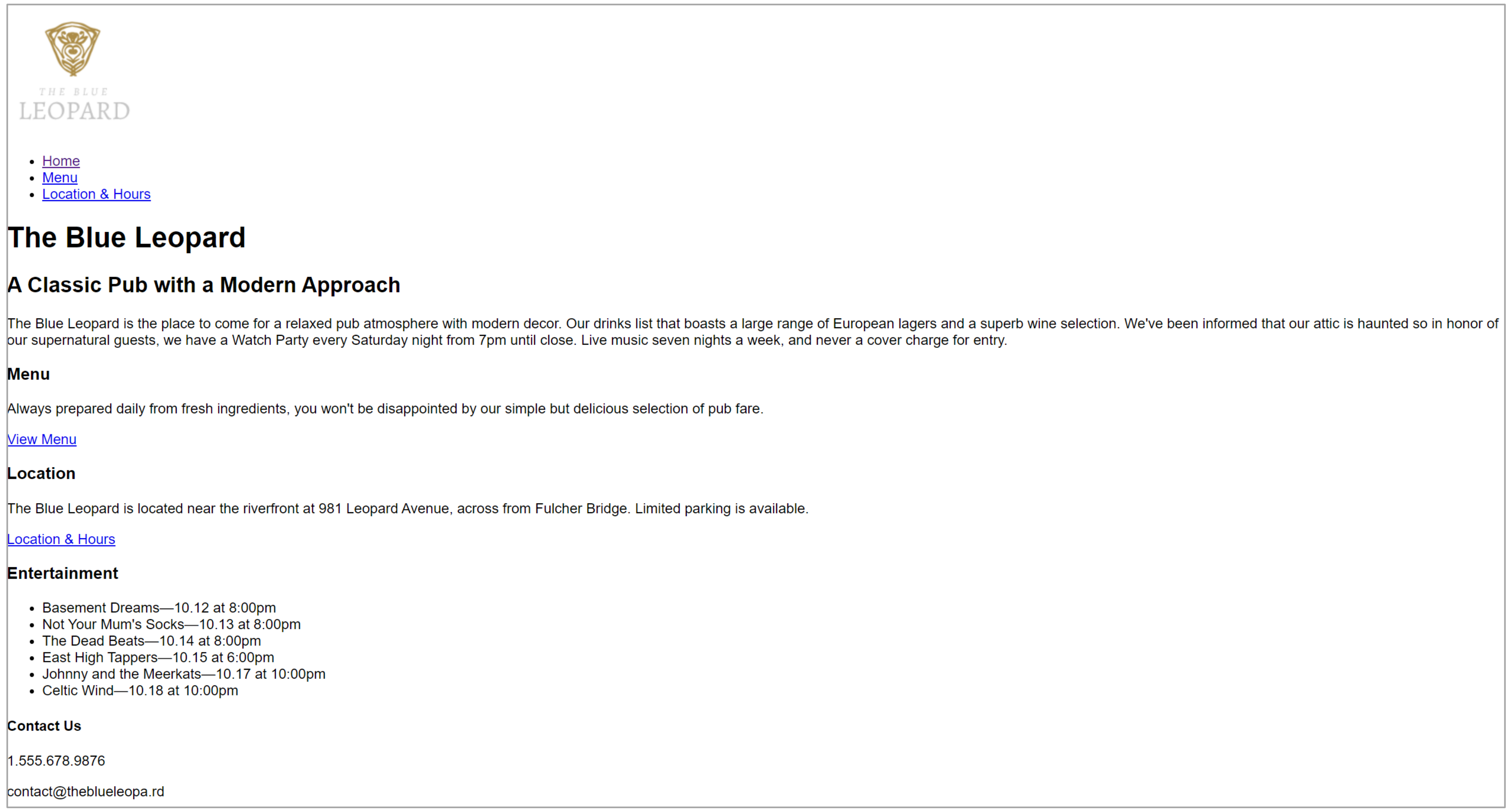
WHY USE CSS?



```
body {
  color: blue;
}
h1, h2, h3 {
  font-size:12px;
}
```

CSS IS AWESOME

HTML



THE BLUE
LEOPARD

- Home
- Menu
- Location & Hours

The Blue Leopard

A Classic Pub with a Modern Approach

The Blue Leopard is the place to come for a relaxed pub atmosphere with modern decor. Our drinks list that boasts a large range of European lagers and a superb wine selection. We've been informed that our attic is haunted so in honor of our supernatural guests, we have a Watch Party every Saturday night from 7pm until close. Live music seven nights a week, and never a cover charge for entry.

Menu

Always prepared daily from fresh ingredients, you won't be disappointed by our simple but delicious selection of pub fare.

View Menu

Location

The Blue Leopard is located near the riverfront at 981 Leopard Avenue, across from Fulcher Bridge. Limited parking is available.

Location & Hours

Entertainment

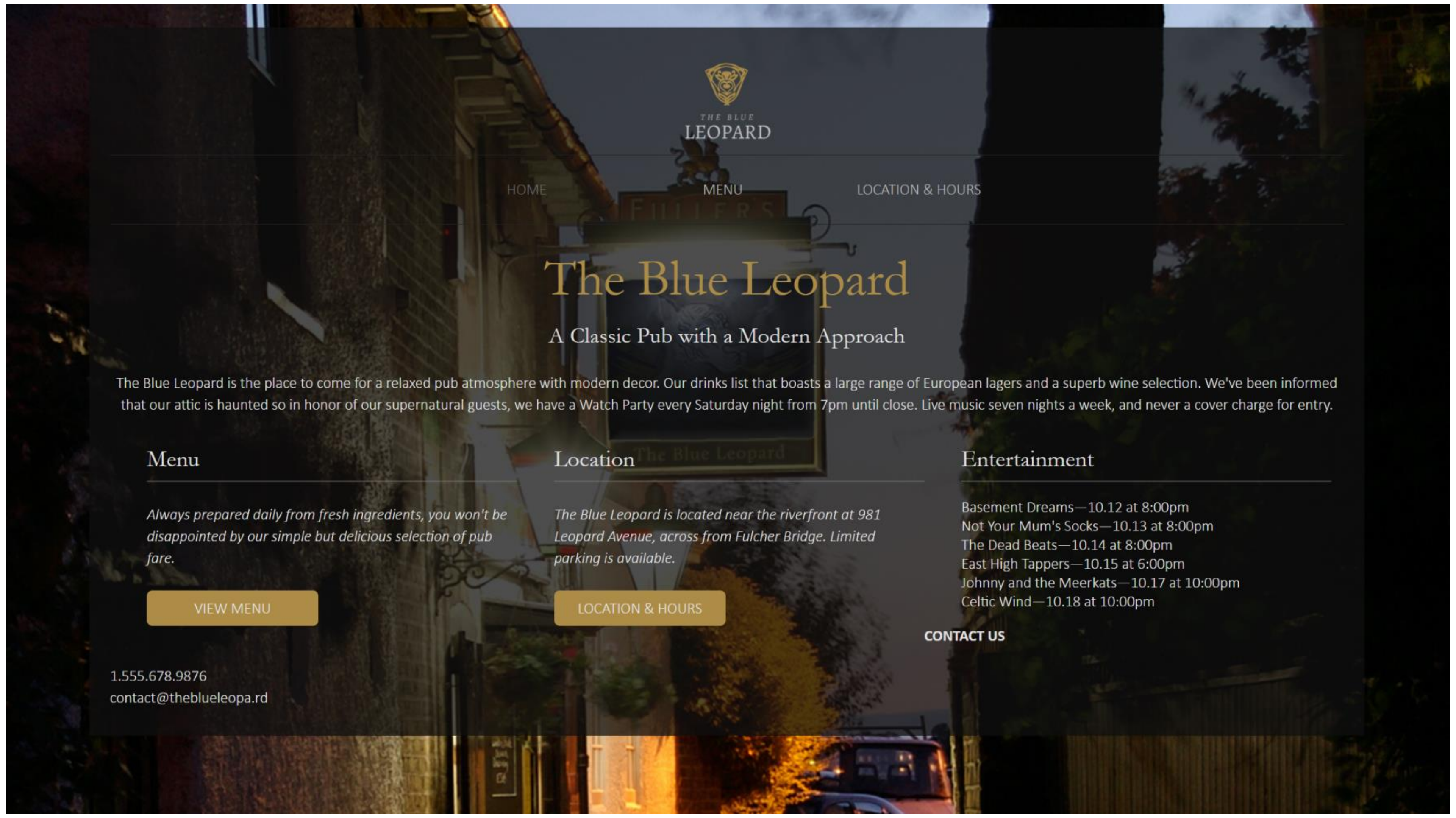
- Basement Dreams—10.12 at 8:00pm
- Not Your Mum's Socks—10.13 at 8:00pm
- The Dead Beats—10.14 at 8:00pm
- East High Tappers—10.15 at 6:00pm
- Johnny and the Meerkats—10.17 at 10:00pm
- Celtic Wind—10.18 at 10:00pm

Contact Us

1.555.678.9876

contact@theblueleopa.rd

HTML + CSS



THE BLUE
LEOPARD

HOME MENU LOCATION & HOURS

The Blue Leopard

A Classic Pub with a Modern Approach

The Blue Leopard is the place to come for a relaxed pub atmosphere with modern decor. Our drinks list that boasts a large range of European lagers and a superb wine selection. We've been informed that our attic is haunted so in honor of our supernatural guests, we have a Watch Party every Saturday night from 7pm until close. Live music seven nights a week, and never a cover charge for entry.

Menu

Always prepared daily from fresh ingredients, you won't be disappointed by our simple but delicious selection of pub fare.

VIEW MENU

Location

The Blue Leopard is located near the riverfront at 981 Leopard Avenue, across from Fulcher Bridge. Limited parking is available.

LOCATION & HOURS

Entertainment

Basement Dreams—10.12 at 8:00pm
Not Your Mum's Socks—10.13 at 8:00pm
The Dead Beats—10.14 at 8:00pm
East High Tappers—10.15 at 6:00pm
Johnny and the Meerkats—10.17 at 10:00pm
Celtic Wind—10.18 at 10:00pm

CONTACT US

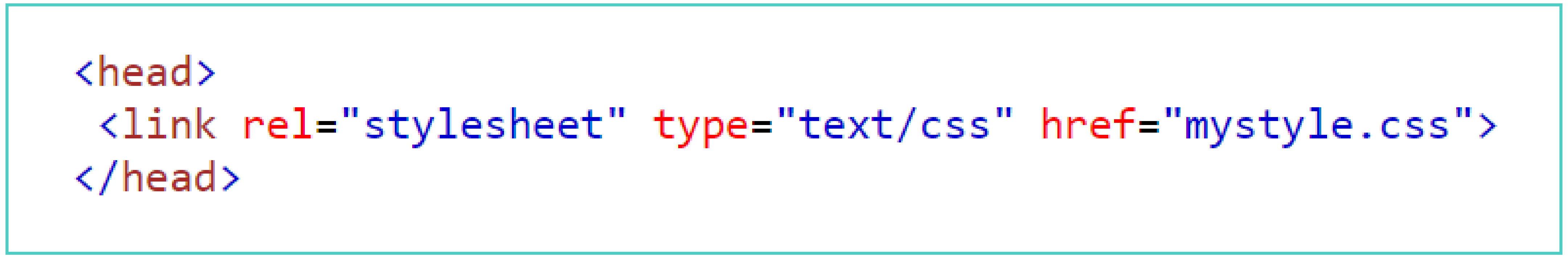
1.555.678.9876
contact@theblueleopa.rd

2

CSS Basic Concepts

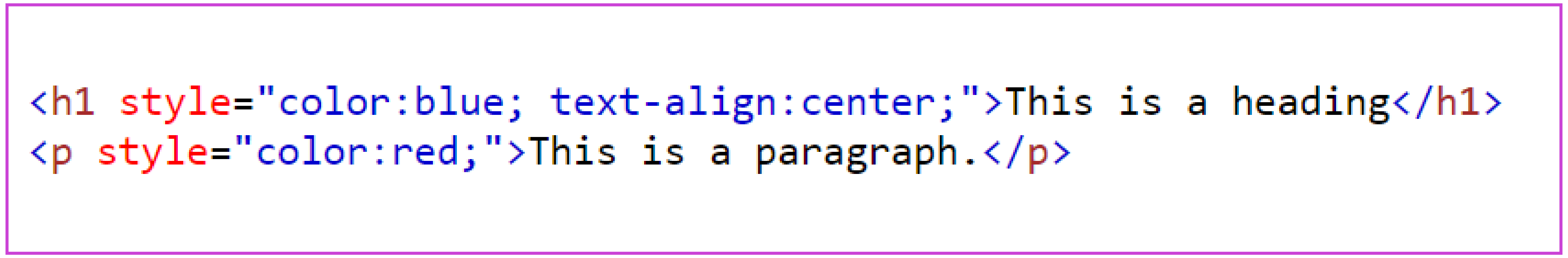
Apply CSS to HTML

There are three ways of inserting a style sheet:



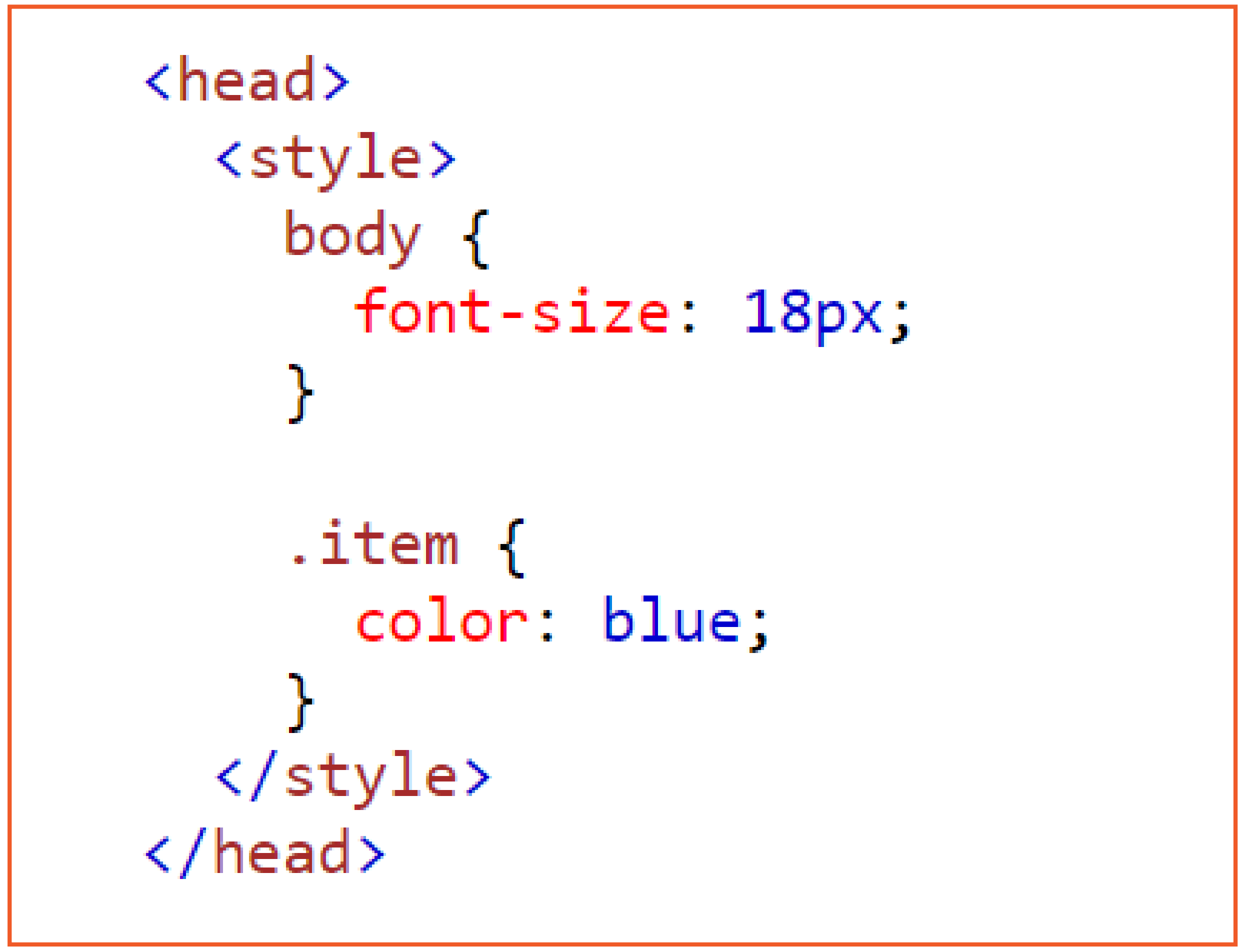
```
<head>
 <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

External CSS



```
<h1 style="color:blue; text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

Inline CSS



```
<head>
  <style>
    body {
      font-size: 18px;
    }

    .item {
      color: blue;
    }
  </style>
</head>
```

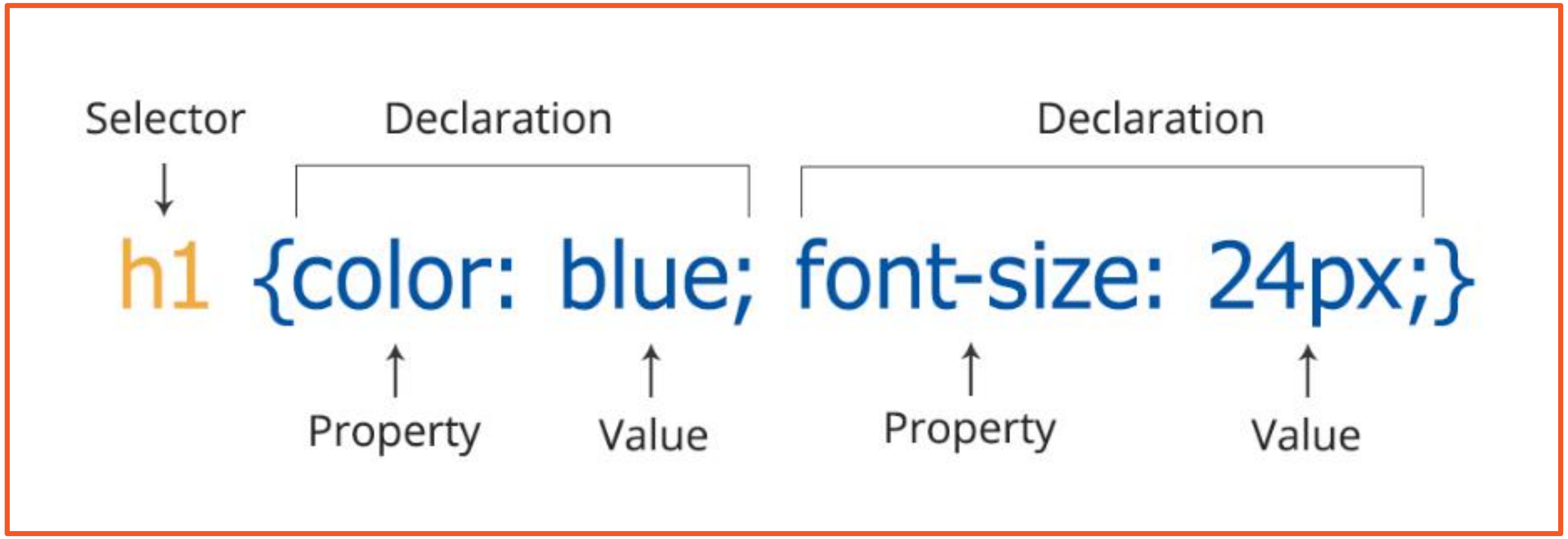
Internal CSS

CSS Cascade

HOW ABOUT ?

