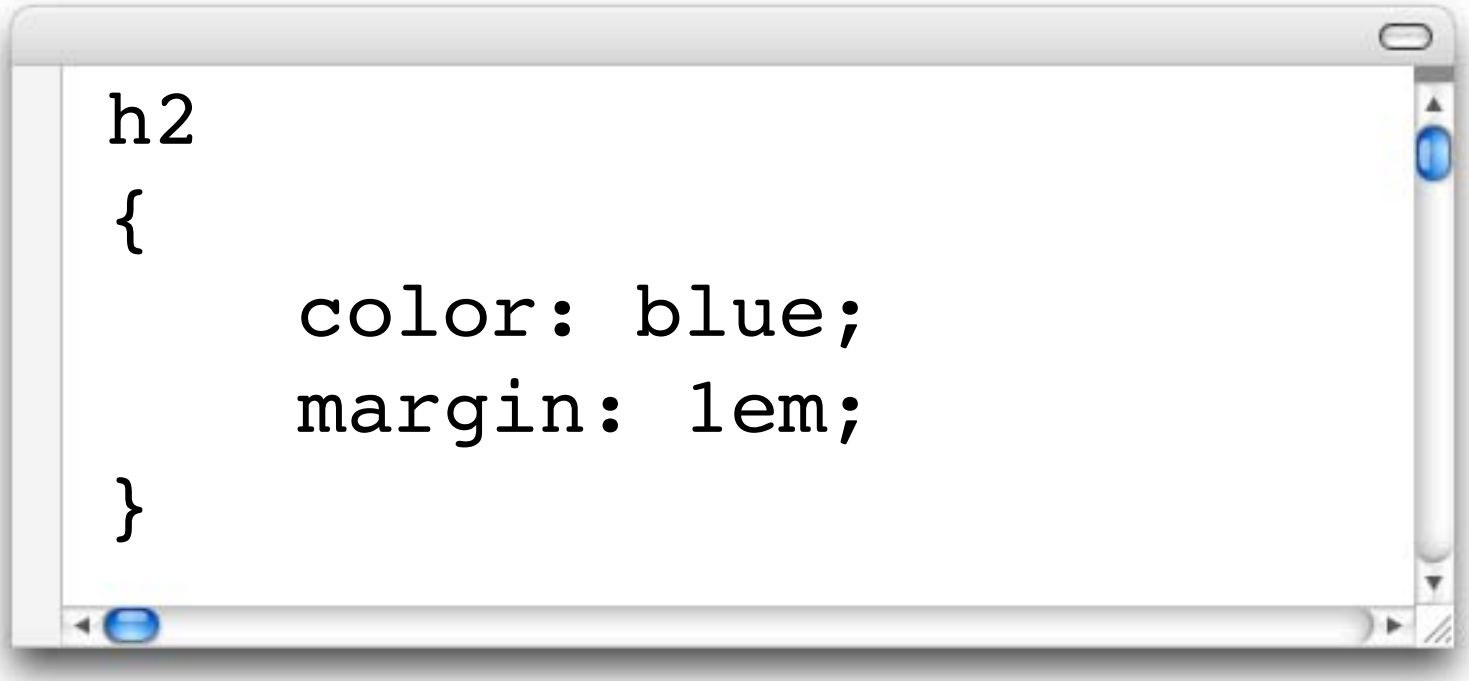
CSS

CASCADE

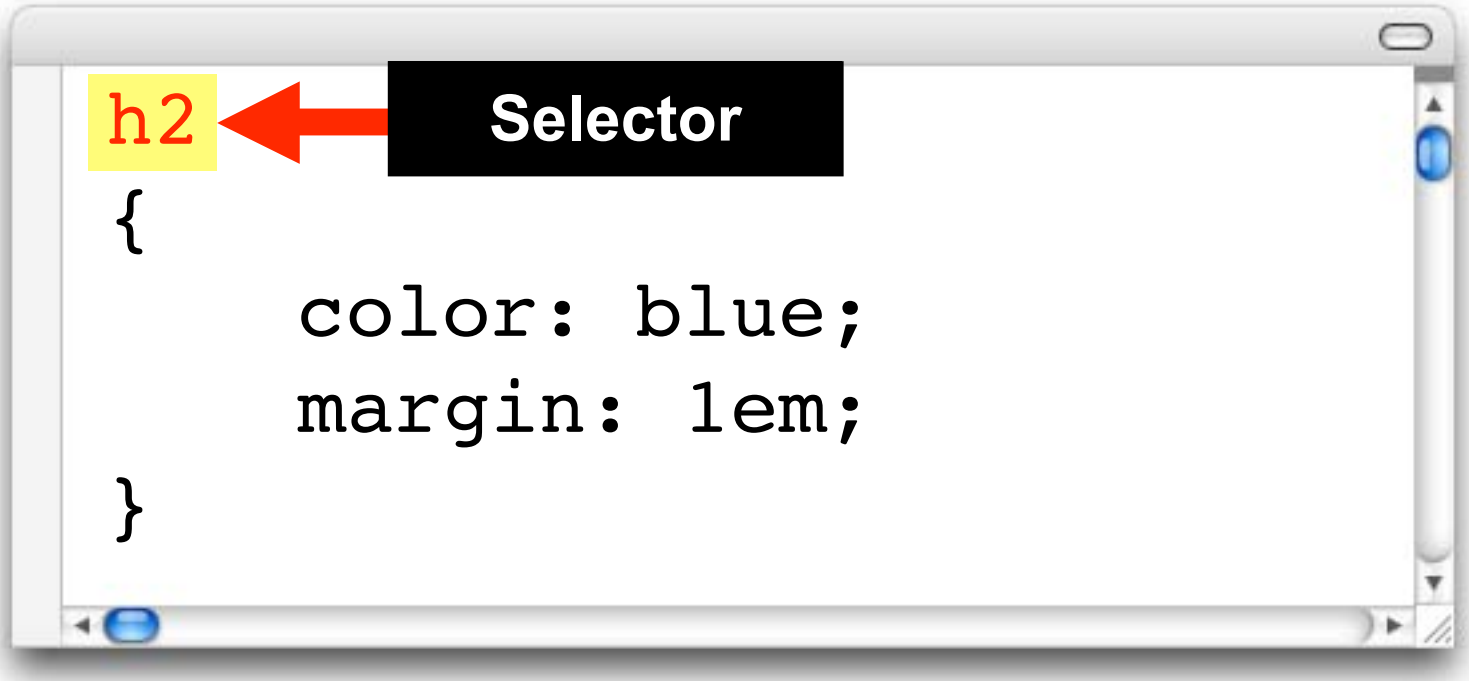
A quick background on CSS rules

CSS rules tell browsers how to render elements in an HTML document.

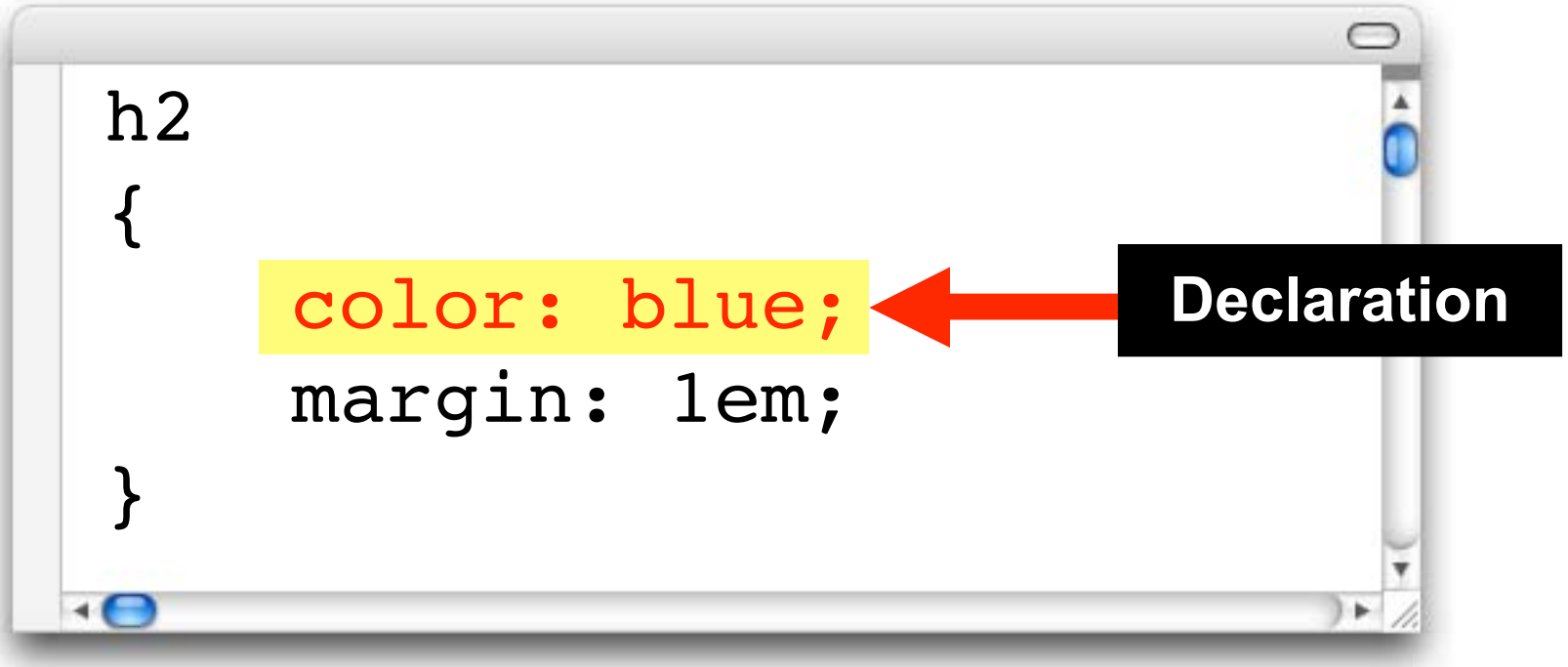


```
h2
{
    color: blue;
    margin: 1em;
}
```