EXTERNAL CSS < INTERNAL CSS (CSS in <head>) < INLINE CSS

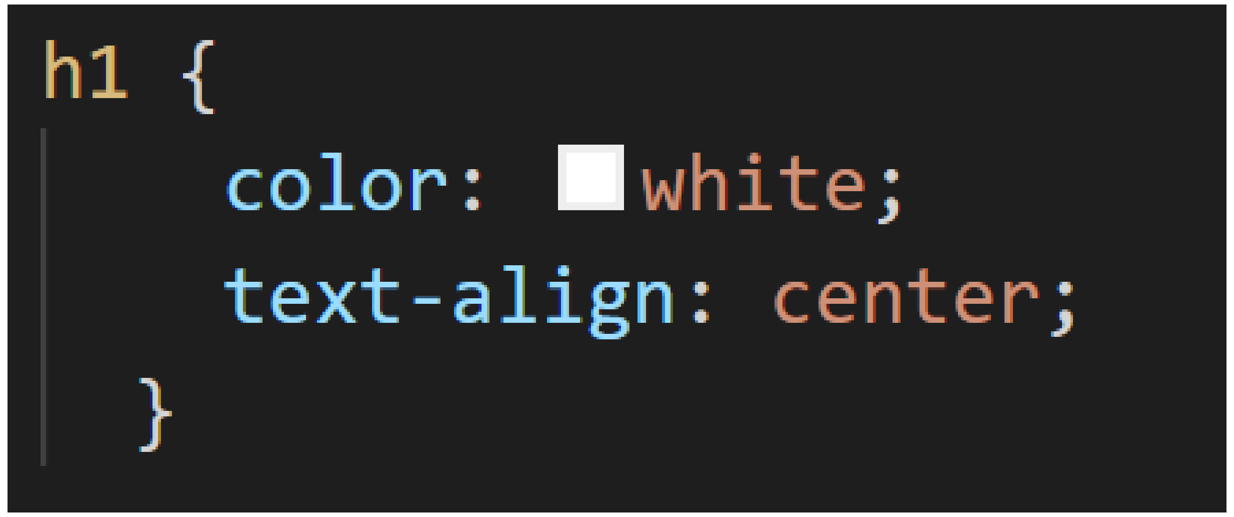
Syntax



Example on w3schools:

```
h1 {
  color: white;
  text-align: center;
}
```

Example on Visual Studio Code:



```
h1 {
    color: □white;
    text-align: center;
  }
```

CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

```
<div>
FPT Software Academy
</div>
```

```
div {
   color: red;
}
```

Element Selector

```
<div class="test-color">
    FPT Software Academy
</div>
```

```
 .test-color {
   color: red;
 }
```

Class Selector

```
<div id="test-color">
    FPT Software Academy
</div>
```

```
 #test-color {
   color: red;
 }
```

Id Selector

```
<h1>FPT Software Academy</h1>
<button>Submit</button>
<a href="">fsoft-academy.edu.vn/</a>
```

```
h1, button, a {
    color: red;
}
```

Group Selector

```
<div id="test-color">
    FPT Software Academy
</div>
```

```
Div#testcolor {
    color: red;
}
```

Nesting Selector

CSS Selectors

Universal selectors are used to select any element.

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

CSS Combinators

>C~S+S

Combinators

A combinator is something that explains the relationship between the selectors.

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

There are four different combinators in CSS3:

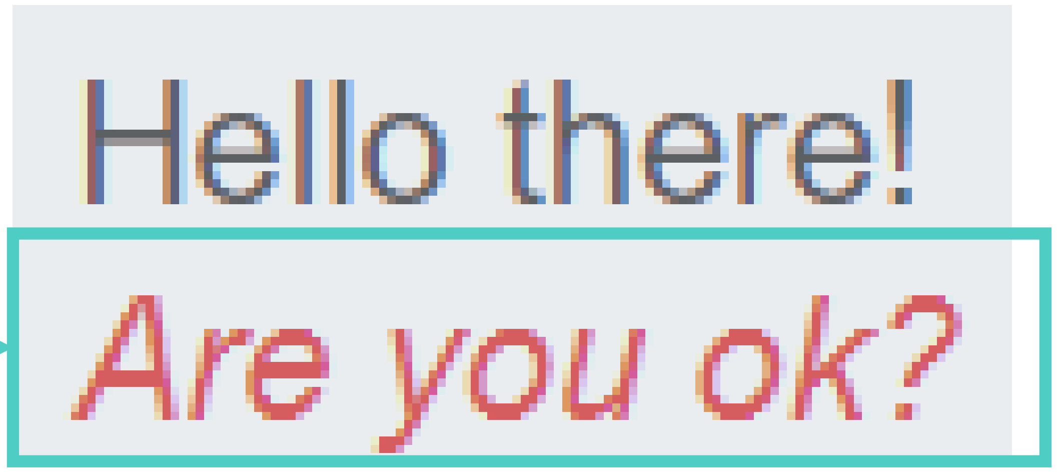
- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

Child selectors

A child selector is used to select an element that is a direct child of another element (parent). Child selectors will not select all descendants, only direct children.

```
<div class="abc">
    <div>
        <p>Hello there!</p>
    </div>
    <p>Are you ok?</p>
</div>
```

```
DIV.abc > P {
    font-
style: italic;
    color:#D55C5F;
}
```



Hello there!

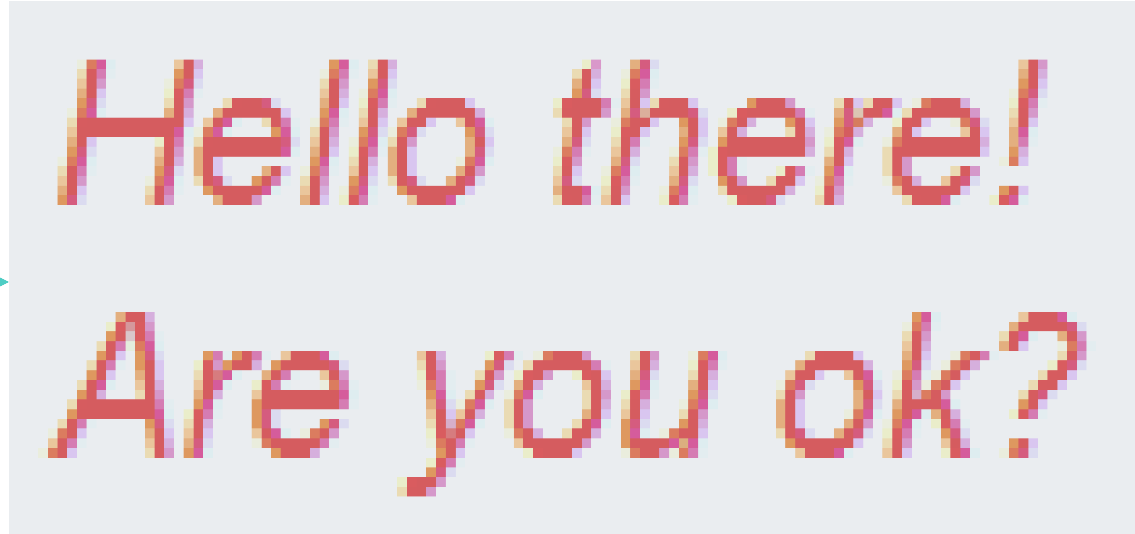
Are you ok?

Descendant selectors

Descendant selectors are used to select elements that are descendants (**not necessarily children**) of another element in the document tree.

```html
<div class="abc">
    <div>
        <p>Hello there!</p>
    </div>
    <p>Are you ok?</p>
</div>
```

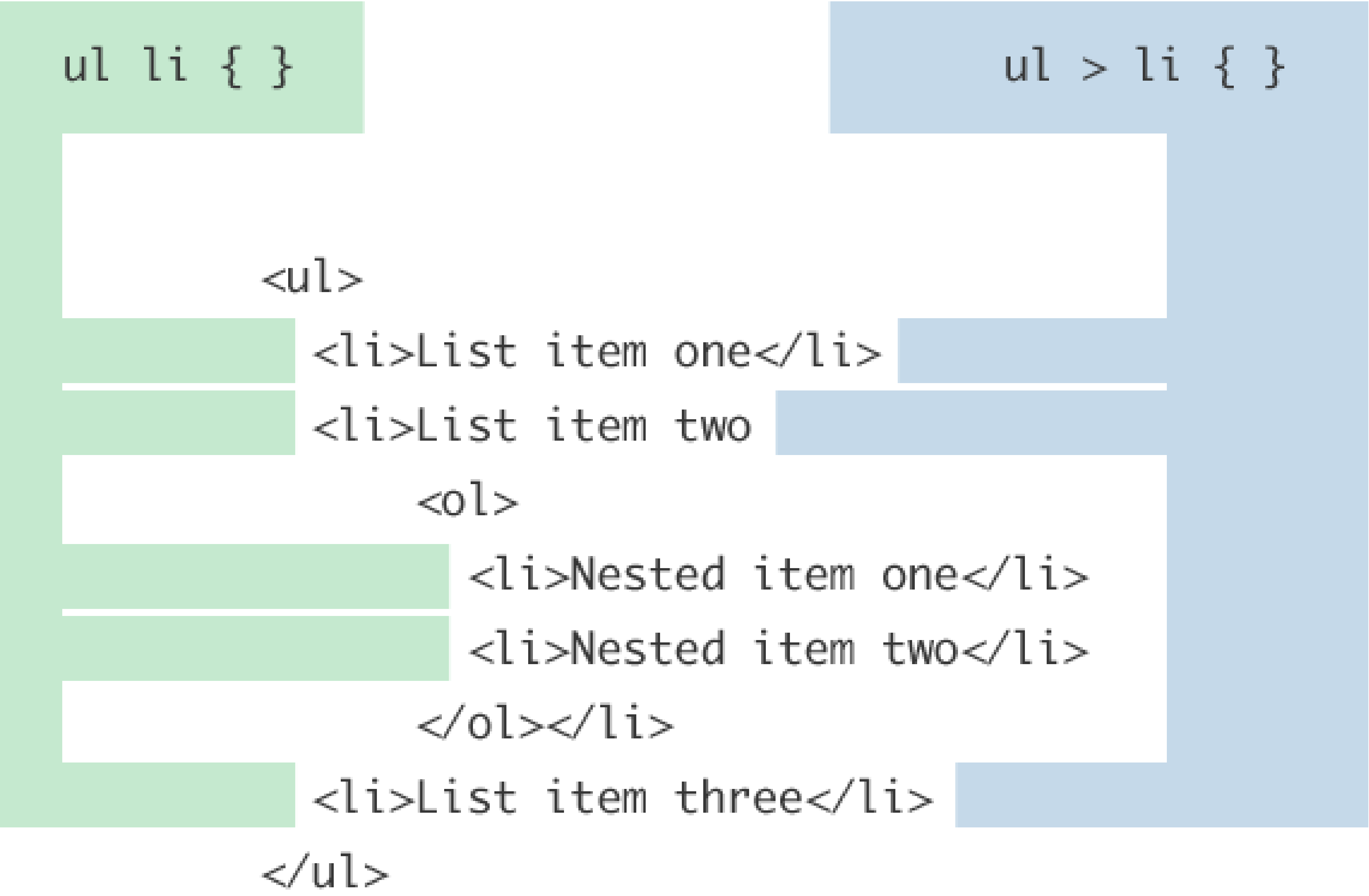
```css
DIV.abc P {
    font-style: italic;
    color:#D55C5F;
}
```



Hello there!

Are you ok?

CSS Child vs Descendant selectors



ul li { }

ul > li { }

```
<ul>
  <li>List item one</li>
  <li>List item two
    <ol>
      <li>Nested item one</li>
      <li>Nested item two</li>
    </ol></li>
  <li>List item three</li>
</ul>
```

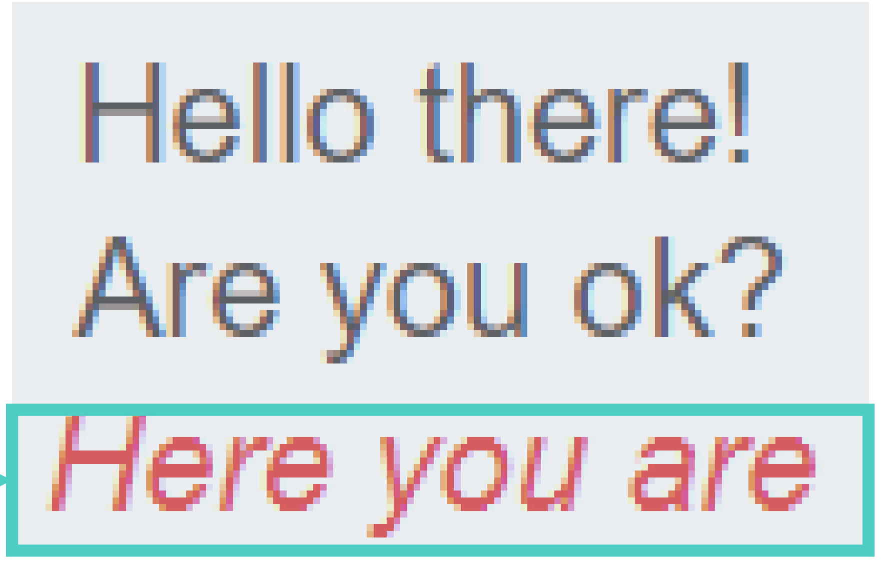
- ➢ My daughter is both my child and my descendant
- ➢ My granddaughter is not my child, but she is my descendant.

Adjacent sibling selectors

Adjacent sibling selectors will select the sibling immediately following an element.

```
<div class="abc">
    <div>
        <p>Hello there!</p>
    </div>
<p>Are you ok?</p>
</div>
<p>Here you are</p>
```

```
DIV.abc + P {
    font-style: italic;
    color:#D55C5F;
}
```



Hello there!

Are you ok?

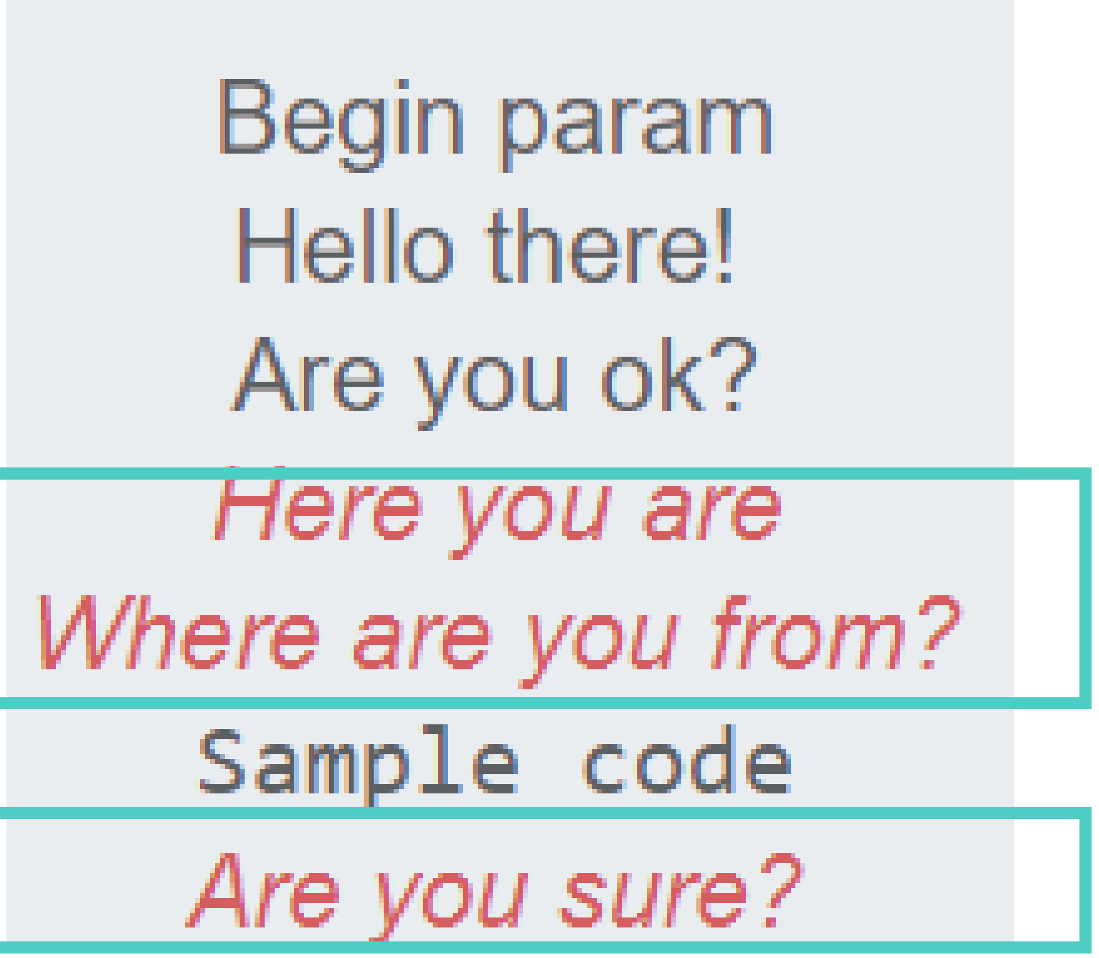
Here you are

General Sibling Selector

The general sibling selector selects all elements that are siblings of a specified element.