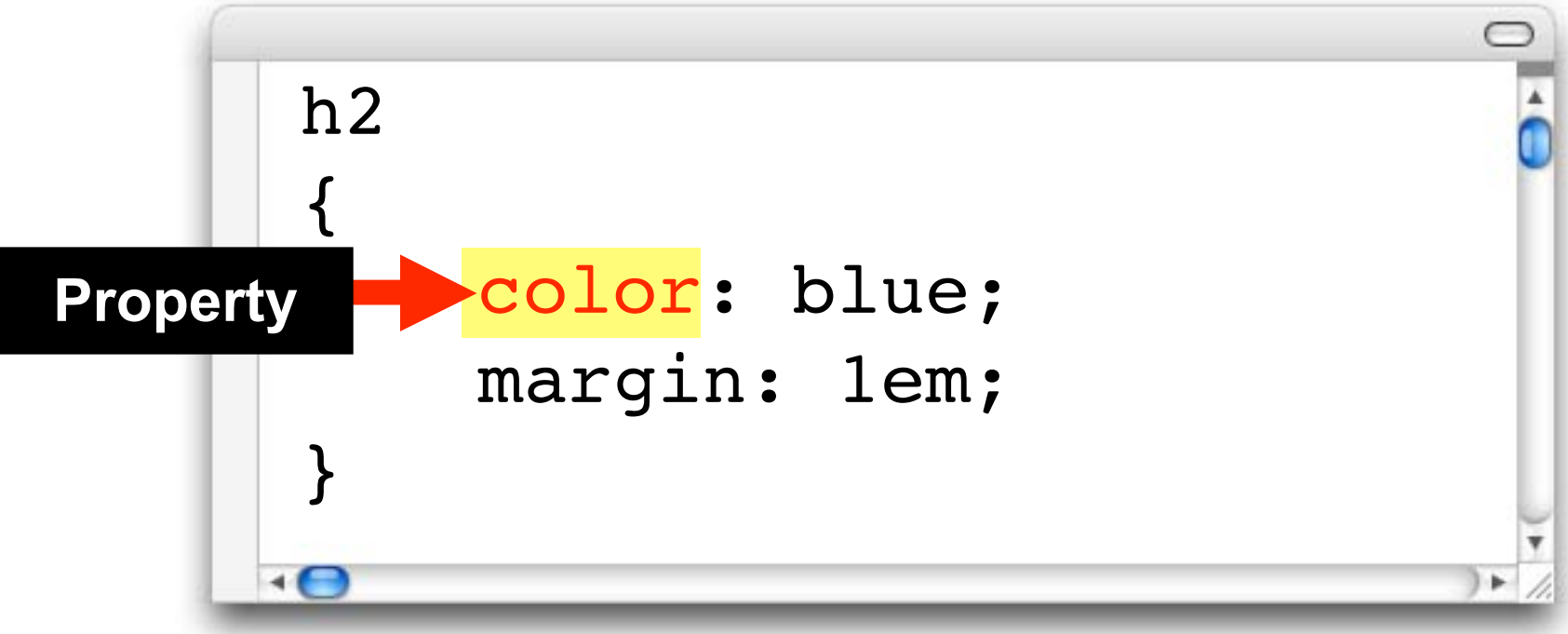
The **selector** "selects" the elements in an HTML document that are to be styled.



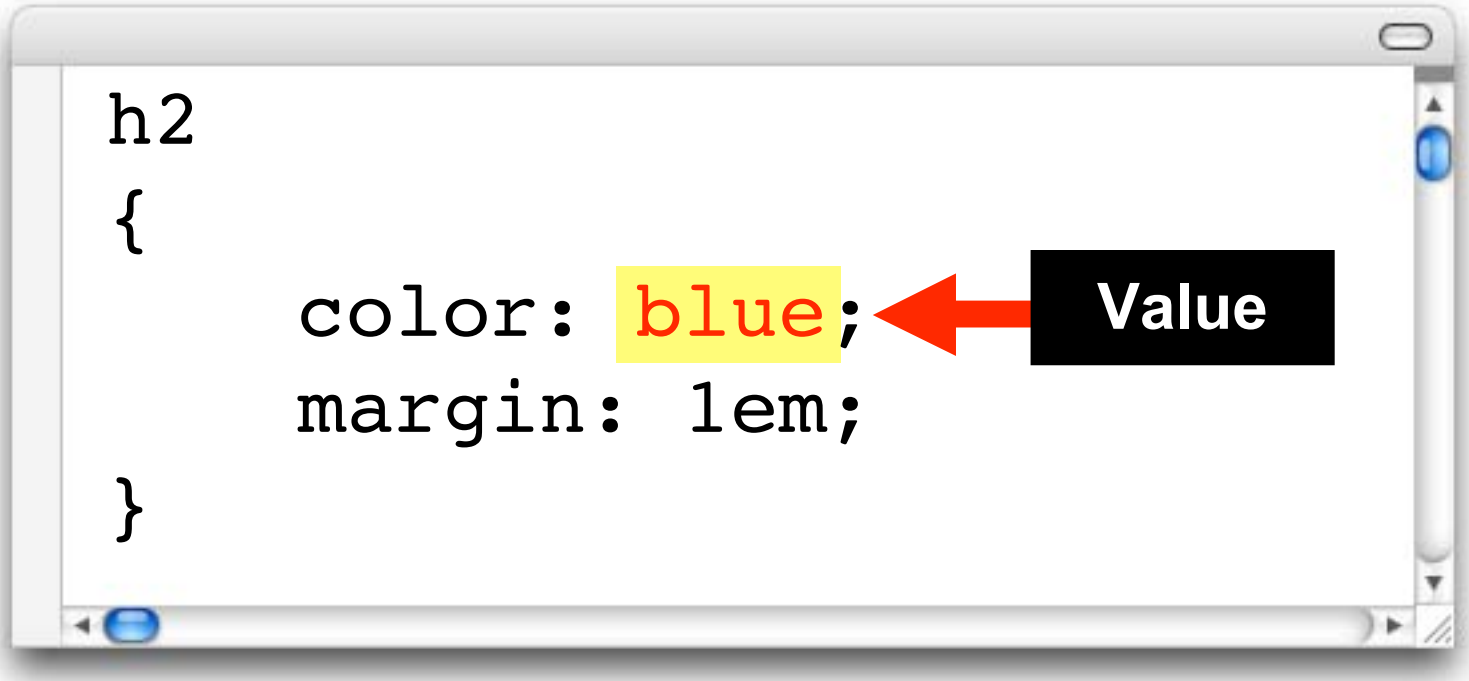
The **declaration** tells a browser how to style the element.



The **property** is the aspect of that element that you are choosing to style.



The **value** is the exact style you wish to set for the property.

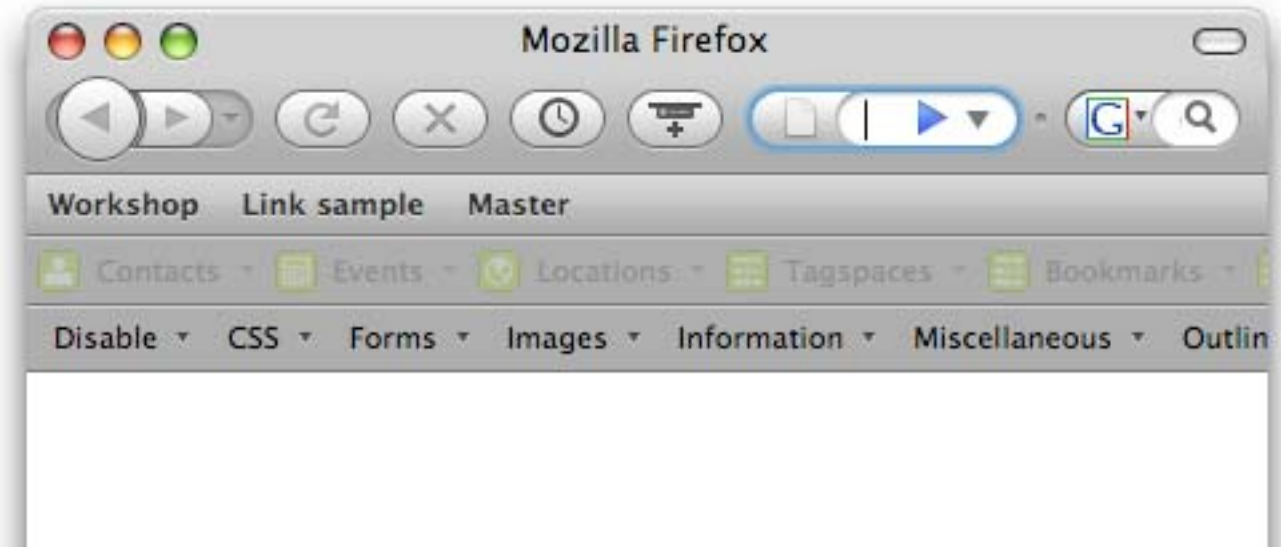


Types of style sheets

HTML documents may have **three types of style sheets** applied to them.

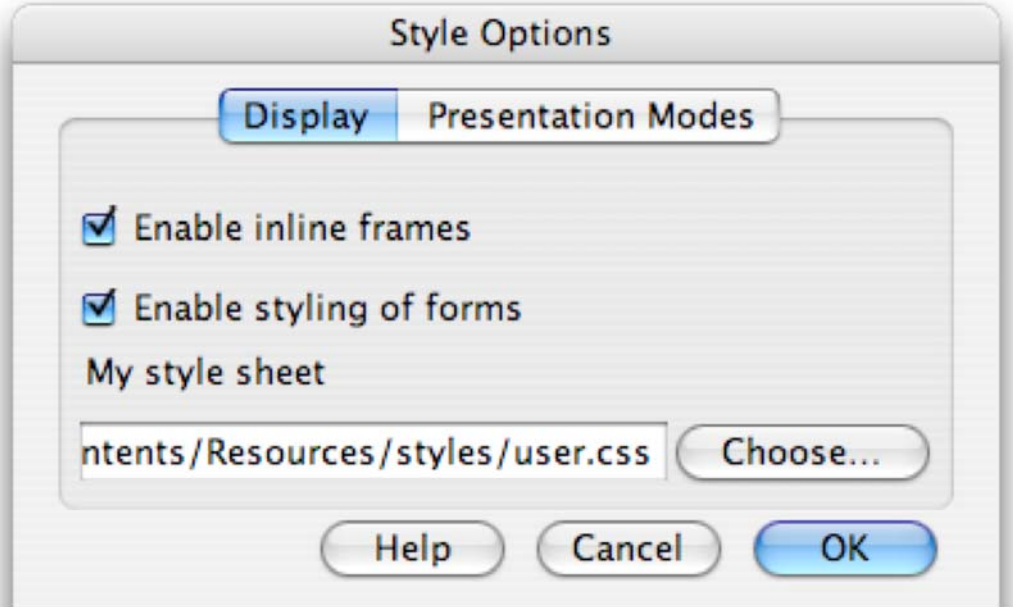
Browser style sheets

Browsers apply style sheets to all web documents. These are referred to as a "default" browser style sheet.



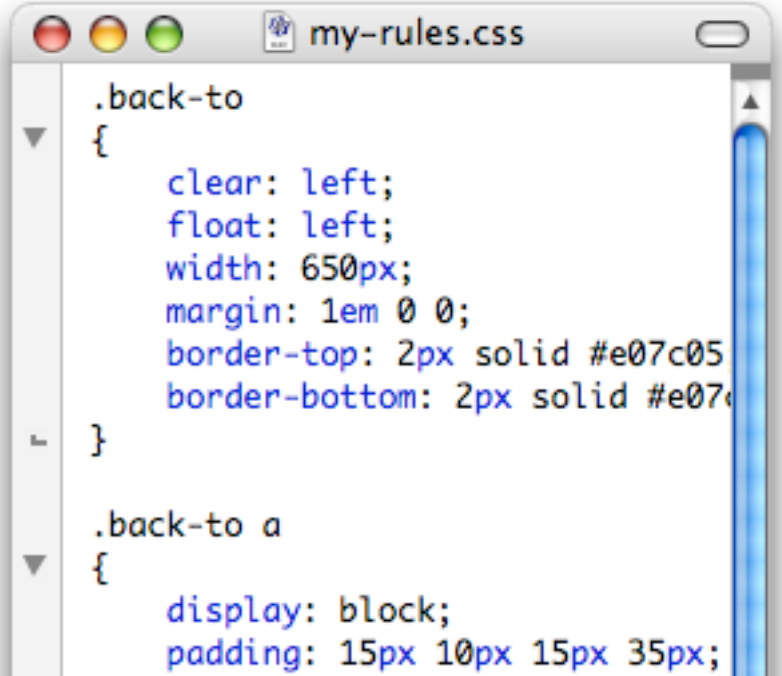
User style sheets

Most modern browsers allow users to apply their own style sheets within the browser.



Author style sheets

Web authors can apply one or more style sheets to an HTML document.

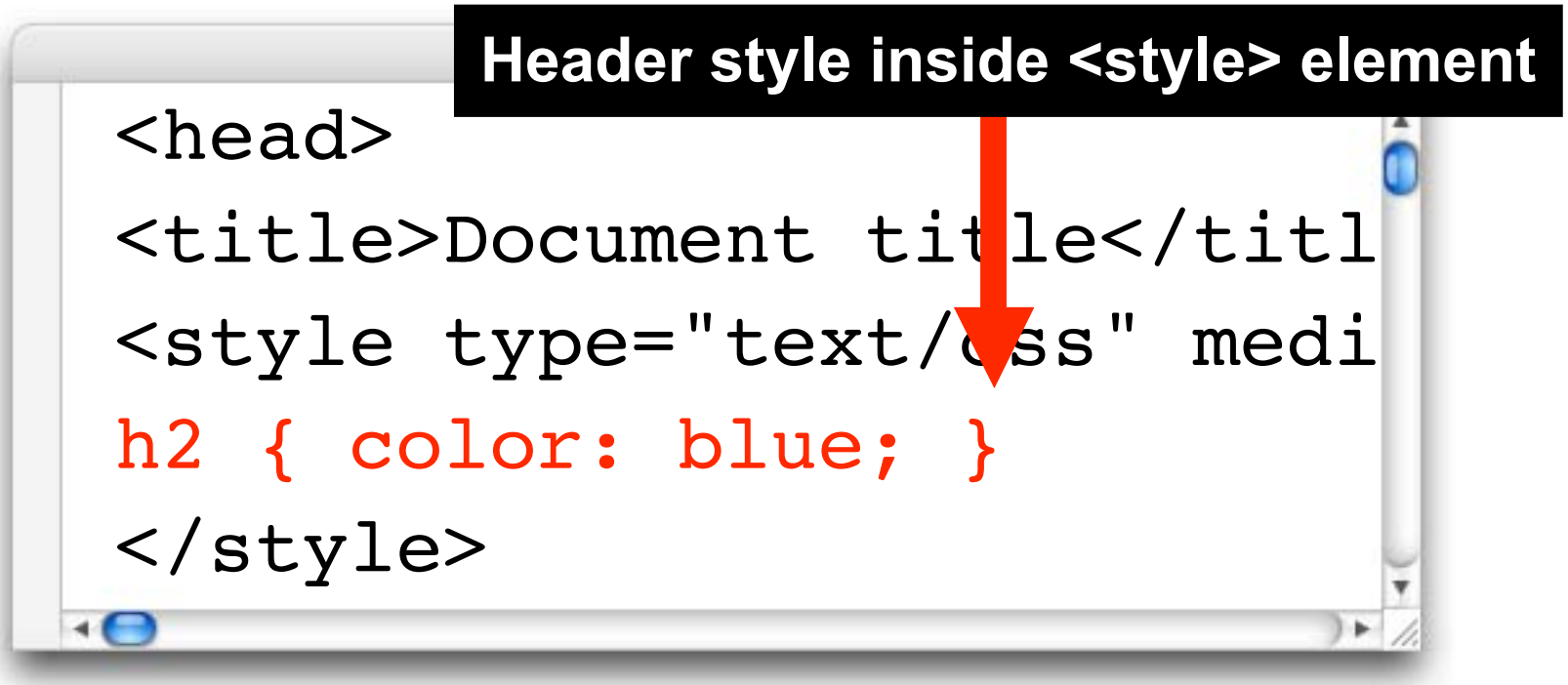


Author styles

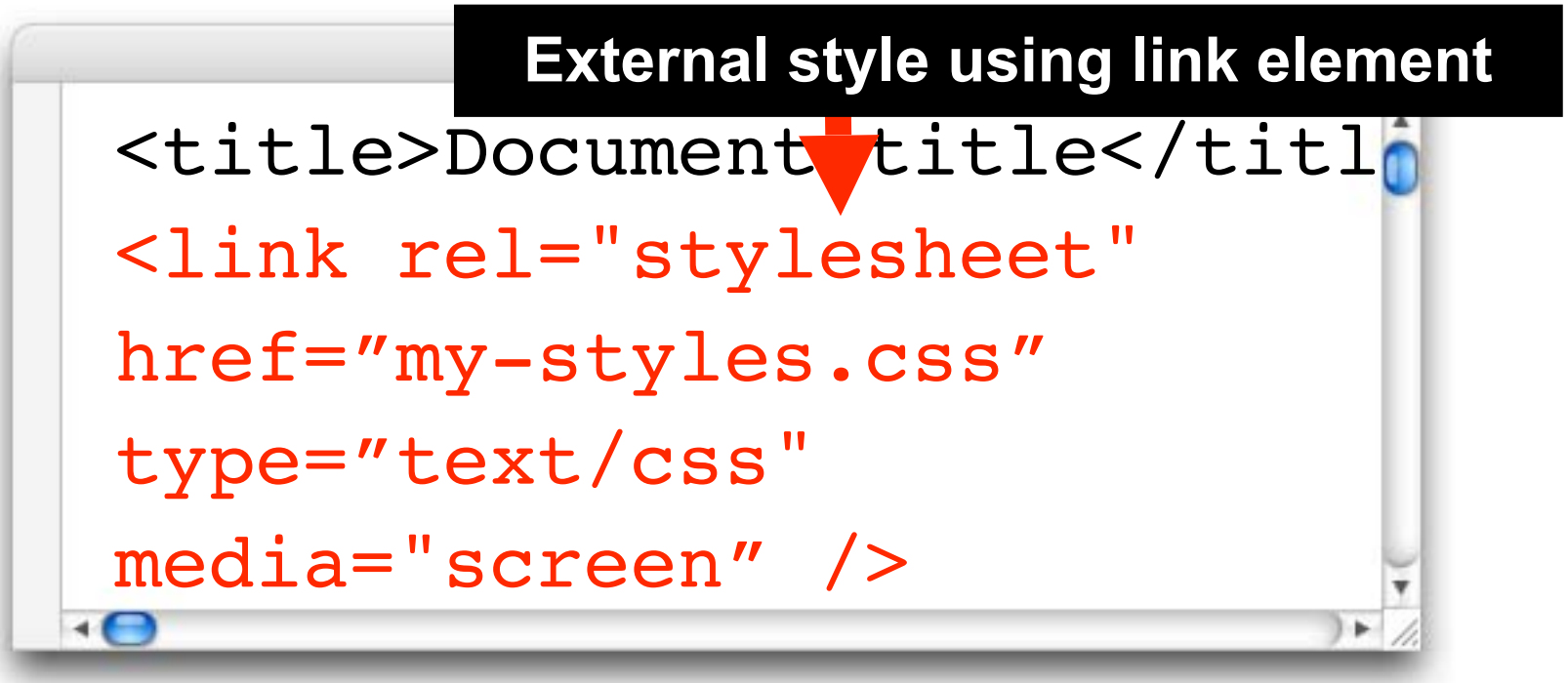
There are **three methods** that authors can use to add CSS styles to an HTML document

Inline styles are applied to elements in the HTML code using the style attribute.

Header styles are placed in the head of the document using the style element



External style sheets are applied using the link or @import.



CSS rule overload!

Browsers have to deal with CSS rules coming from the **browser, user and author style sheets**.