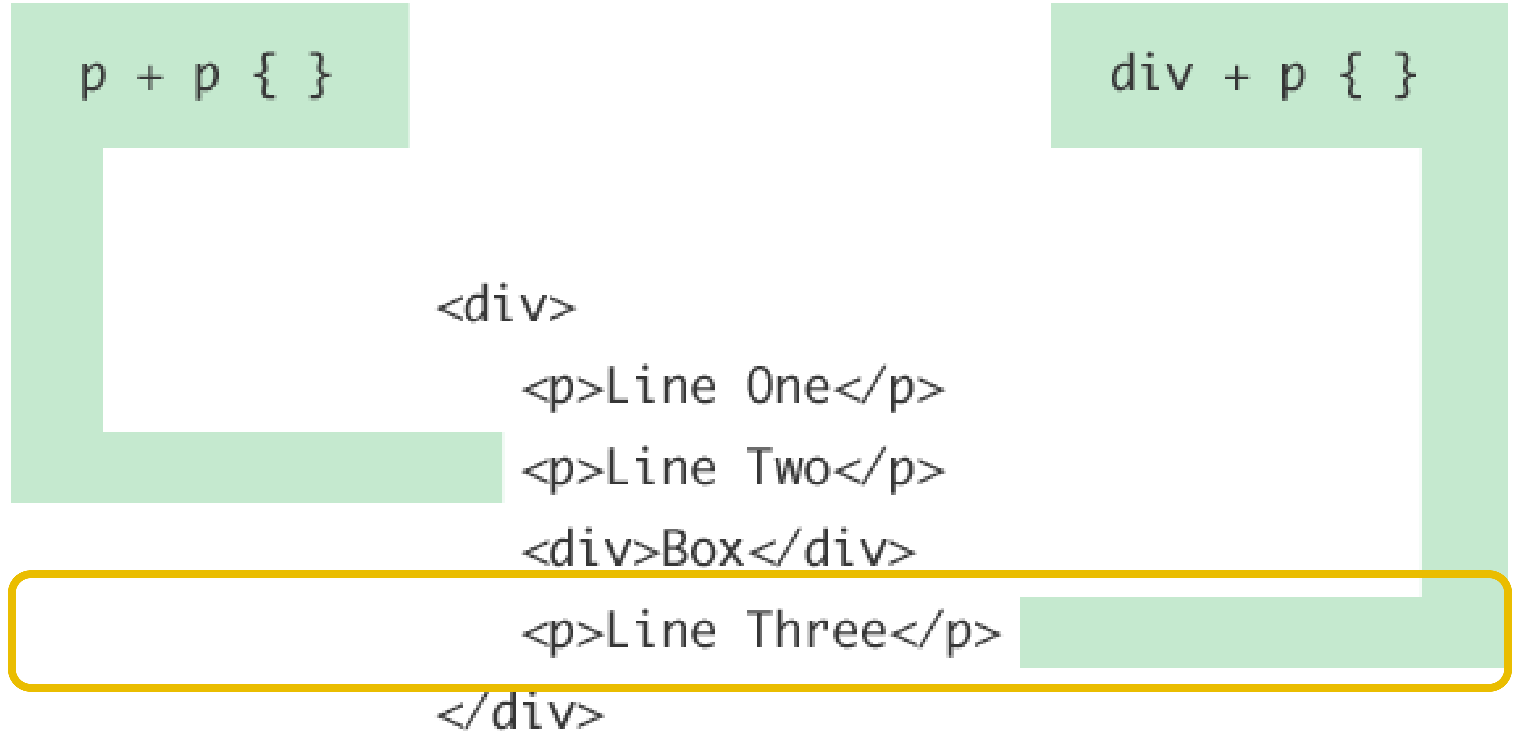
```
DIV.abc ~ P {
    font-style: italic;
    color:#D55C5F;
}
```

```
<p>Begin param</p>
<div class="abc">
    <div>
        <p>Hello there!</p>
    </div>
<p>Are you ok?</p>
</div>
<p>Here you are</p>
<p>Where are you from?</p>
<Code> Sample code </Code>
<p>Are  you sure?</p>
```



Begin param
Hello there!
Are you ok?
Here you are
Where are you from?
Sample code
Are you sure?

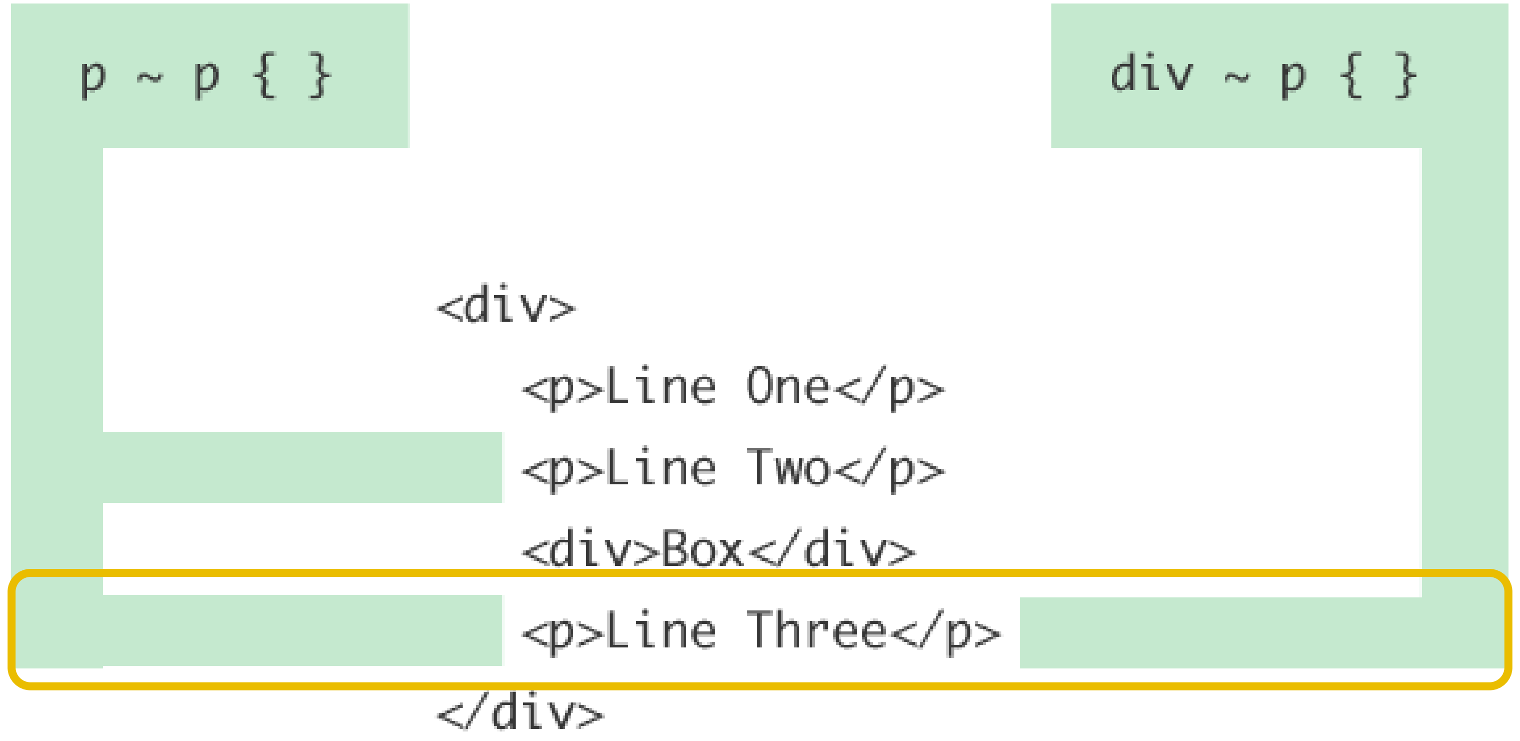
Adjacent sibling vs General Sibling selectors



p + p { }

div + p { }

```
<div>
   <p>Line One</p>
   <p>Line Two</p>
   <div>Box</div>
   <p>Line Three</p>
</div>
```

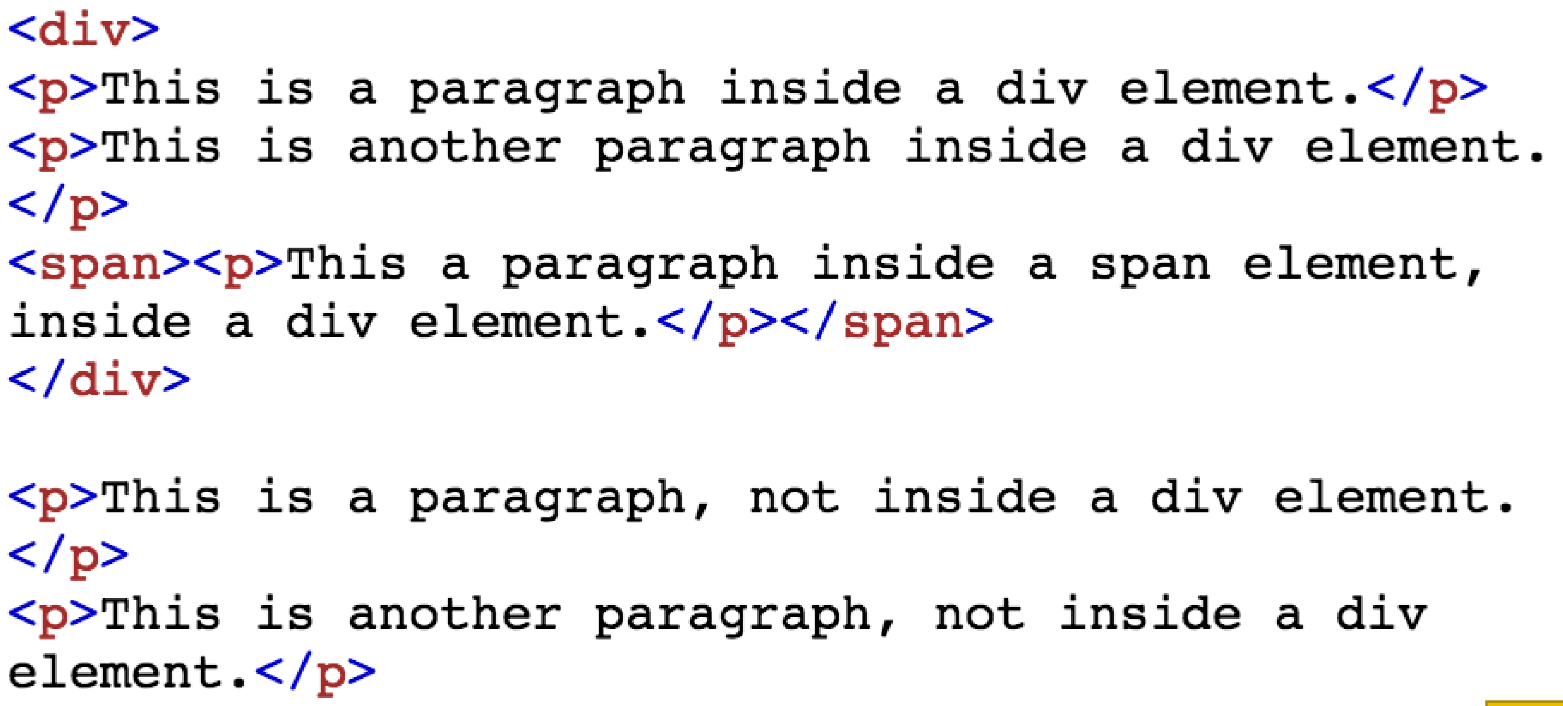


p ~ p { }

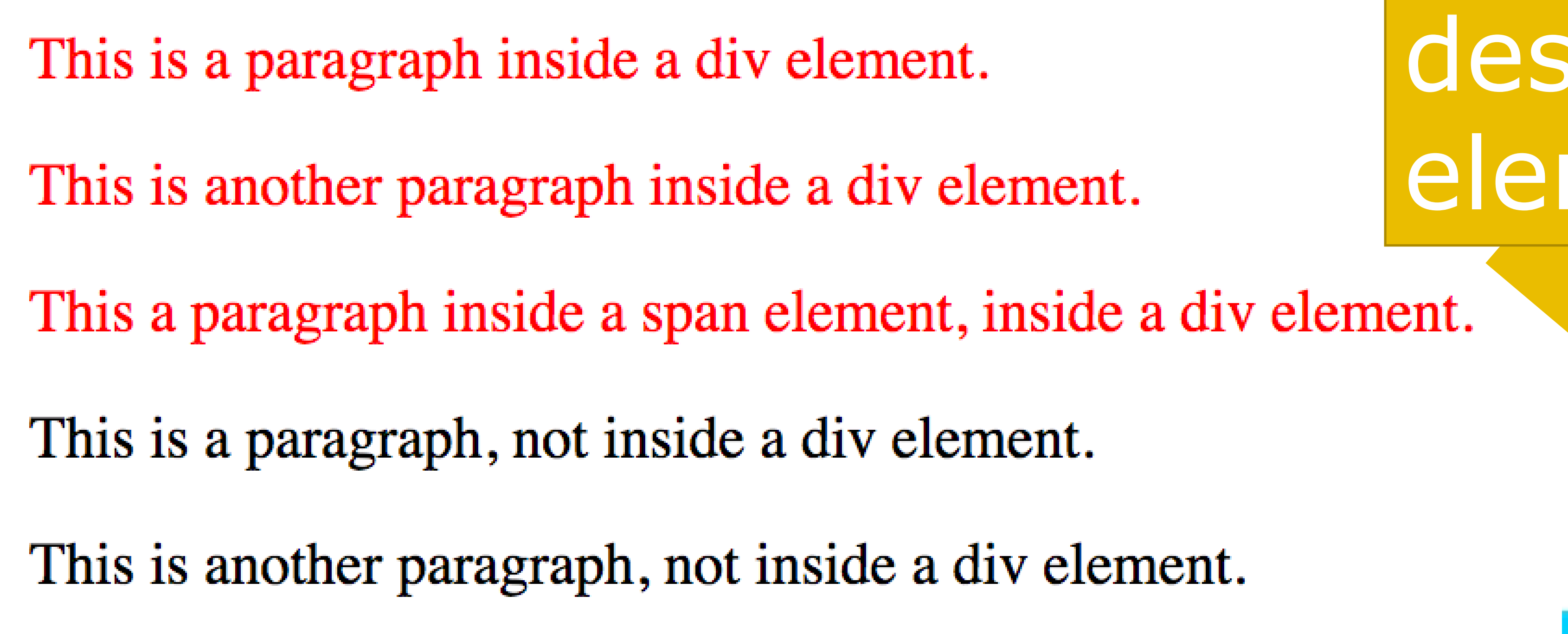
div ~ p { }

```
<div>
   <p>Line One</p>
   <p>Line Two</p>
   <div>Box</div>
   <p>Line Three</p>
</div>
```

Mini Excercise



```
<div>
<p>This is a paragraph inside a div element.</p>
<p>This is another paragraph inside a div element.
</p>
<span><p>This a paragraph inside a span element,
inside a div element.</p></span>
</div>

<p>This is a paragraph, not inside a div element.
</p>
<p>This is another paragraph, not inside a div
element.</p>
```



This is a paragraph inside a div element.

This is another paragraph inside a div element.