Browsers also have to deal with CSS rules coming from different types of **author style sheets** (external, header and inline)

At some point, Browsers have to deal with CSS rules that **conflict**.

What does “conflict” mean?

Conflict is where more than one CSS rule refers to the same **element** and **property**.

Conflict can occur between CSS rules in **different types of style sheets**.

Conflict can occur between CSS rules in within the **one or more author style sheets**.

So which CSS rules “win”?

There are **four steps**
to determine which CSS rules
will “win” (be applied to an
HTML document)

Step 1

Gather all the **declarations** that apply to an **element and property** from browser, author and user style sheets

For example, find any **declarations** that matches:

element = `h2`

property = `color`

Gathered declarations

Browser style sheet

```
h2 { color: black; }
```

User style sheet

```
h2 { color: green; }
```

Author style sheets

```
h2 { color: blue; }
#nav h2 { color: lime; }
```

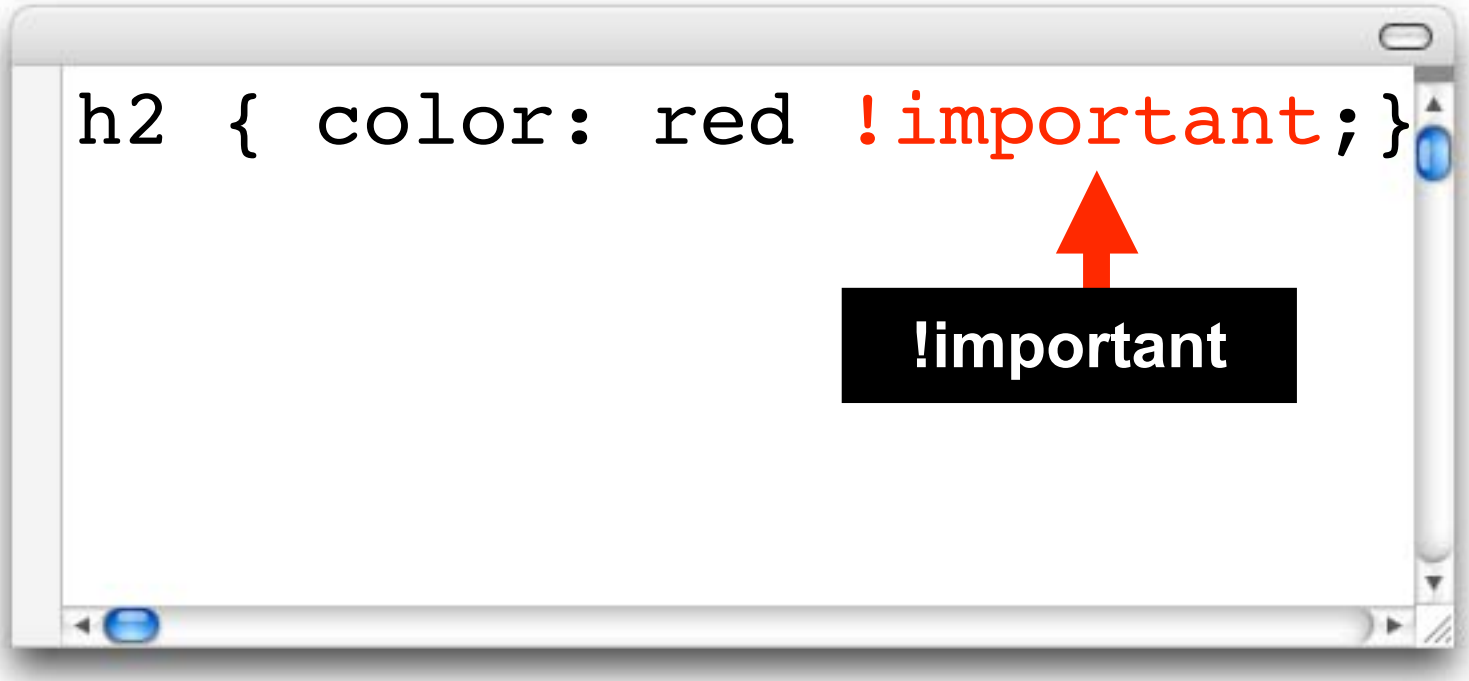
If there are declarations from **more than one of these three sources**, proceed to step 2.

Step 2

Sort the gathered declarations according to **origin** (browser, author, user style sheets) and **importance** (normal or !important).

What is !important?

Authors can assign "**!important**" to any declaration.



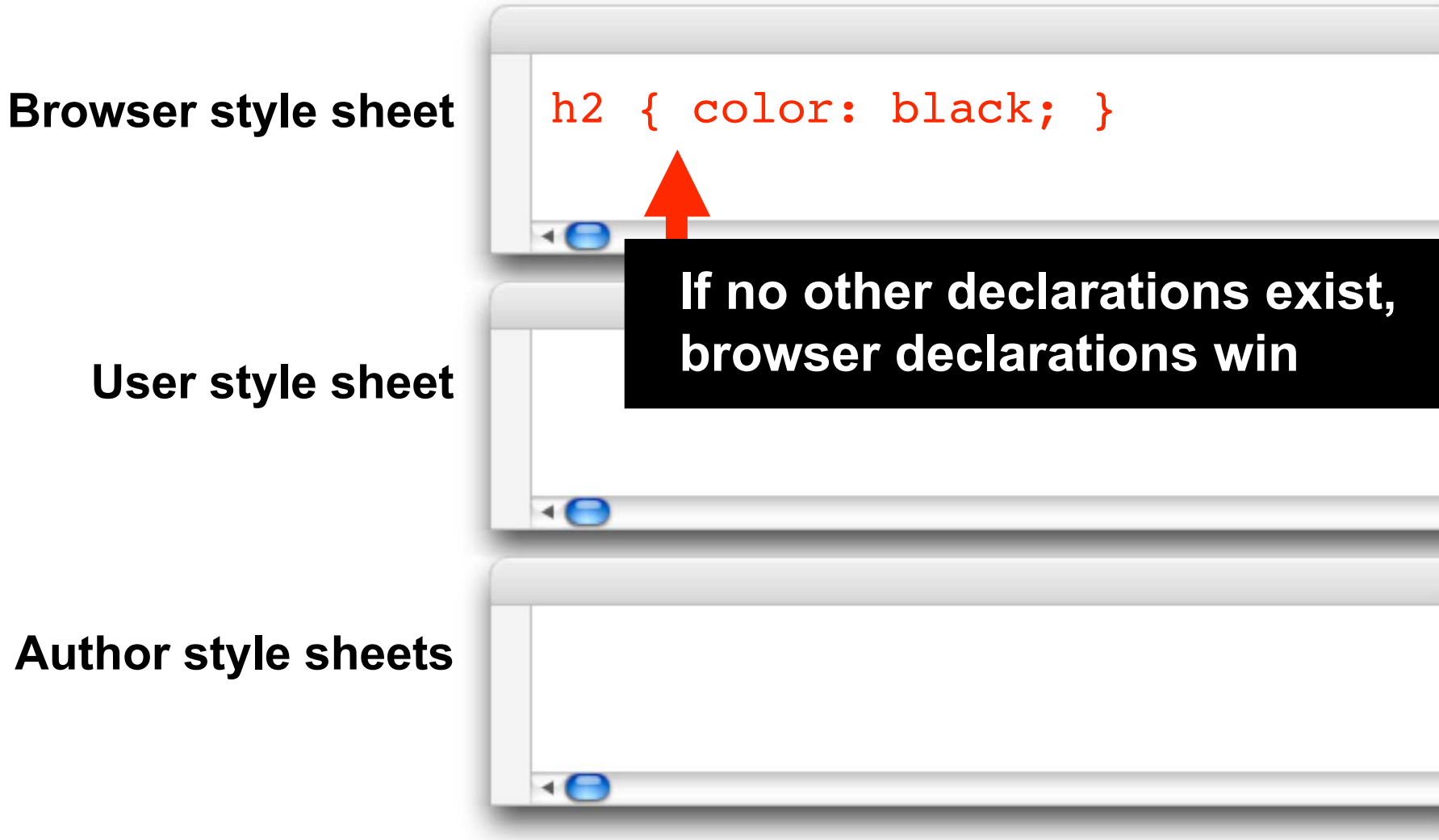
"!important" declarations **override normal declarations** (Normal declarations are declarations that do not contain !important).

So, how are declarations sorted?