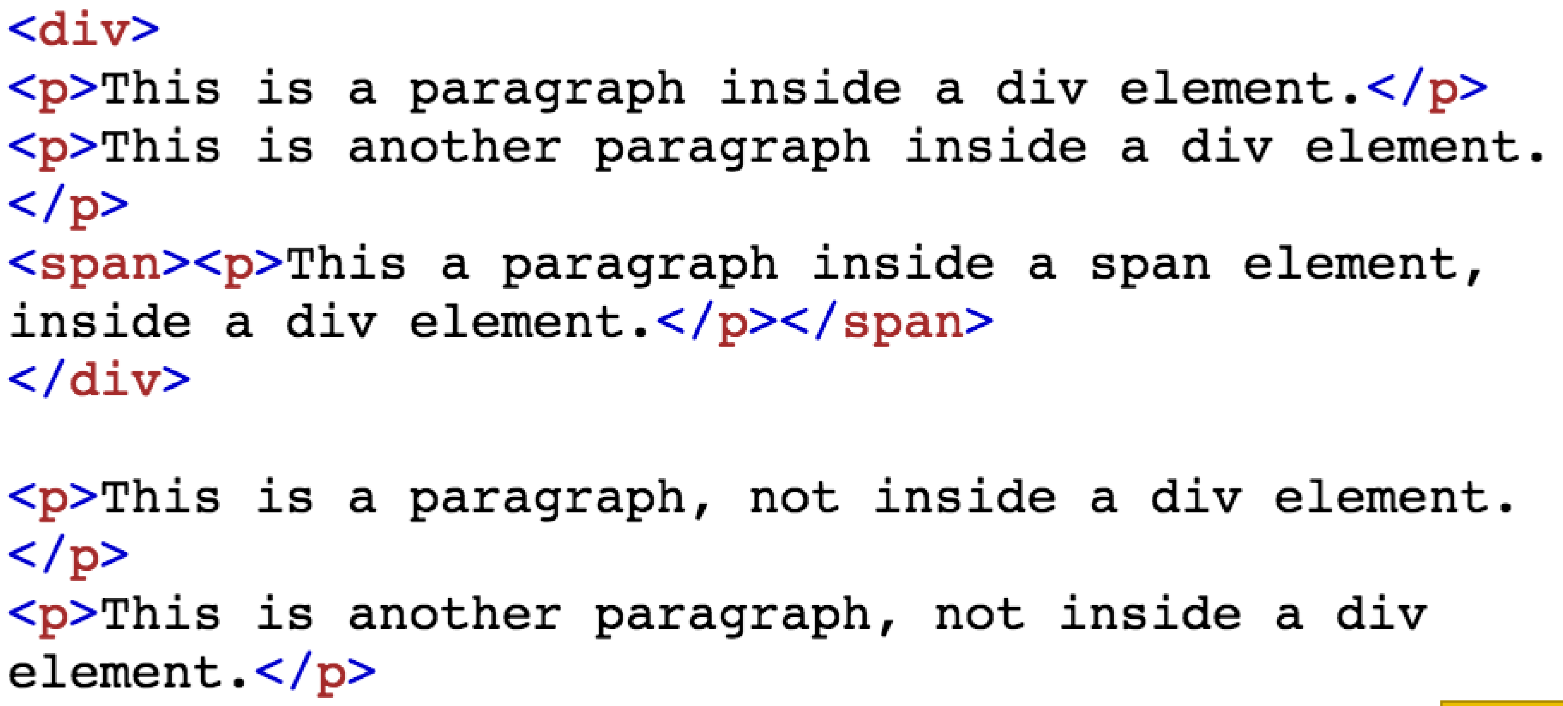
This a paragraph inside a span element, inside a div element.

This is a paragraph, not inside a div element.

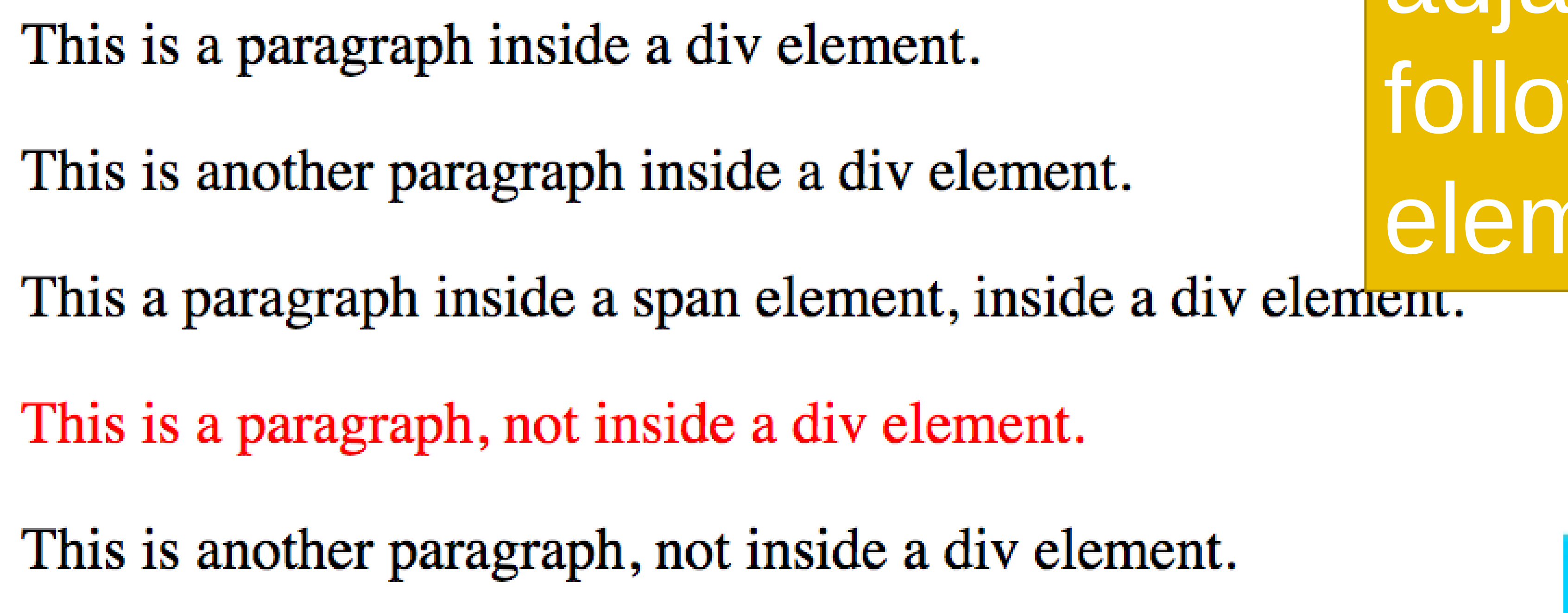
This is another paragraph, not inside a div element.

Change the color of all <p> elements, that are descendants of <div> elements, to "red".

Mini Excercise



```
<div>
<p>This is a paragraph inside a div element.</p>
<p>This is another paragraph inside a div element.
</p>
<span><p>This a paragraph inside a span element,
inside a div element.</p></span>
</div>

<p>This is a paragraph, not inside a div element.
</p>
<p>This is another paragraph, not inside a div
element.</p>
```



This is a paragraph inside a div element.

This is another paragraph inside a div element.

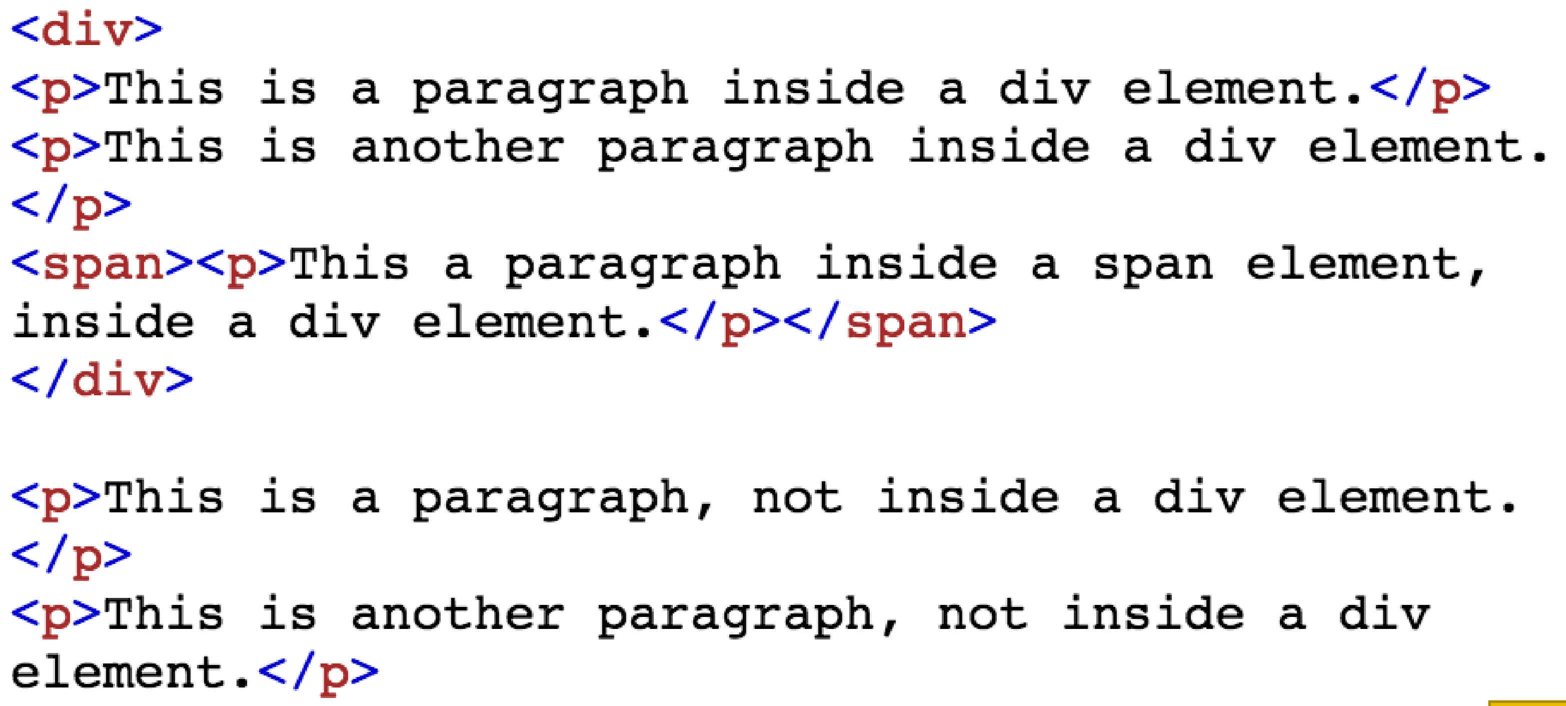
This a paragraph inside a span element, inside a div element.

This is a paragraph, not inside a div element.

This is another paragraph, not inside a div element.

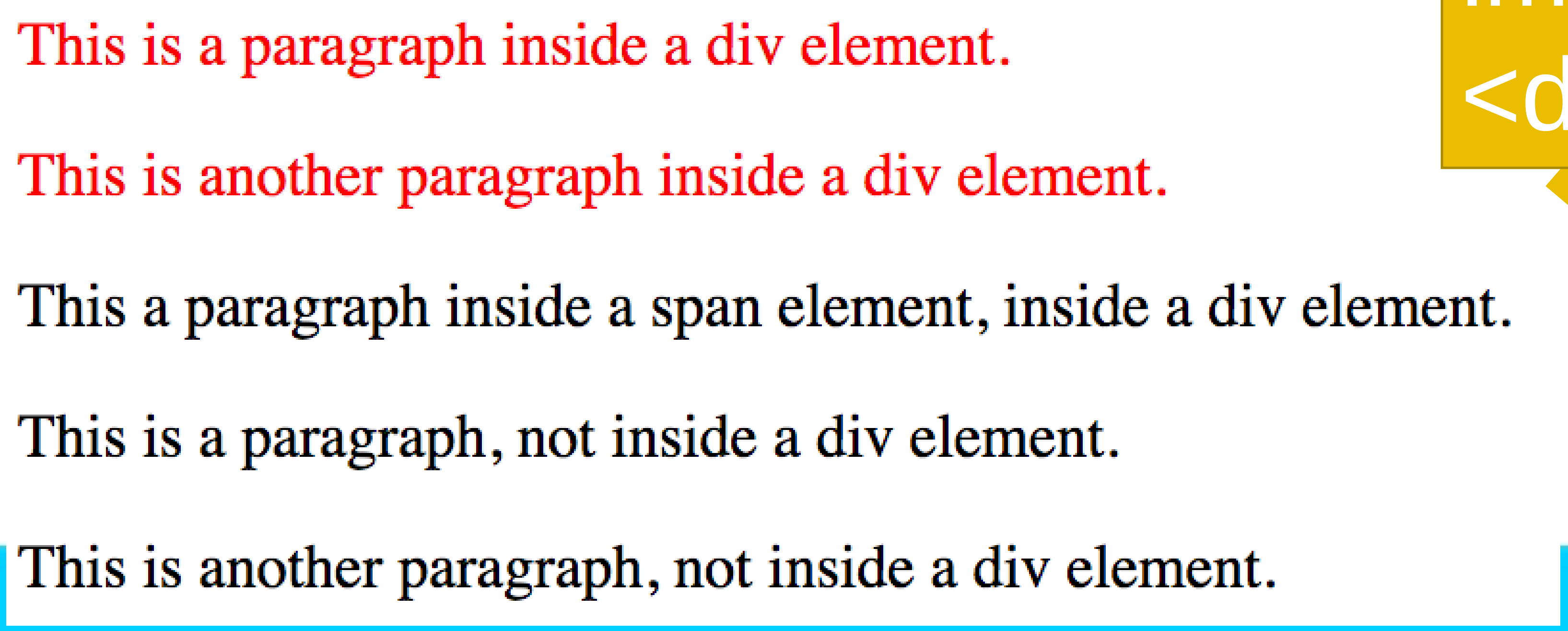
Change the color of <p> elements, that are the adjacent (immediately following) sibling of a <div> element, to "red".

Mini Excercise



```
<div>
<p>This is a paragraph inside a div element.</p>
<p>This is another paragraph inside a div element.
</p>
<span><p>This a paragraph inside a span element,
inside a div element.</p></span>
</div>

<p>This is a paragraph, not inside a div element.
</p>
<p>This is another paragraph, not inside a div
element.</p>
```

Change the color of all <p> elements, that are immediate children of <div> elements, to "red".



This is a paragraph inside a div element.

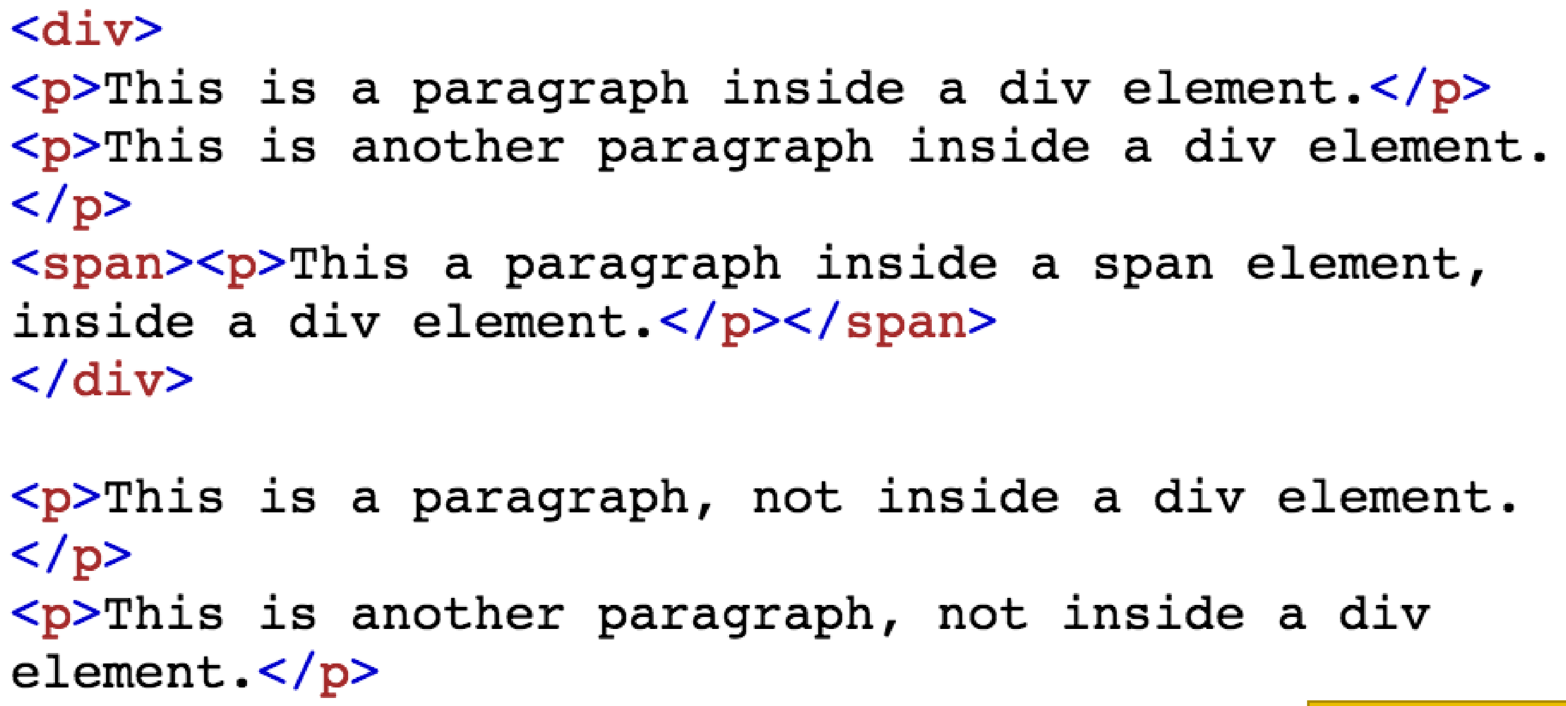
This is another paragraph inside a div element.

This a paragraph inside a span element, inside a div element.

This is a paragraph, not inside a div element.