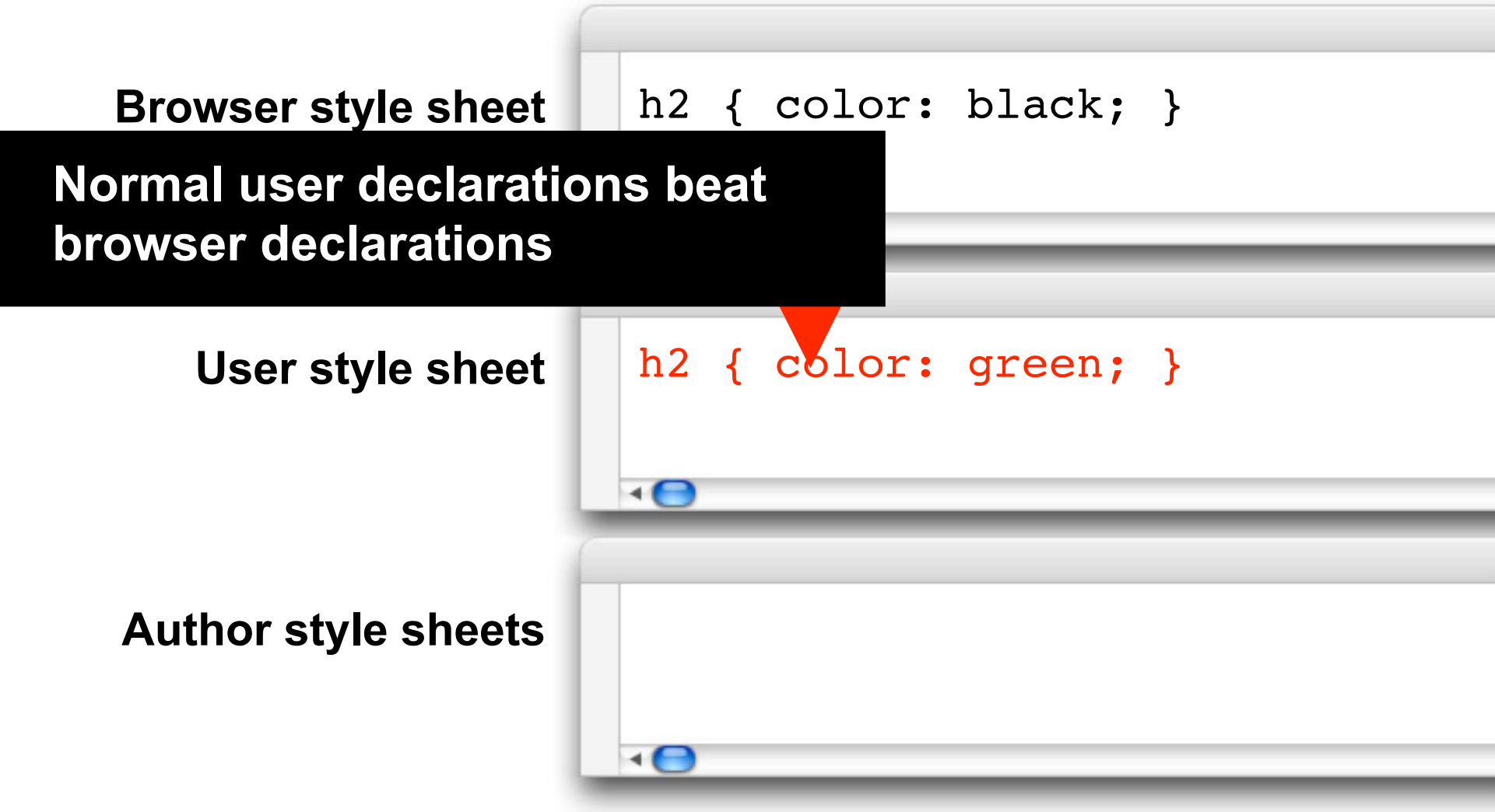
From lowest to highest **priority**

1. browser styles
2. normal declarations in user style sheet
3. normal declarations in author style sheet
4. !important declarations in author style sheet
5. !important declarations in user style sheet

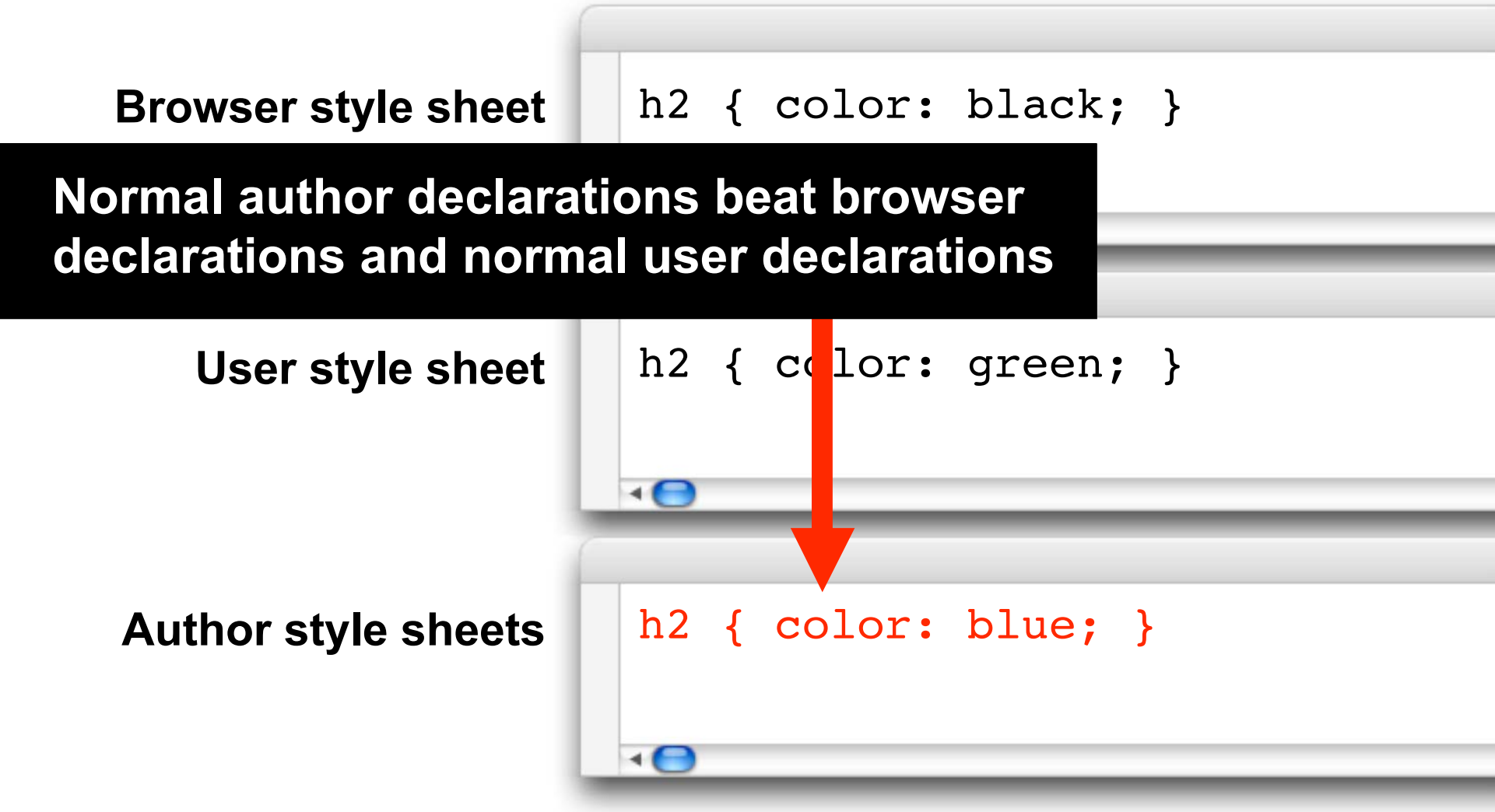
1. Browser styles



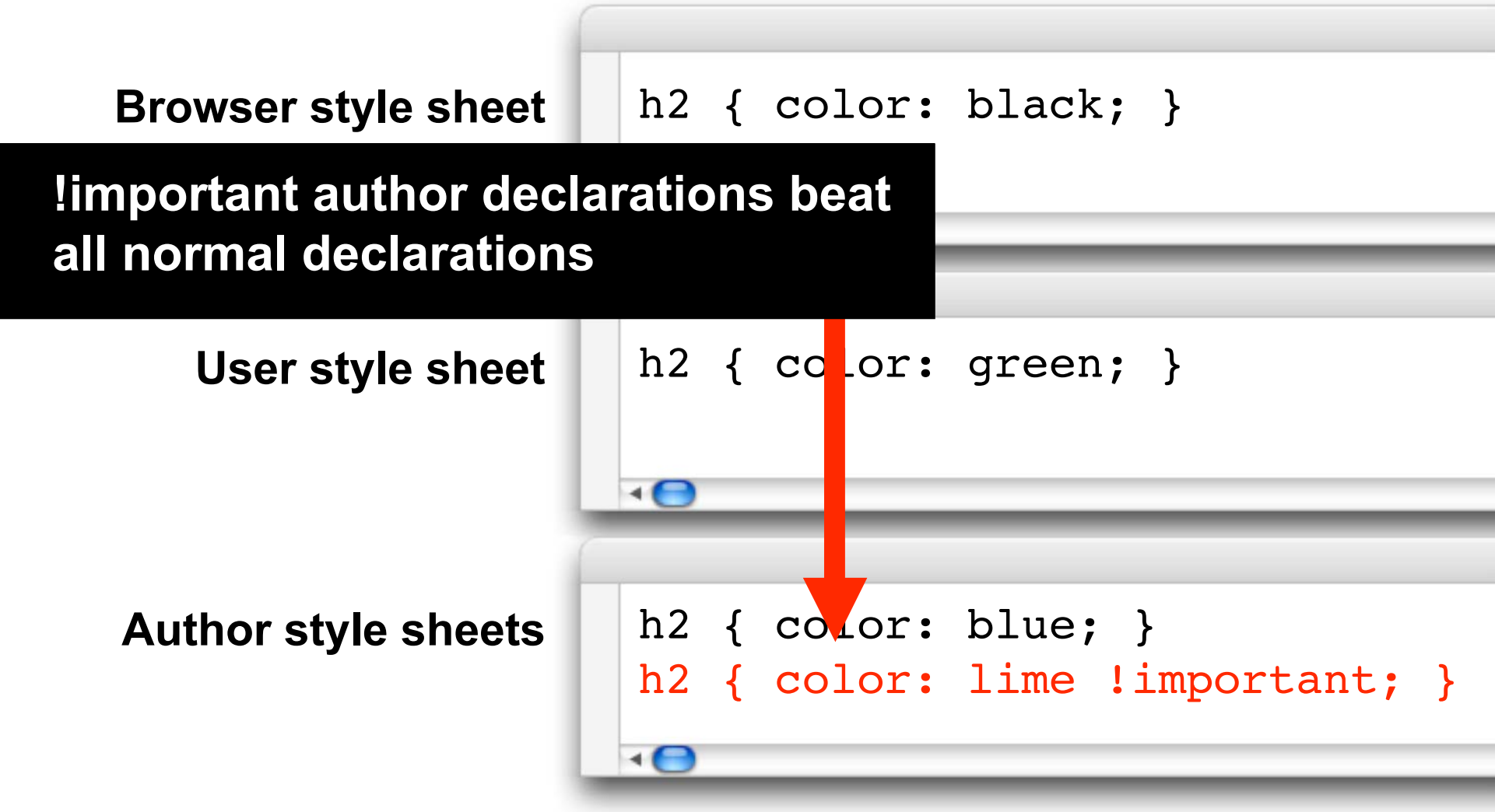
2. Normal user styles



3. Normal author styles



4. !important author styles



5. !important user styles

Browser style sheet

```
h2 { color: black; }
```

!important user declarations beat !important author declarations and all normal declarations

User style sheet

```
h2 { color: green; }
h2 { color: red !important;}
```

Author style sheets

```
h2 { color: blue; }
h2 { color: lime !important; }
```

But what if two declarations have **the same origin or importance**?