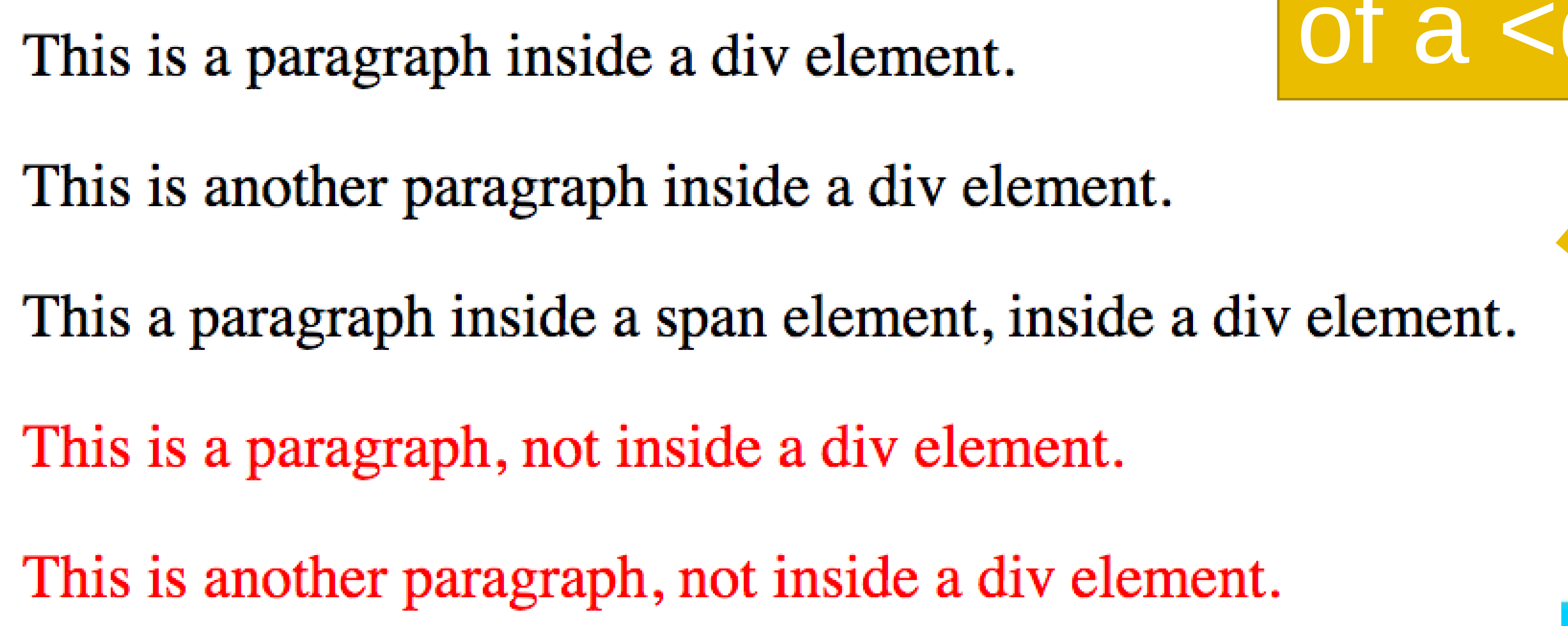
This is another paragraph, not inside a div element.

Mini Excercise



```
<div>
<p>This is a paragraph inside a div element.</p>
<p>This is another paragraph inside a div element.
</p>
<span><p>This a paragraph inside a span element,
inside a div element.</p></span>
</div>

<p>This is a paragraph, not inside a div element.
</p>
<p>This is another paragraph, not inside a div
element.</p>
```

Change the color of <p> elements, that are the siblings of a <div> element, to "red".



This is a paragraph inside a div element.

This is another paragraph inside a div element.

This a paragraph inside a span element, inside a div element.

This is a paragraph, not inside a div element.

This is another paragraph, not inside a div element.

Attribute selectors

Attribute selectors selects elements based upon the attributes present in the HTML Tags and their value.

```
IMG[src="small.gif"] {
        border: 1px solid #000;
  }
```

will work for

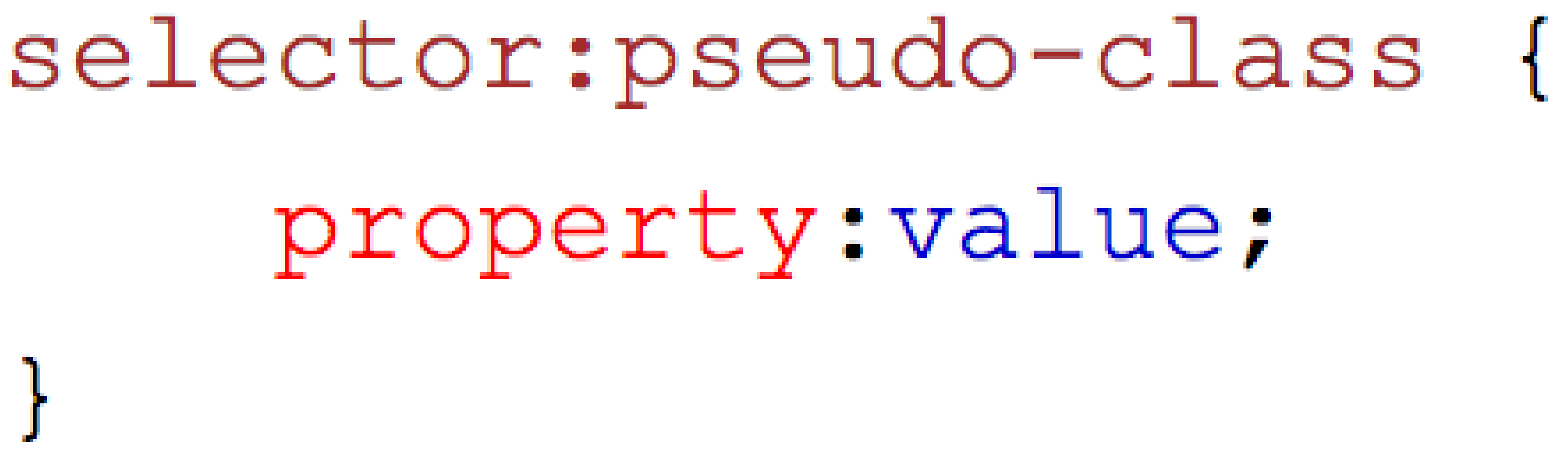
```
<img src=“small.gif” />
```

CSS Pseudo-classes

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- ➢Style an element when a user mouse over it
- ➢Style visited and unvisited links differently
- ➢Style an element when it gets focus



```
selector:pseudo-class {
    property:value;
}
```

CSS Pseudo-classes

Selector	Example	Example description
:active	a:active	Selects the active link
:checked	input:checked	Selects every checked <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent

CSS Pseudo-classes

Selector	Example	Example description
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the <input> element that has focus
:hover	a:hover	Selects links on mouse over

CSS Pseudo-classes

Selector	Example	Example description
:in-range	input:in-range	Selects <input> elements with a value within a specified range
:invalid	input:invalid	Selects all <input> elements with an invalid value
:lang(*language*)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
:last-child	p:last-child	Selects every <p> elements that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent

CSS Pseudo-classes

Selector	Example	Example description
:link	a:link	Selects all unvisited links
:not(selector)	:not(p)	Selects every element that is not a <p> element
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child

CSS Pseudo-classes

Selector	Example	Example description
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent

CSS Pseudo-classes

Selector	Example	Example description
:optional	input:optional	Selects <input> elements with no "required" attribute
:out-of-range	input:out-of-range	Selects <input> elements with a value outside a specified range
:read-only	input:read-only	Selects <input> elements with a "readonly" attribute specified
:read-write	input:read-write	Selects <input> elements with no "readonly" attribute

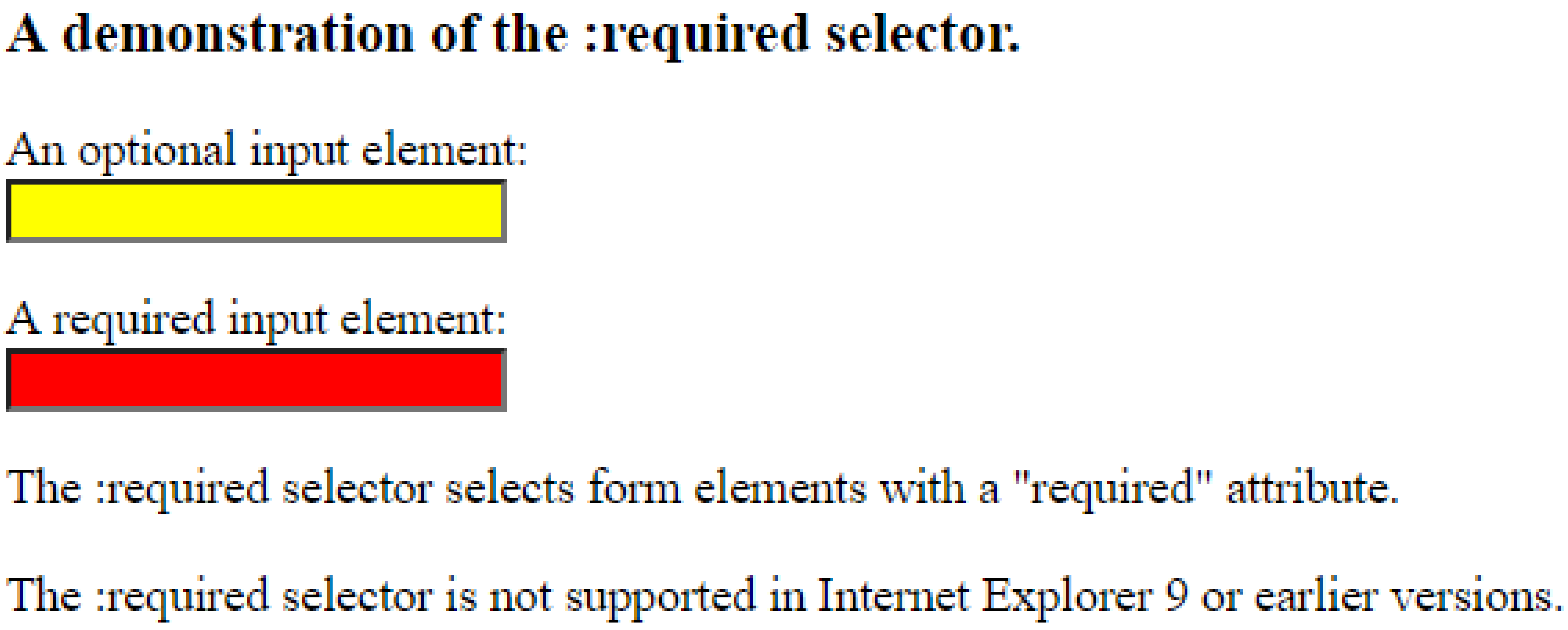
CSS Pseudo-classes

Selector	Example	Example description
:required	input:required	Selects <input> elements with a "required" attribute specified
:root	root	Selects the document's root element
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	input:valid	Selects all <input> elements with a valid value
:visited	a:visited	Selects all visited links

CSS Pseudo-classes

```
<h3>A demonstration of the :optional selector.</h3>

<p>An optional input element:<br><input></p>

<p>A required input element:<br><input required></p>

<p>The :optional selector selects form elements with no
"required" attribute.</p>

<p><strong>Note:</strong> The :optional selector is not supported
in Internet Explorer 9 or earlier versions.</p>
```

```
<style>

/* CSS set Background color
   is red for input item */
input:required {
    background-color: Red;
}

input:Optional {
    background-color: yellow;
}

</style>
```



A demonstration of the :required selector.

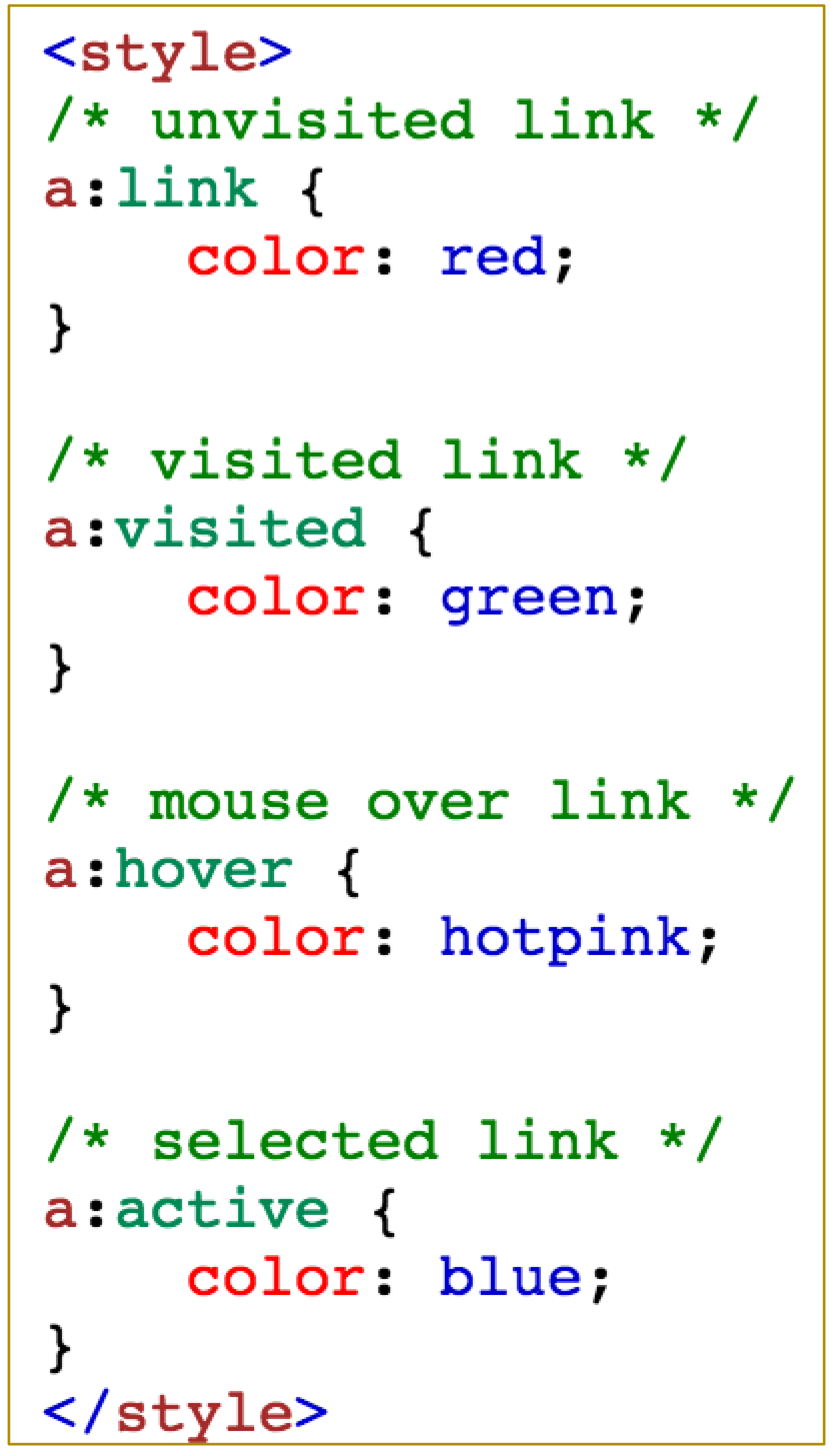
An optional input element:

A required input element:

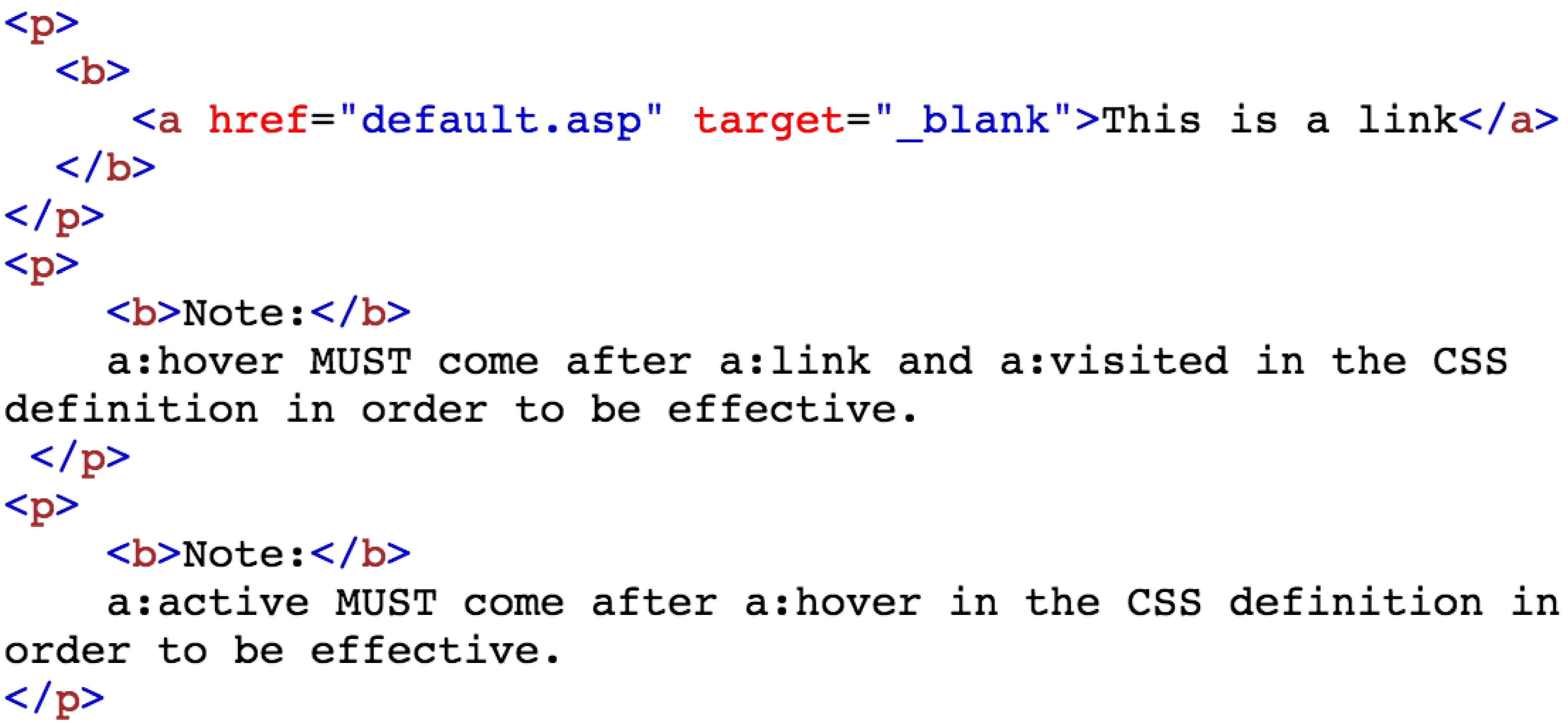
The :required selector selects form elements with a "required" attribute.

The :required selector is not supported in Internet Explorer 9 or earlier versions.

CSS Pseudo-classes



```
<style>
/* unvisited link */
a:link {
    color: red;
}

/* visited link */
a:visited {
    color: green;
}

/* mouse over link */
a:hover {
    color: hotpink;
}

/* selected link */
a:active {
    color: blue;
}
</style>
```



```
<p>
  <b>
     <a href="default.asp" target="_blank">This is a link</a>
  </b>
</p>
<p>
    <b>Note:</b>
    a:hover MUST come after a:link and a:visited in the CSS
definition in order to be effective.
 </p>
<p>
    <b>Note:</b>
    a:active MUST come after a:hover in the CSS definition in
order to be effective.
</p>
```

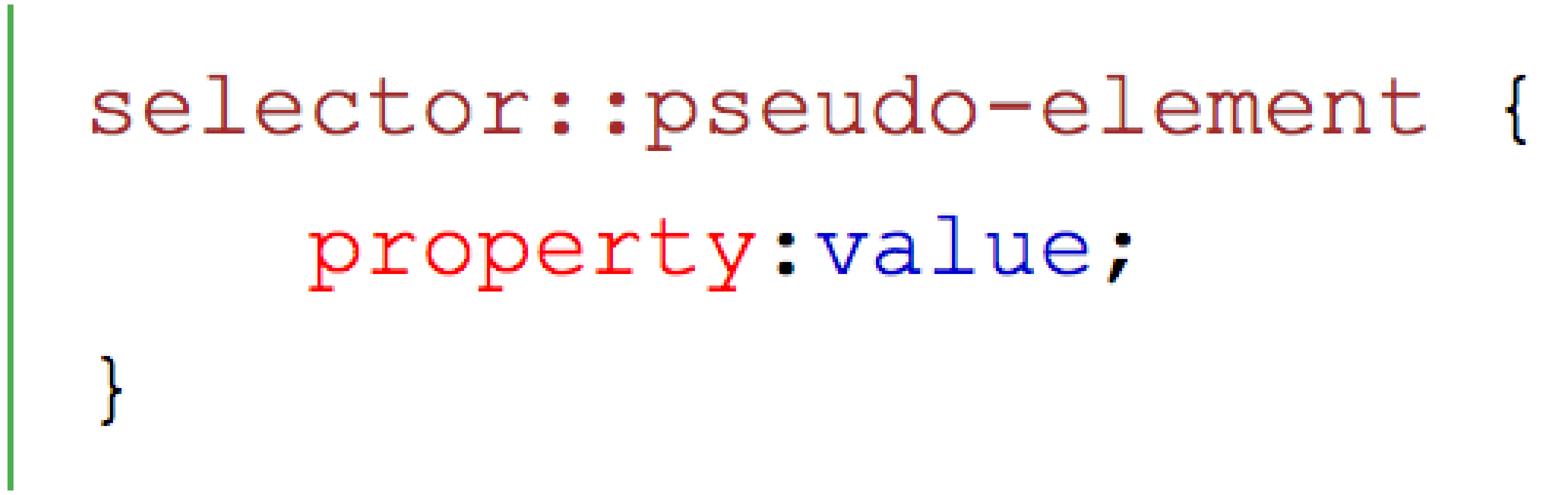
https://www.w3schools.com/css/tryit.asp?filename=trycss_link

CSS Pseudo-element

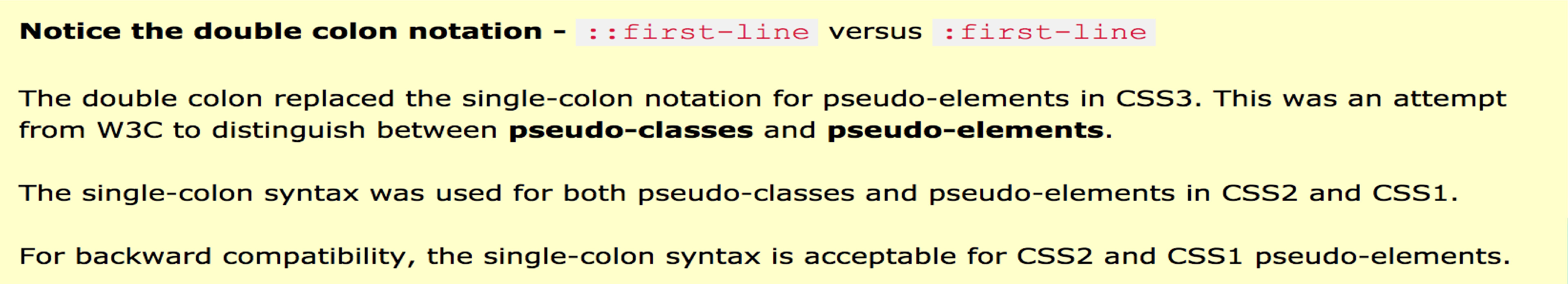
A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

- ➢Style the first letter, or line, of an element
- ➢Insert content before, or after, the content of an element



```
selector::pseudo-element {
    property:value;
}
```



Notice the double colon notation - `::first-line` versus `:first-line`

The double colon replaced the single-colon notation for pseudo-elements in CSS3. This was an attempt from W3C to distinguish between **pseudo-classes** and **pseudo-elements**.

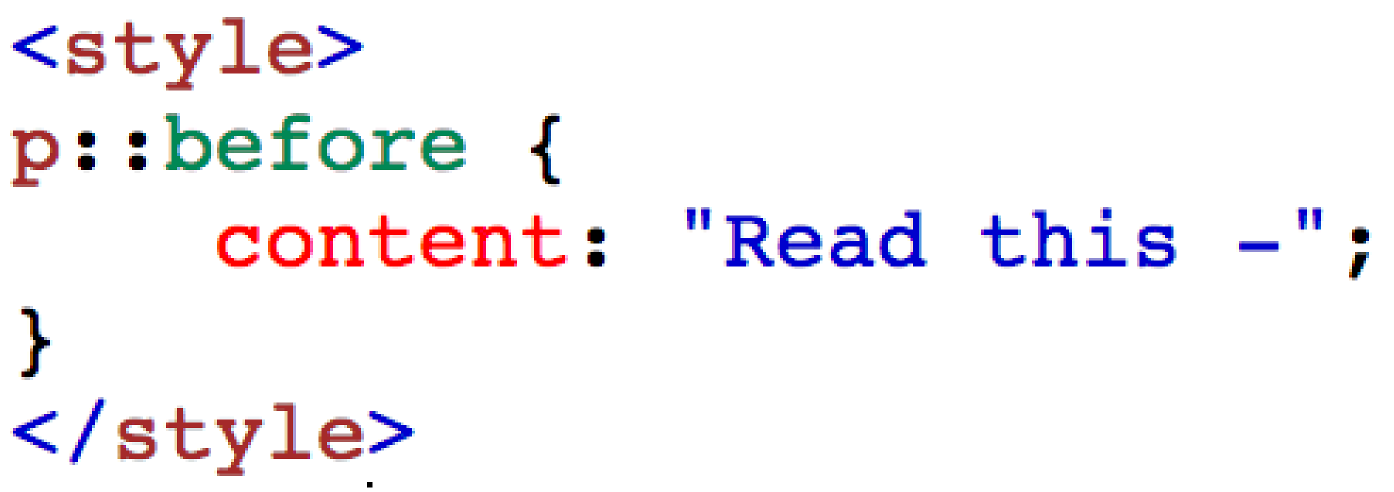
The single-colon syntax was used for both pseudo-classes and pseudo-elements in CSS2 and CSS1.

For backward compatibility, the single-colon syntax is acceptable for CSS2 and CSS1 pseudo-elements.

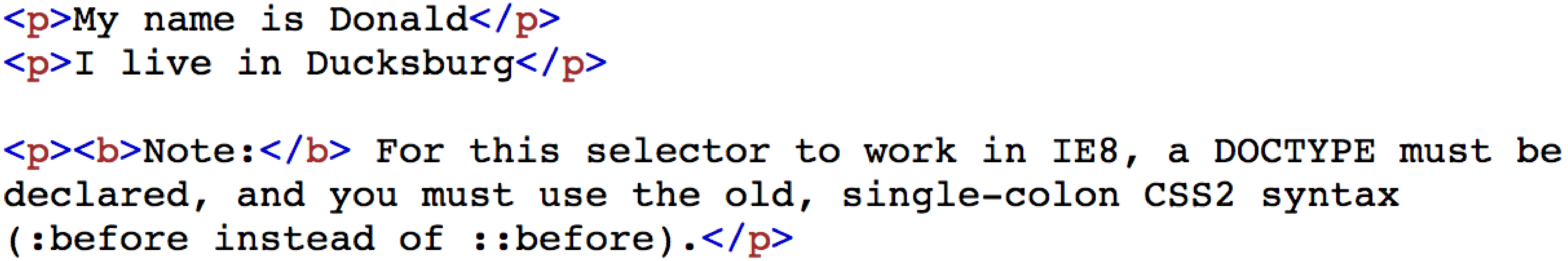
CSS Pseudo-element

Selector	Example	Example description
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
::first-letter	p::first-letter	Selects the first letter of each <p> element
::first-line	p::first-line	Selects the first line of each <p> element
::selection	p::selection	Selects the portion of an element that is selected by a user

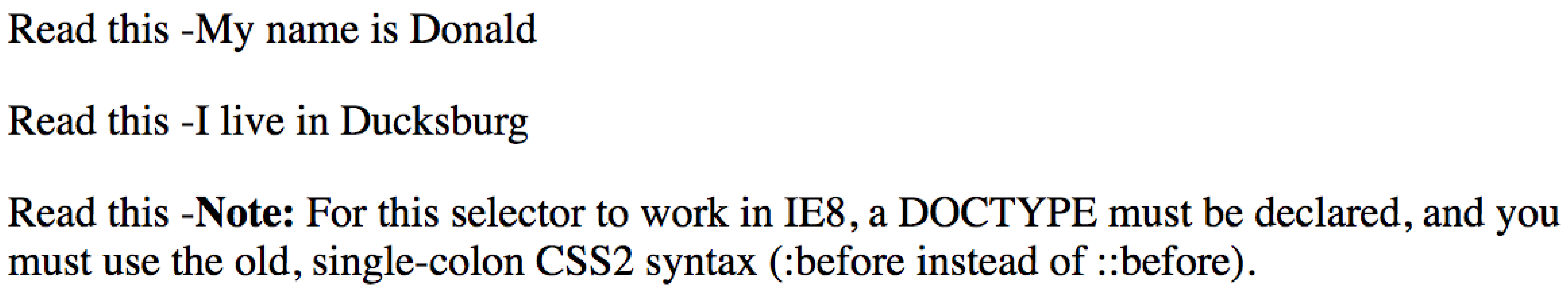
CSS Pseudo-element



```
<style>
p::before {
    content: "Read this -";
}
</style>
```



```
<p>My name is Donald</p>
<p>I live in Ducksburg</p>

<p><b>Note:</b> For this selector to work in IE8, a DOCTYPE must be
declared, and you must use the old, single-colon CSS2 syntax
(:before instead of ::before).</p>
```

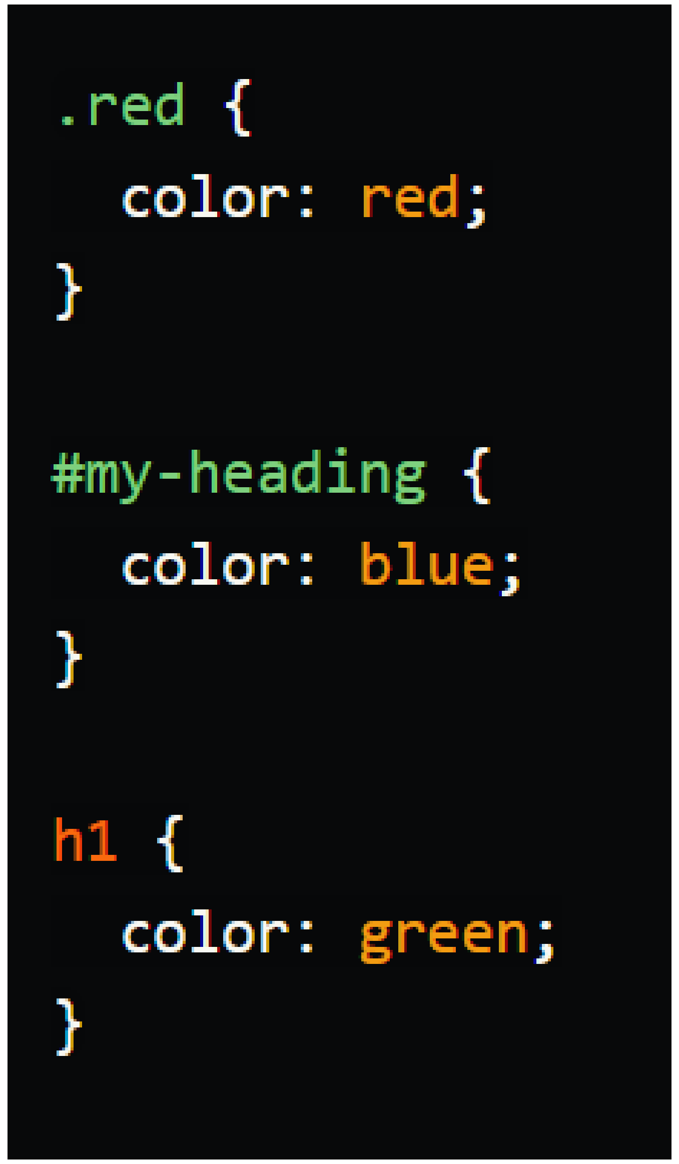


Read this -My name is Donald

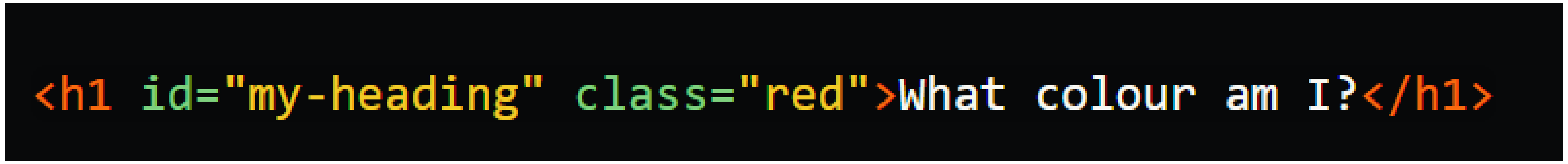
Read this -I live in Ducksburg

Read this -**Note:** For this selector to work in IE8, a DOCTYPE must be declared, and you must use the old, single-colon CSS2 syntax (:before instead of ::before).

CSS Cascade



```css
.red {
  color: red;
}

#my-heading {
  color: blue;
}

h1 {
  color: green;
}
```

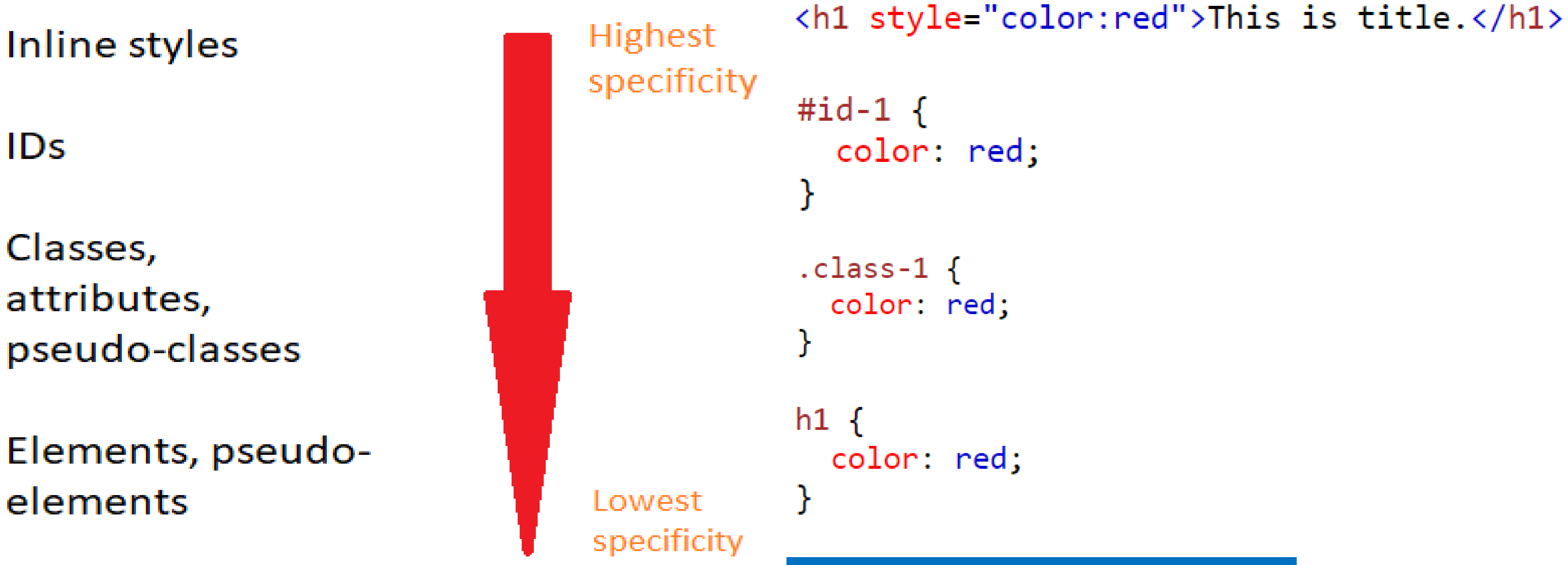


```html
<h1 id="my-heading" class="red">What colour am I?</h1>
```

What colour am I?