Two matching declarations

Browser style sheet

```
h2 { color: black; }
```

User style sheet

```
h2 { color: green; }
```

Two declarations with the same origin and importance

Author style sheets

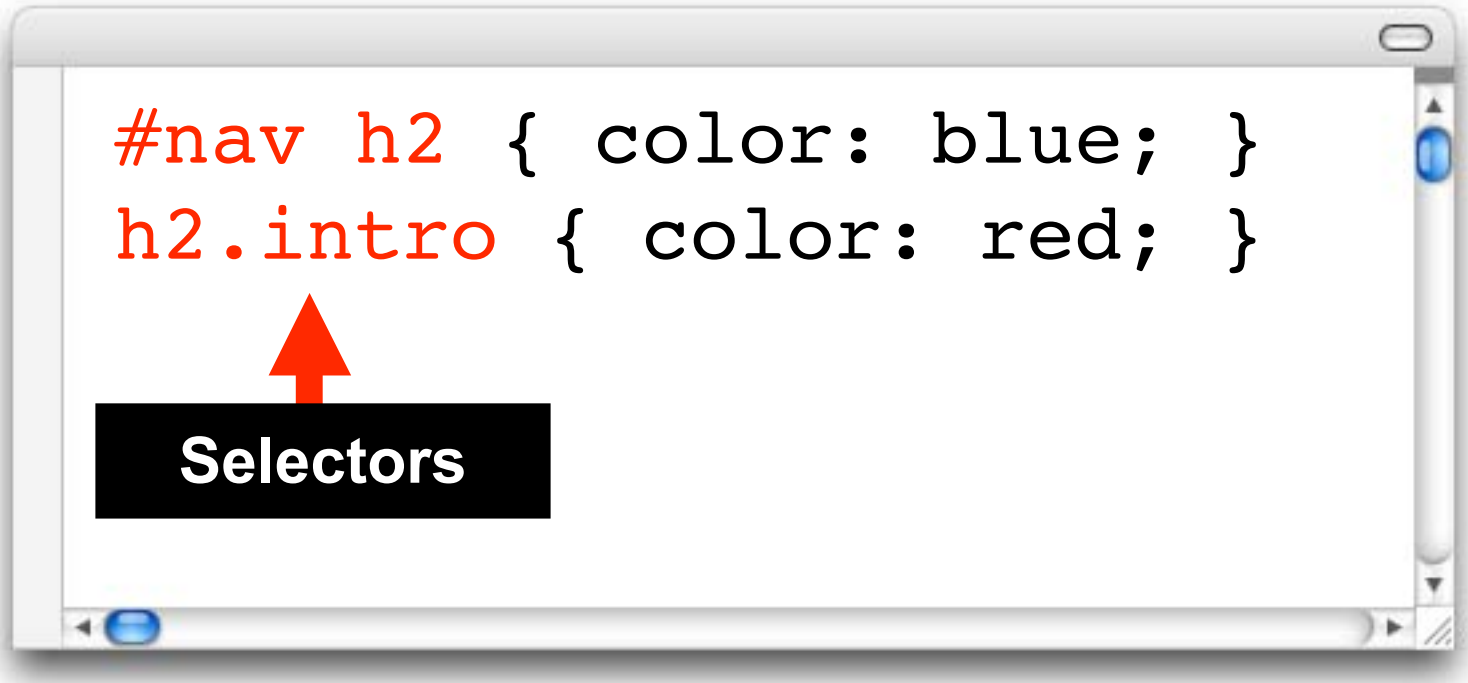
```
h2 { color: blue; }
h2 { color: lime; }
```

If declarations have **the same origin or importance** then proceed to Step 3.

Step 3

If declarations have the same origin or importance then the **declaration's selectors** need to be scored, to see which declaration will "win".

Selectors



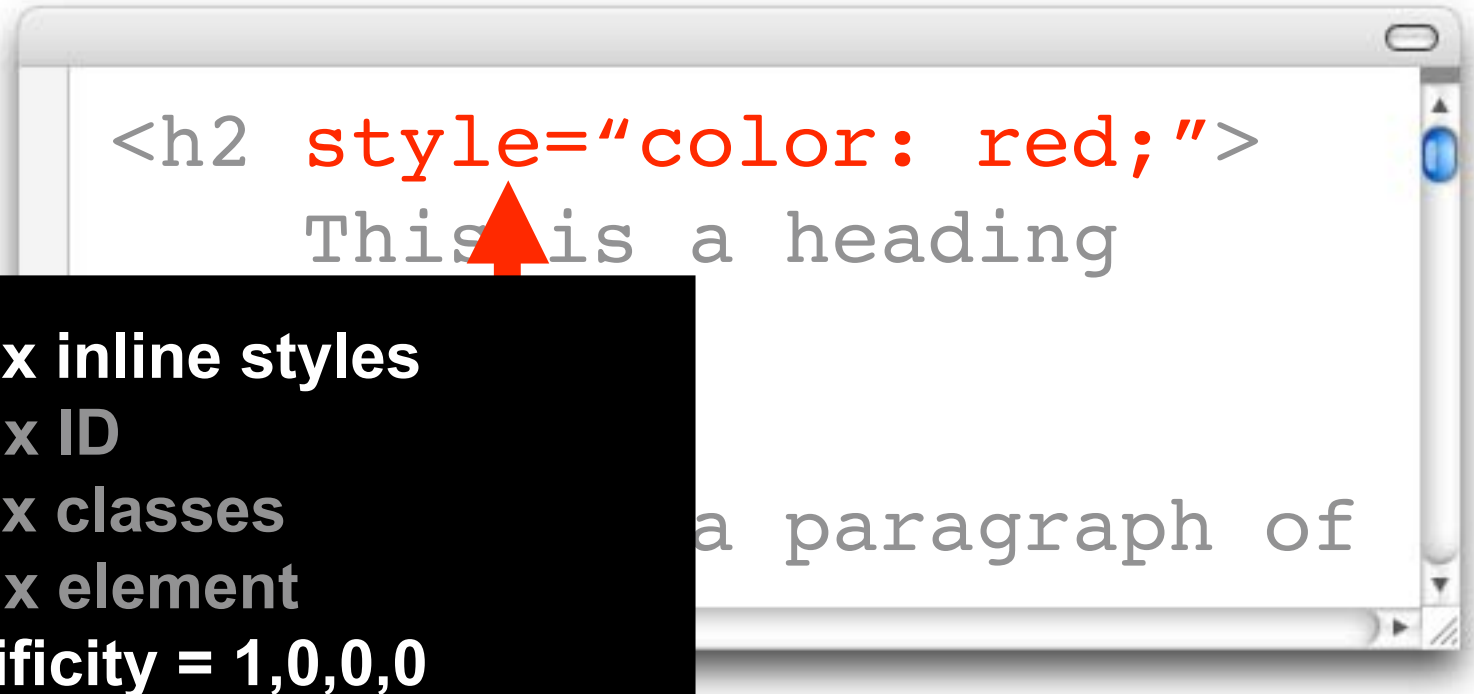
Four scores are **concatenated** (linked together as a chain) to create a final score.

a,b,c,d

This score is referred to as a selector's **specificity**.

So how is specificity calculated?

A. Is there an inline style?

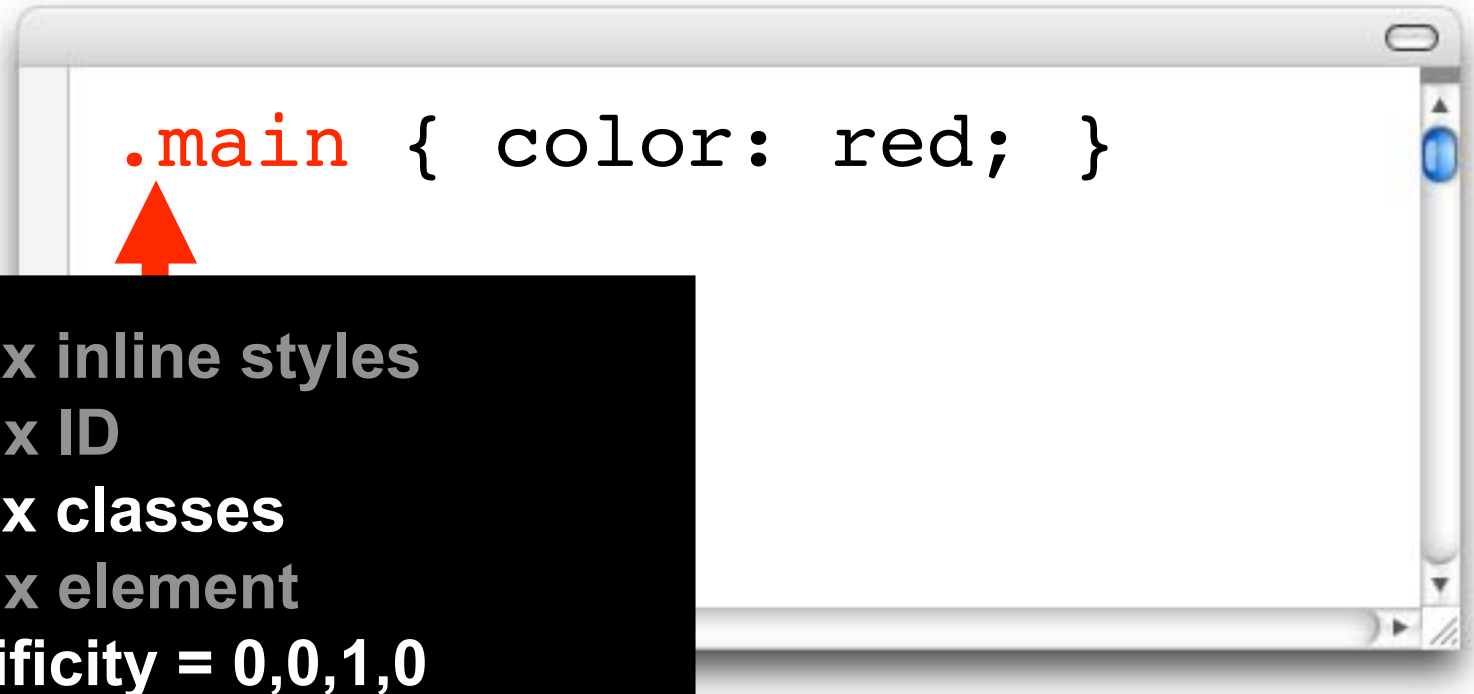


a = 1 x inline styles
b = 0 x ID
c = 0 x classes
d = 0 x element
Specificity = 1,0,0,0