CSS Cascade

Cascade: Is an algorithm that defines how to combine property values originating from different sources



Inline styles

IDs

Classes, attributes, pseudo-classes

Elements, pseudo-elements

Highest specificity

Lowest specificity

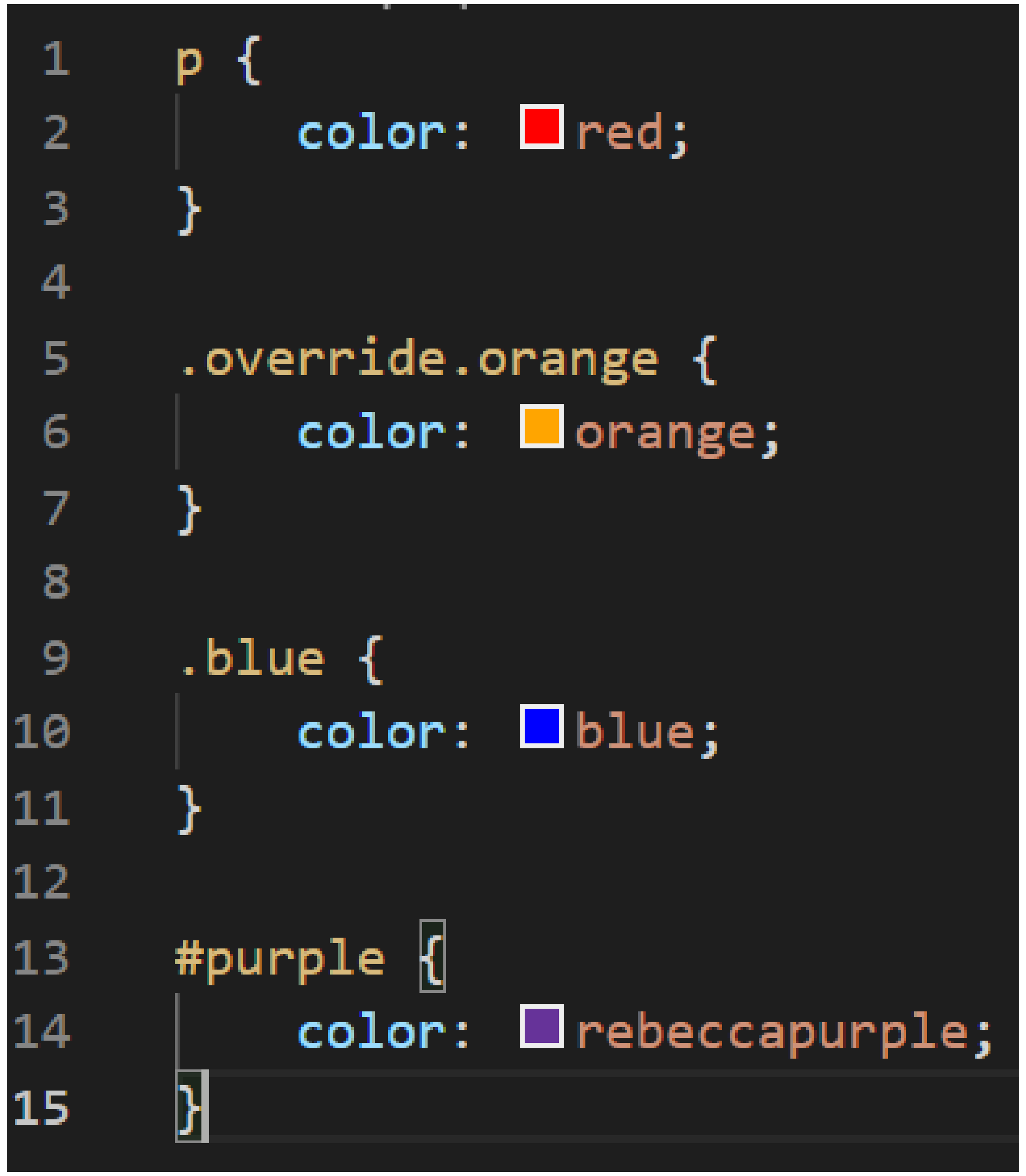
```
<h1 style="color:red">This is title.</h1>

#id-1 {
  color: red;
}

.class-1 {
  color: red;
}

h1 {
  color: red;
}
```

NOTE: !important

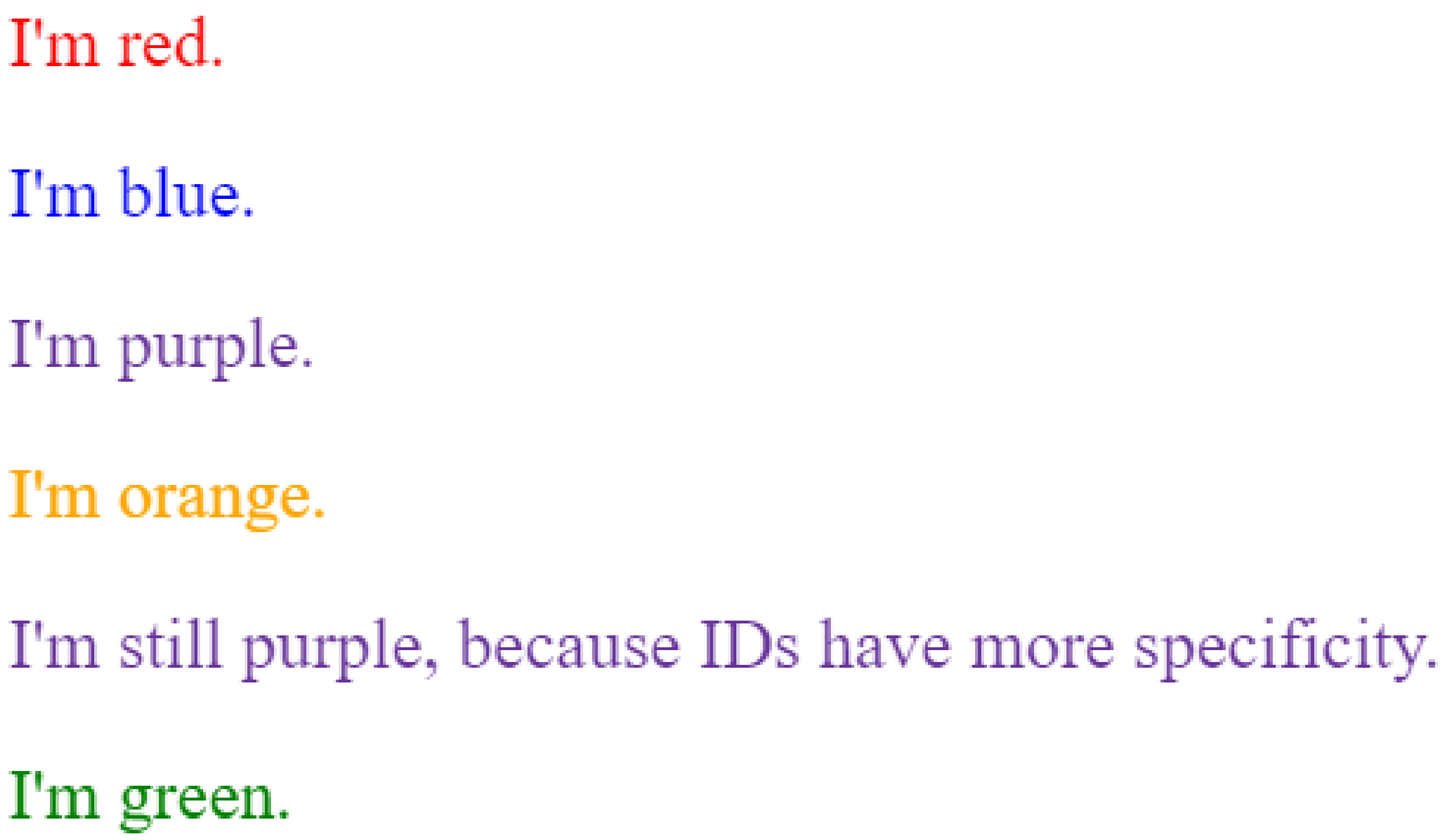
CSS Cascade

- Classes are more **specific** than element selectors
- IDs are more **specific** than classes.
- Combined selectors are more **specific** than single selectors.
- Inline styles are more **specific** than stuff in stylesheets.

CSS Cascade



```css
p {
    color: 🟥red;
}

.override.orange {
    color: 🟧orange;
}

.blue {
    color: 🟦blue;
}

#purple {
    color: 🟪rebeccapurple;
}
```



I'm red.

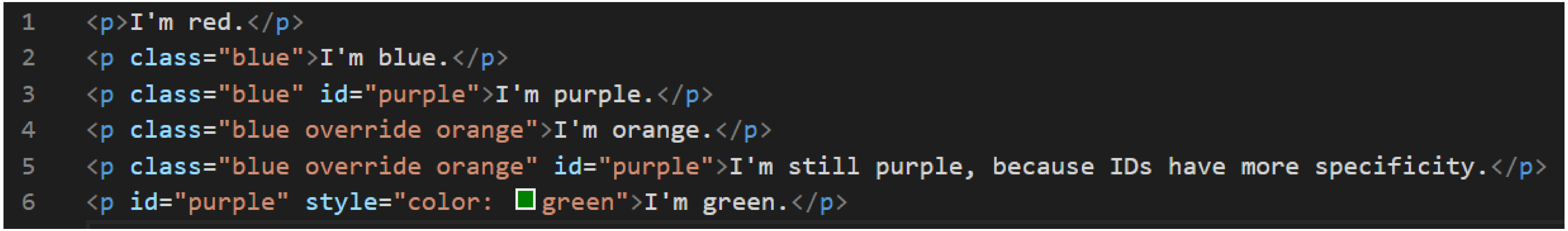
I'm blue.

I'm purple.

I'm orange.

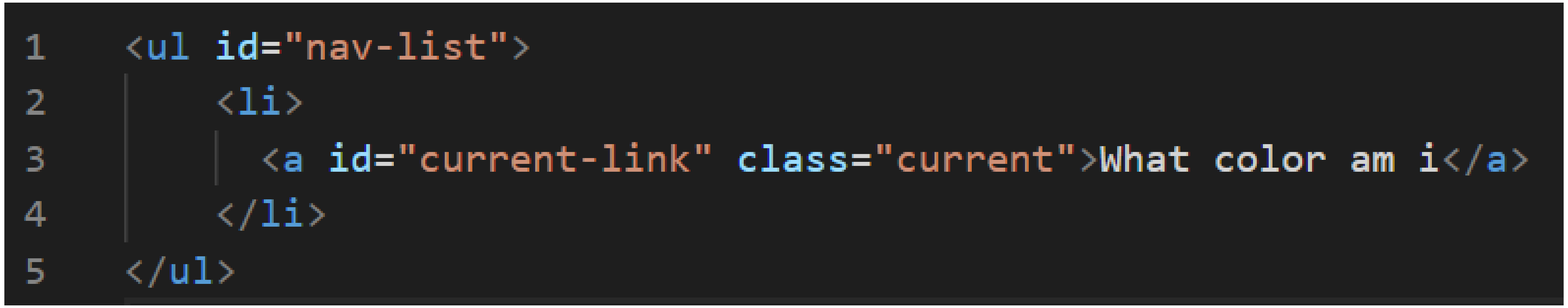
I'm still purple, because IDs have more specificity.

I'm green.

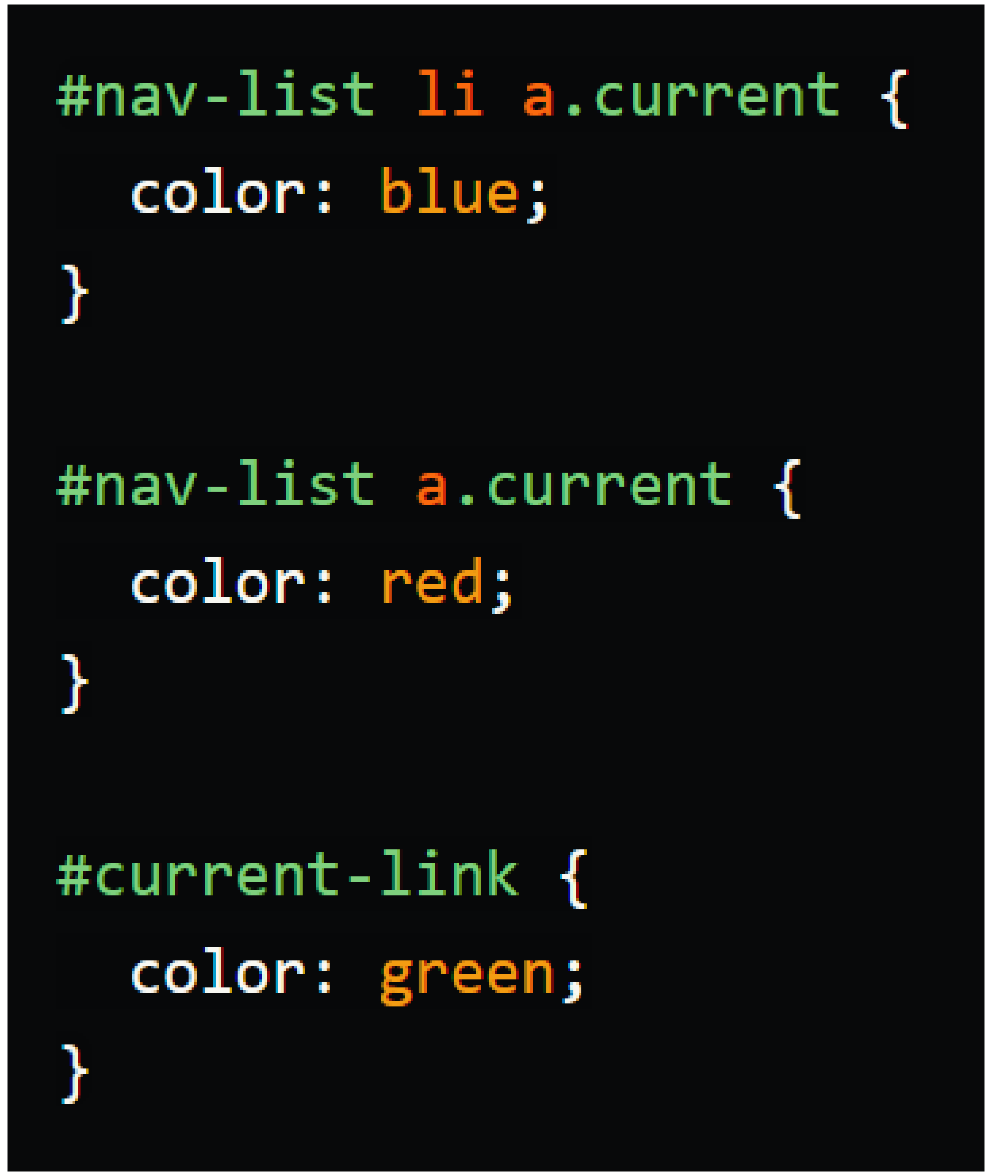


```html
<p>I'm red.</p>
<p class="blue">I'm blue.</p>
<p class="blue" id="purple">I'm purple.</p>
<p class="blue override orange">I'm orange.</p>
<p class="blue override orange" id="purple">I'm still purple, because IDs have more specificity.</p>
<p id="purple" style="color: 🟩green">I'm green.</p>
```