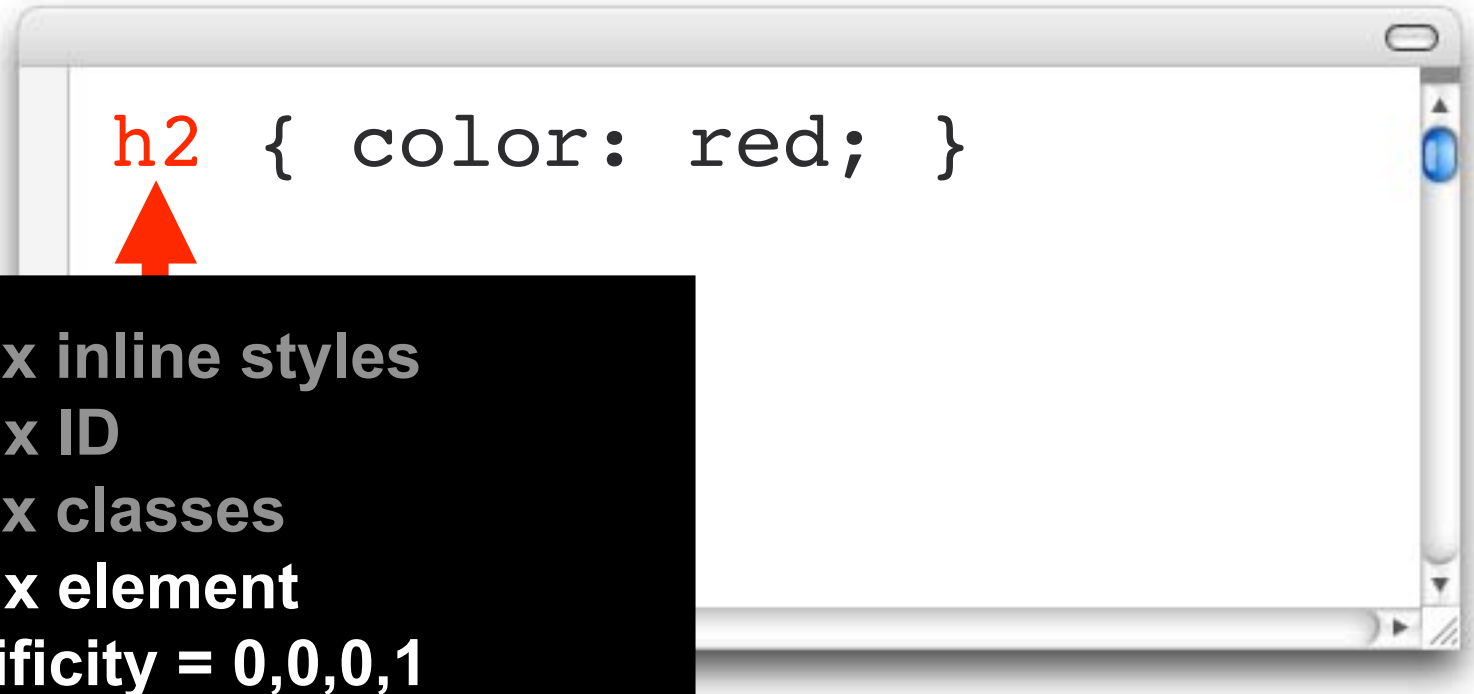
B. Count the number of IDs in the selectors.

a = 0 x inline styles
b = 1 x ID
c = 0 x classes
d = 0 x element
Specificity = 0,1,0,0

C. Count the number of classes, attributes and pseudo-classes.



D. Count the number of element names or pseudo-elements.



a = 0 x inline styles
b = 0 x ID
c = 0 x classes
d = 1 x element
Specificity = 0,0,0,1

A note on concatenation

“**A**” will always beat “**B**”, which will always beat “**C**”, which will always beat “**D**”.

No matter how many **IDs** are used in a selector, an **inline style** will always win.

(unless !important is used within the ID's declaration)

External style sheets and header styles (Author styles)

```
#one #two #three #four #five
#six #seven #eight #nine #ten
{ color: green; }
```

HTML document with inline styles (Author styles)

```
<h2 style="color: purple;">
```

The highlighted style wins due to specificity - 1,0,0,0 beats 0,10,0,0

No matter how many **classes** are applied to a selector, an **ID** can easily win

External style sheets and header styles (Author styles)

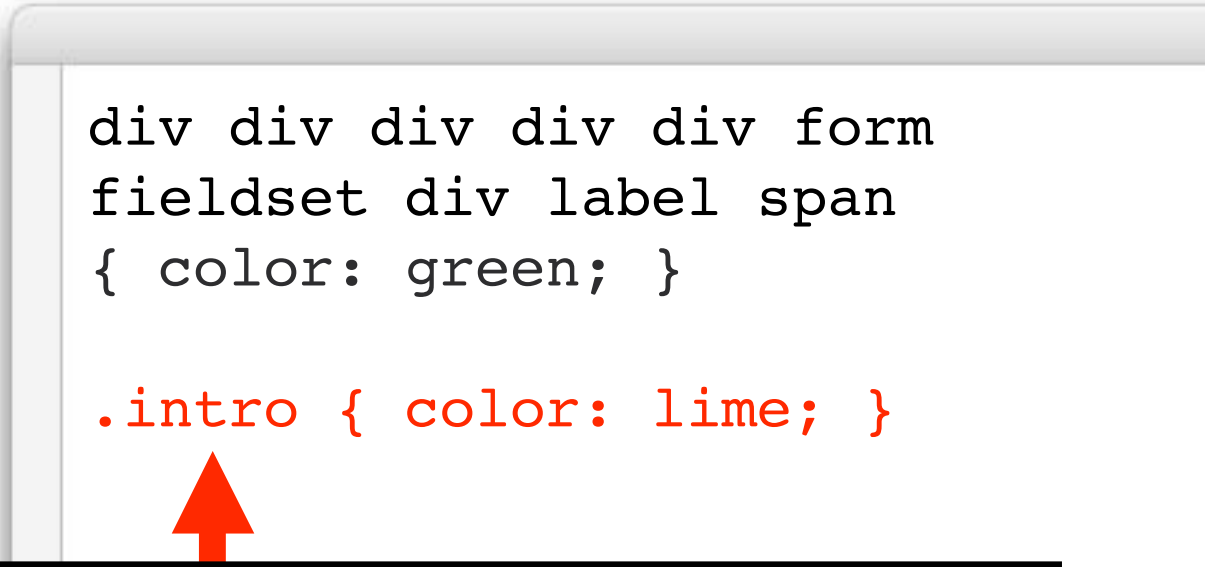
```
.one .two .three .four .five
.six .seven .eight .nine .ten
{ color: green; }

#nav { color: lime; }
```

The highlighted selector wins due to specificity - 0,1,0,0 beats 0,0,10,0

No matter how many **elements** are applied to a selector, a **class** can easily win.

External style sheets and header styles (Author styles)



The highlighted selector wins due to specificity - 0,0,1,0 beats 0,0,0,10

Complex examples of specificity

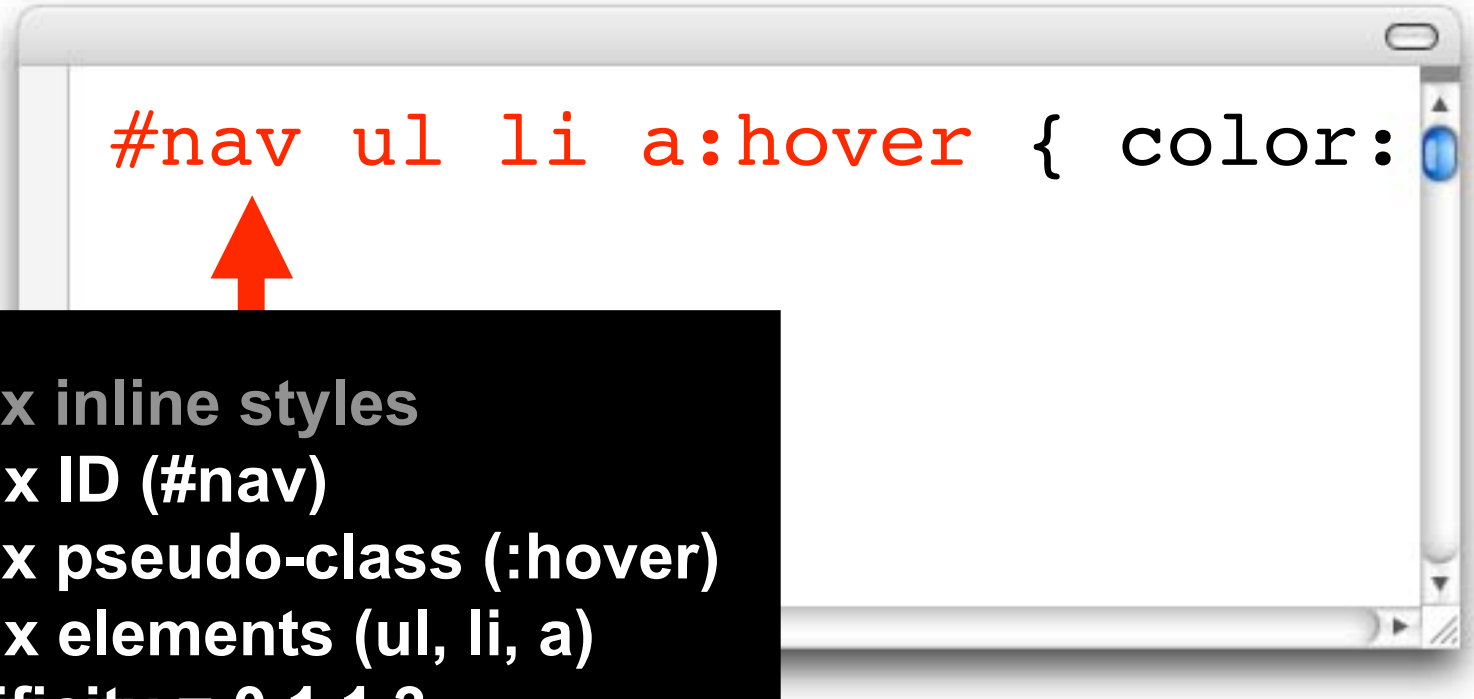
ID and element

a = 0 x inline styles
b = 1 x ID (#nav)
c = 0 x classes
d = 1 x element (h2)
Specificity = 0,1,0,1