CSS Cascade



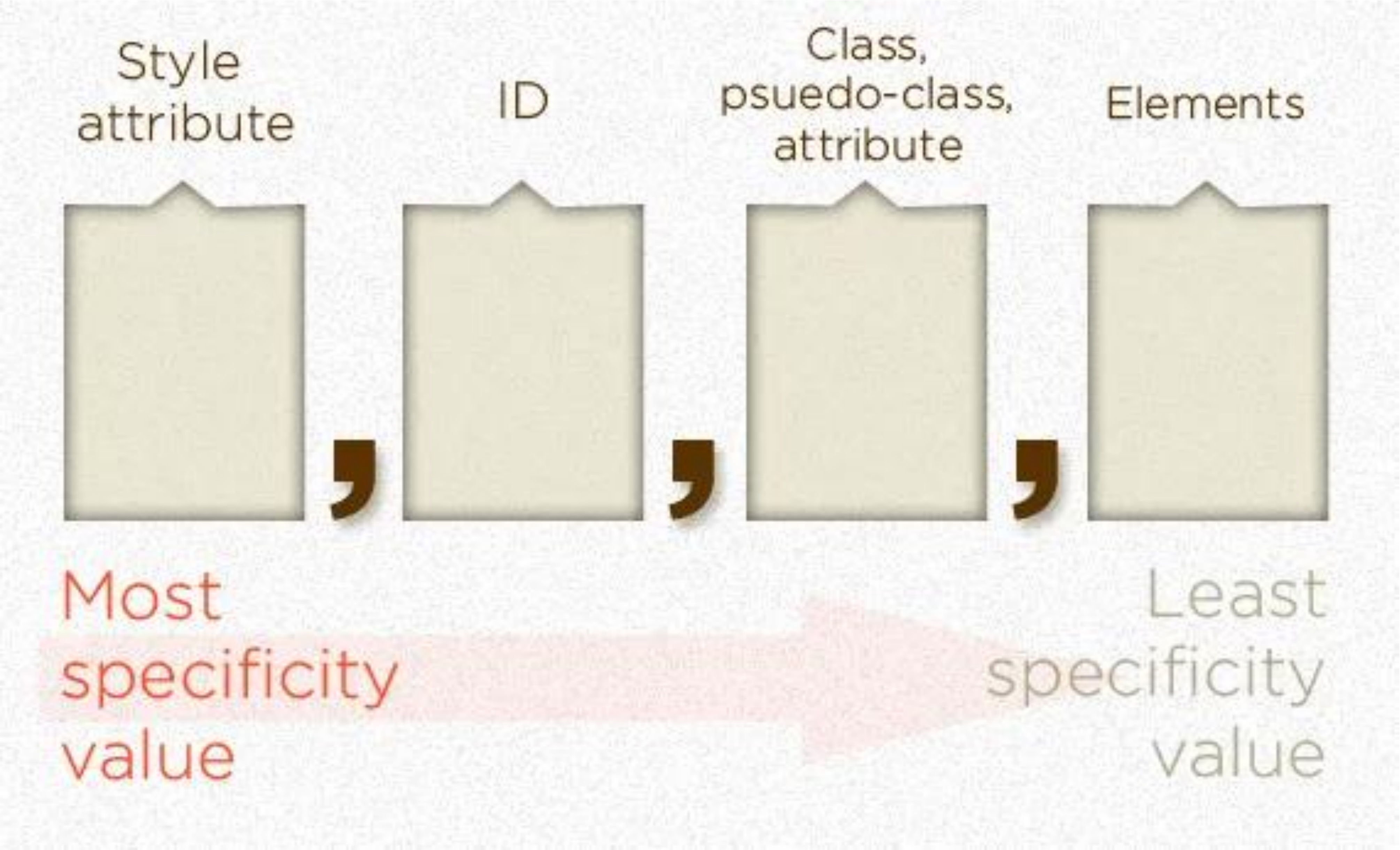
```html
<ul id="nav-list">
    <li>
        <a id="current-link" class="current">What color am i</a>
    </li>
</ul>
```



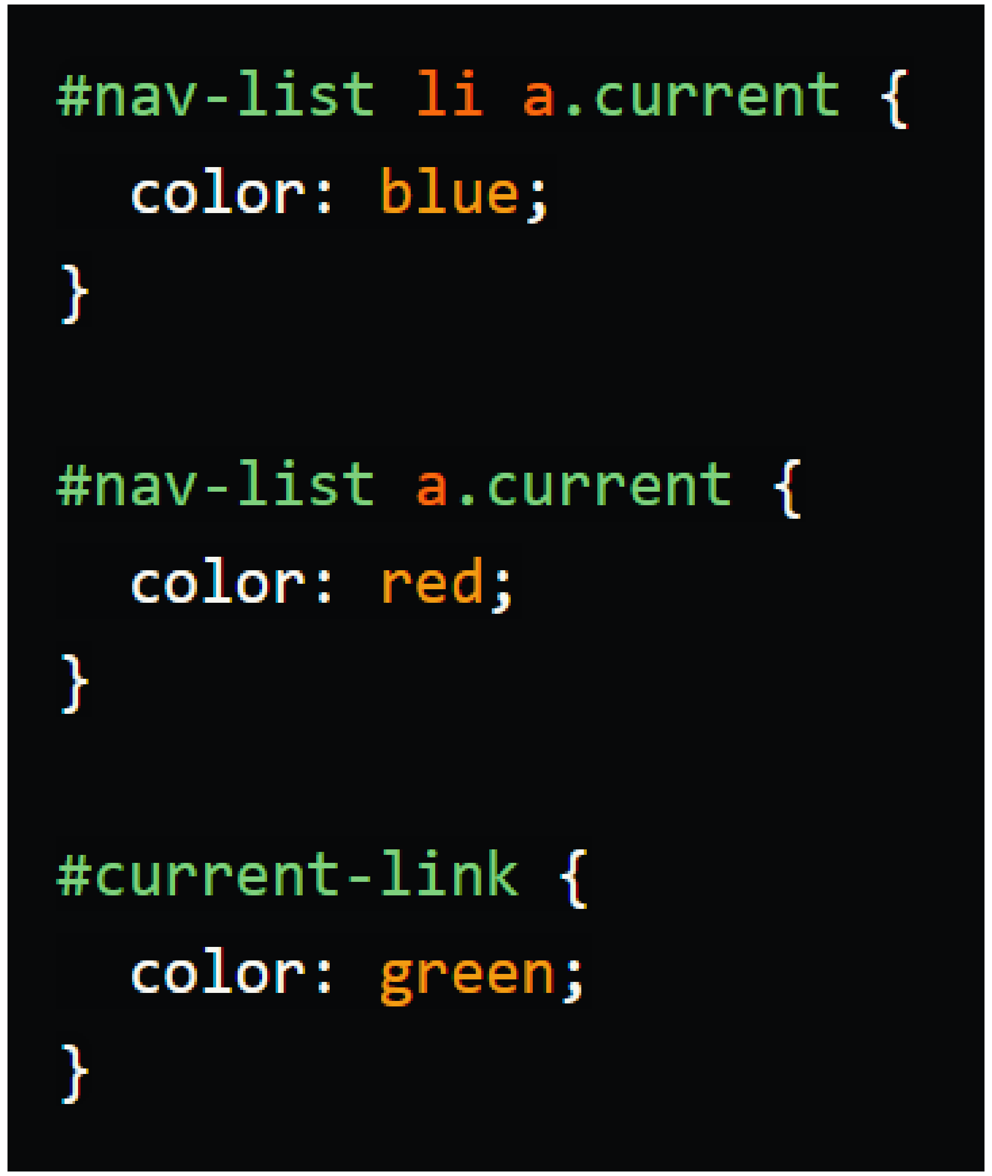
```css
#nav-list li a.current {
  color: blue;
}

#nav-list a.current {
  color: red;
}

#current-link {
  color: green;
}
```

What colour am I?

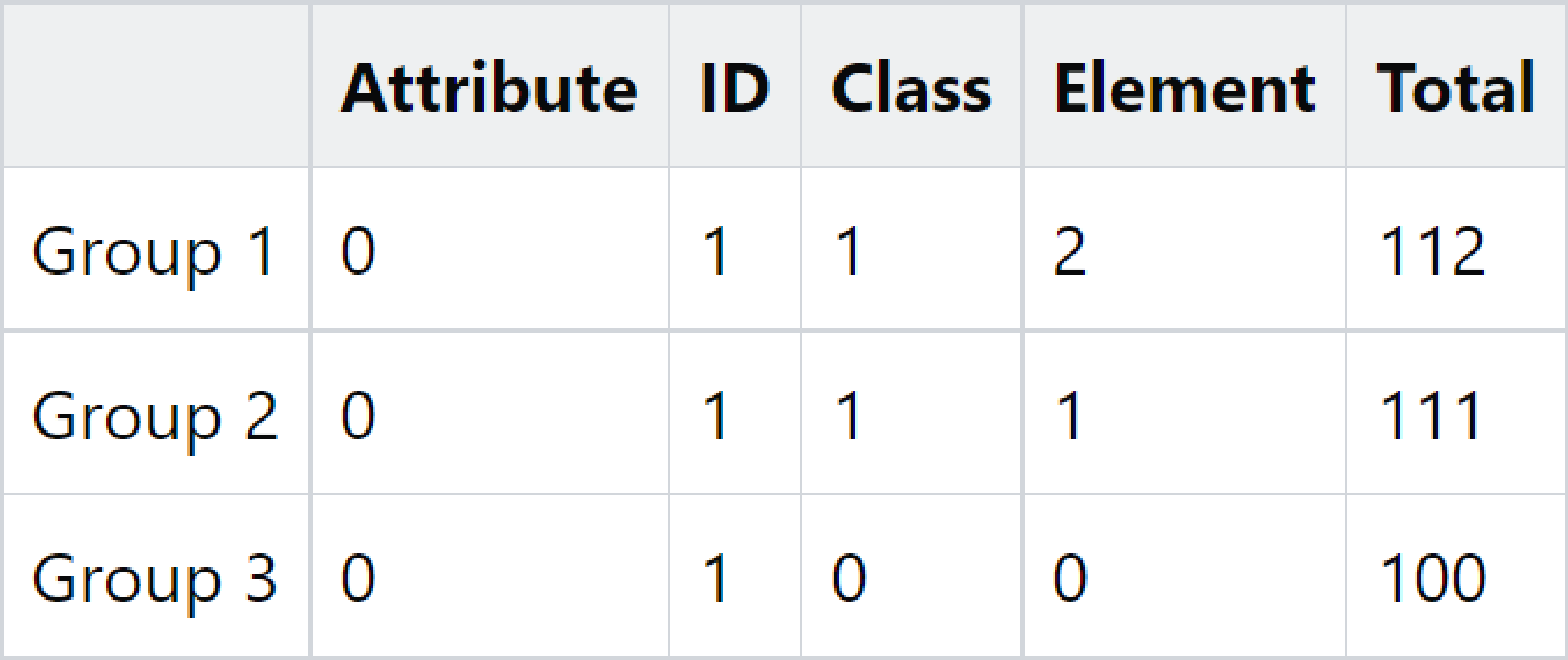
CSS Cascade


Style attribute
ID
Class, psuedo-class, attribute
Elements
,
,
,
Most specificity value
Least specificity value

CSS Cascade



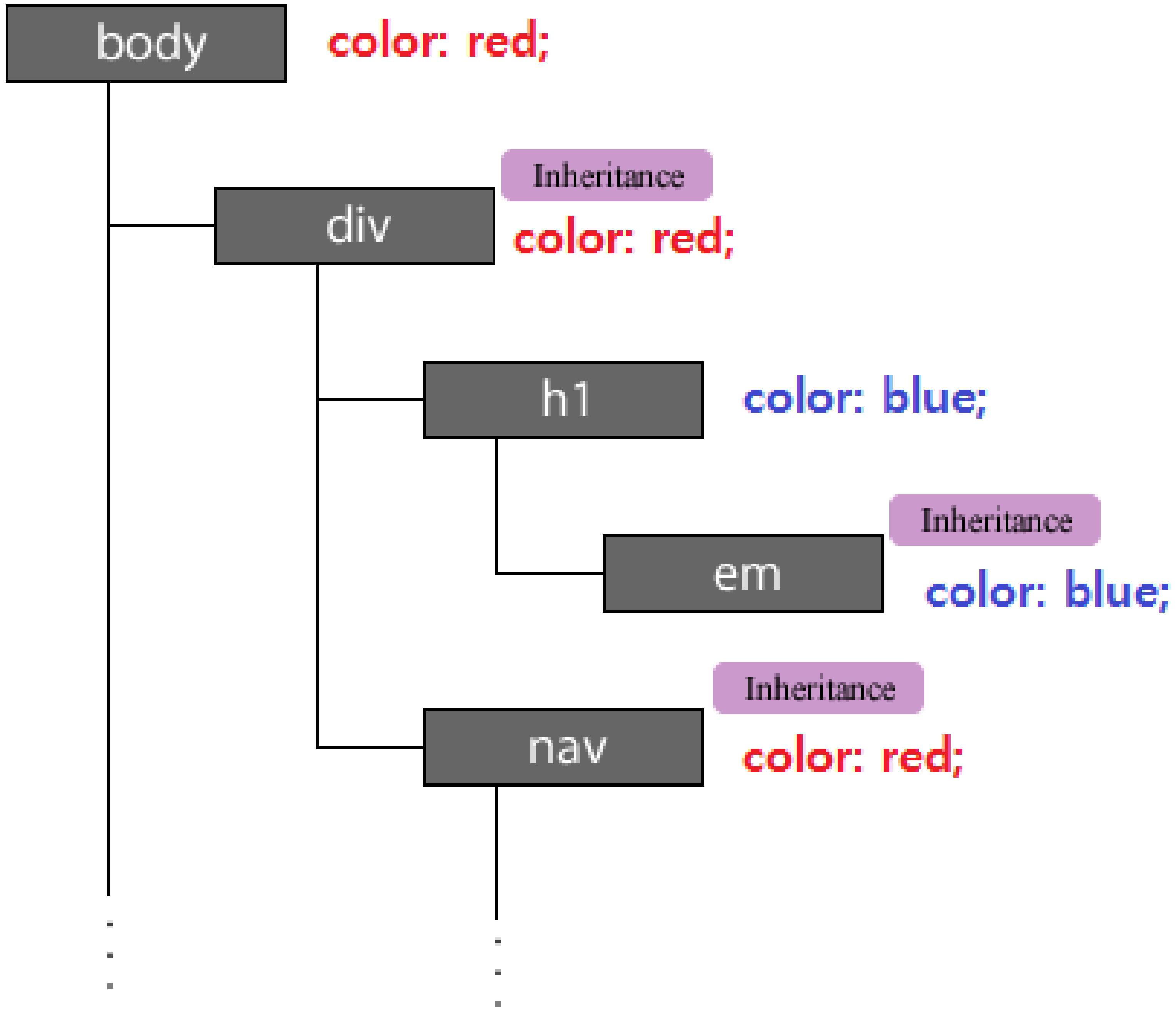
```
#nav-list li a.current {
  color: blue;
}

#nav-list a.current {
  color: red;
}

#current-link {
  color: green;
}
```

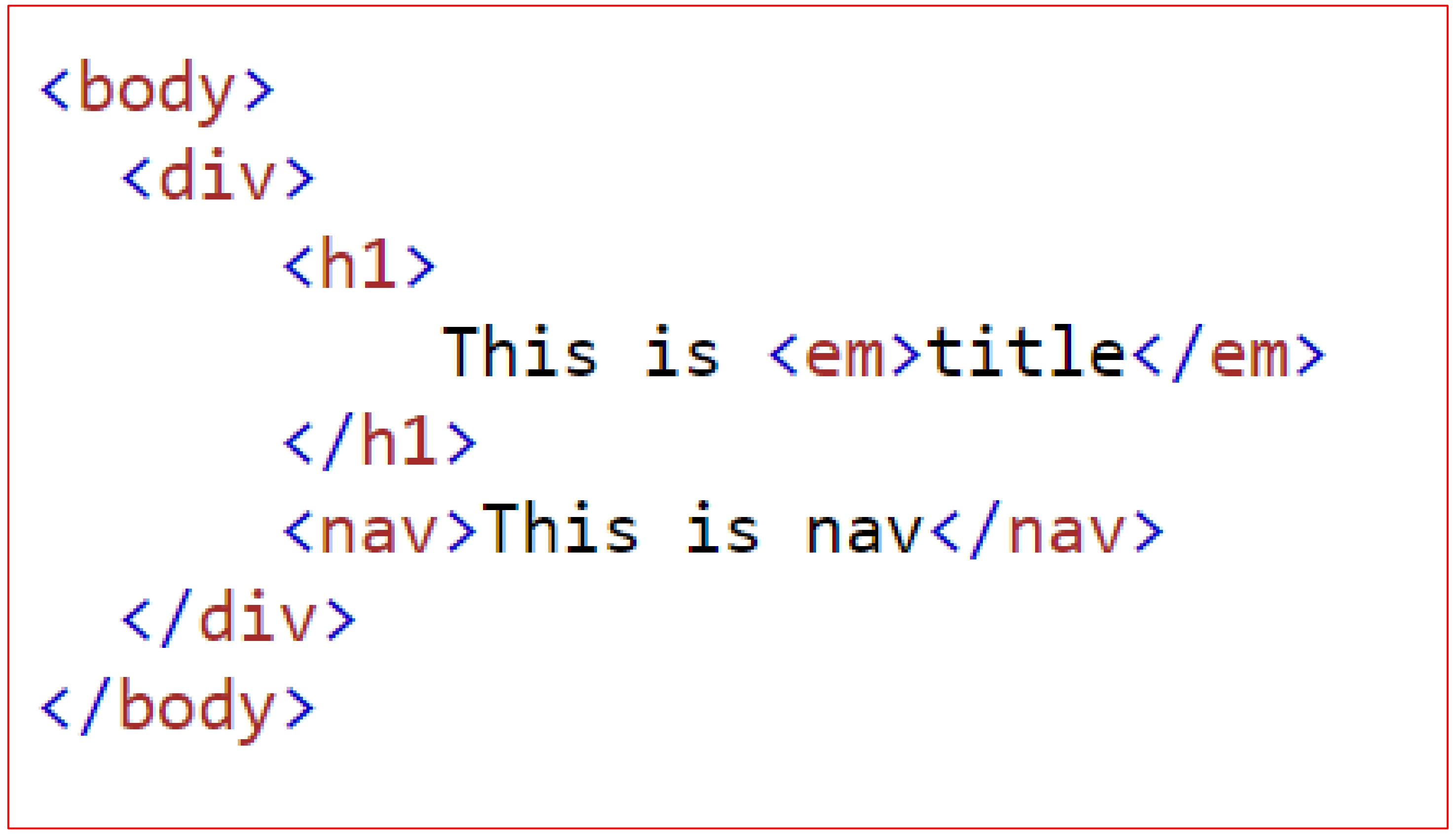


	Attribute	ID	Class	Element	Total
Group 1	0	1	1	2	112
Group 2	0	1	1	1	111
Group 3	0	1	0	0	100

CSS Inheritance

Inheritance works on a property by property basis.
When you set properties on a selector in CSS, they're either inherited by all the children of that selector or they're not





```
<body>
  <div>
      <h1>
          This is <em>title</em>
      </h1>
      <nav>This is nav</nav>
  </div>
</body>
```

Thank you!

Any questions?

Until You Get It Right

Keep Training and Keep Learning