Element and class

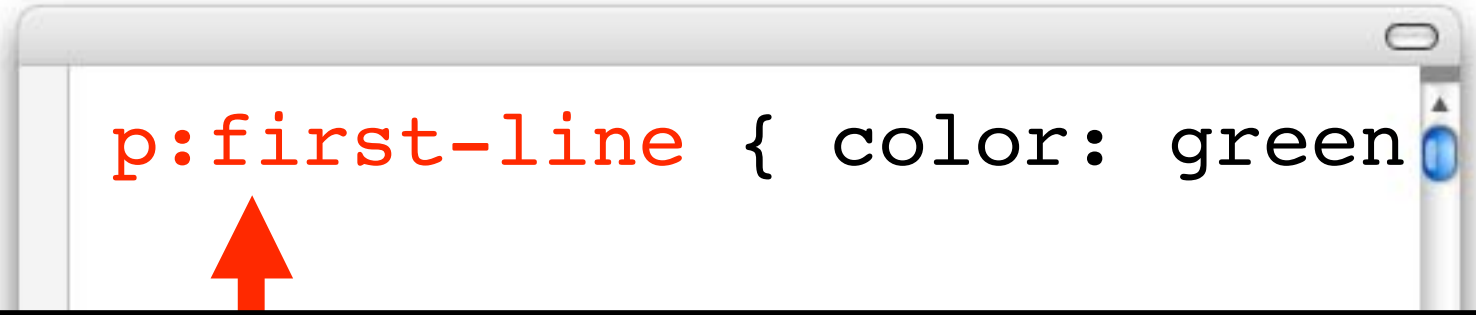
a = 0 x inline styles
b = 0 x ID
c = 1 x classes (.intro)
d = 1 x element (h2)
Specificity = 0,0,1,1

ID, elements and pseudo-class



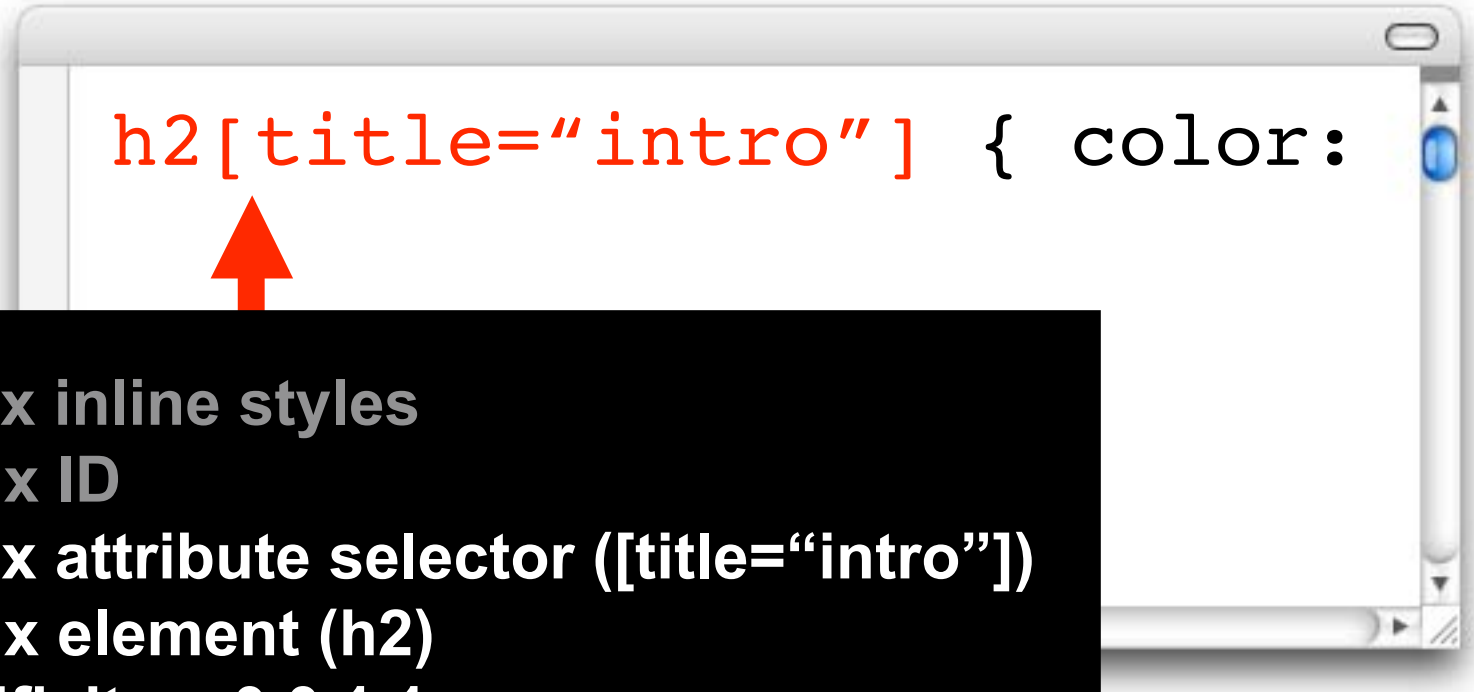
a = 0 x inline styles
b = 1 x ID (#nav)
c = 1 x pseudo-class (:hover)
d = 3 x elements (ul, li, a)
Specificity = 0,1,1,3

Element and pseudo-element



a = 0 x inline styles
b = 0 x ID
c = 0 x classes
d = 2 x element (p) and pseudo-element (:first-line)
Specificity = 0,0,0,2

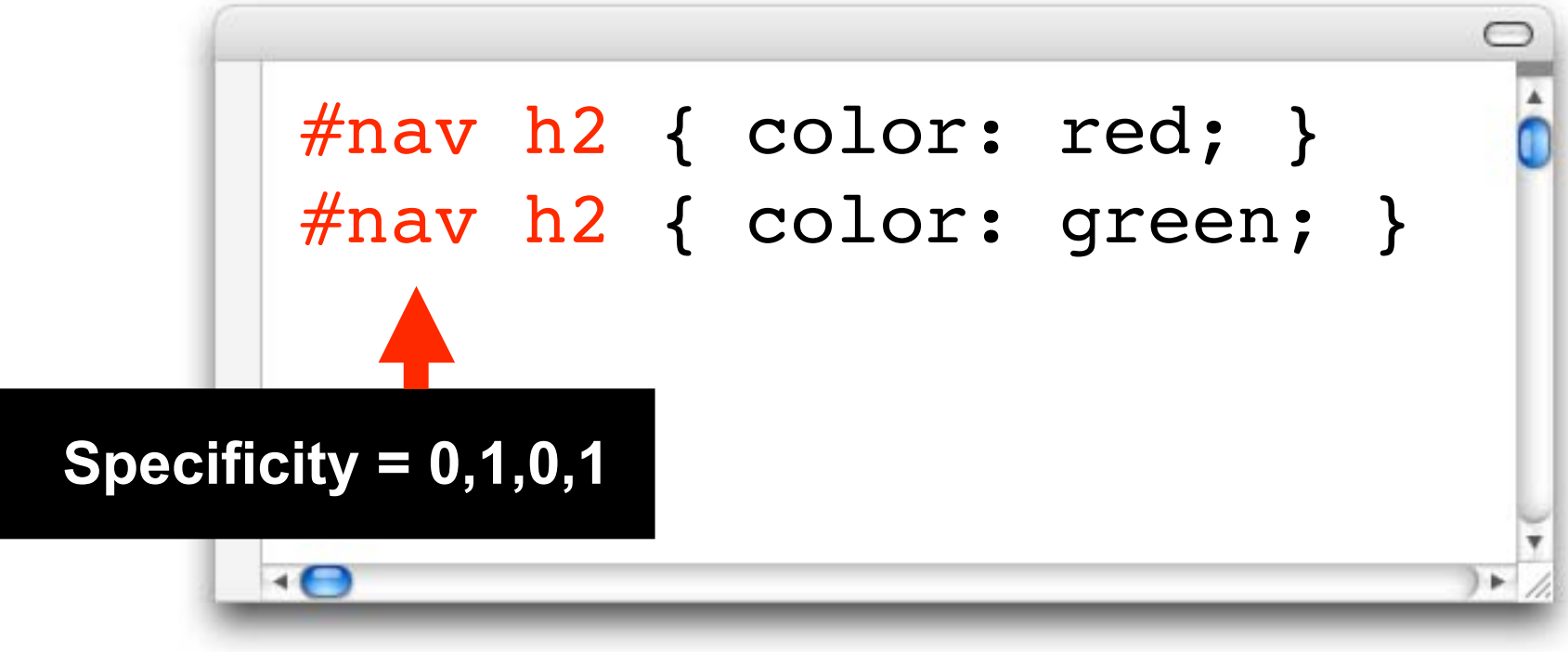
Element and attribute selector



a = 0 x inline styles
b = 0 x ID
c = 1 x attribute selector ([title="intro"])
d = 1 x element (h2)
Specificity = 0,0,1,1

What if there is still no clear winner?

Selectors with same specificity

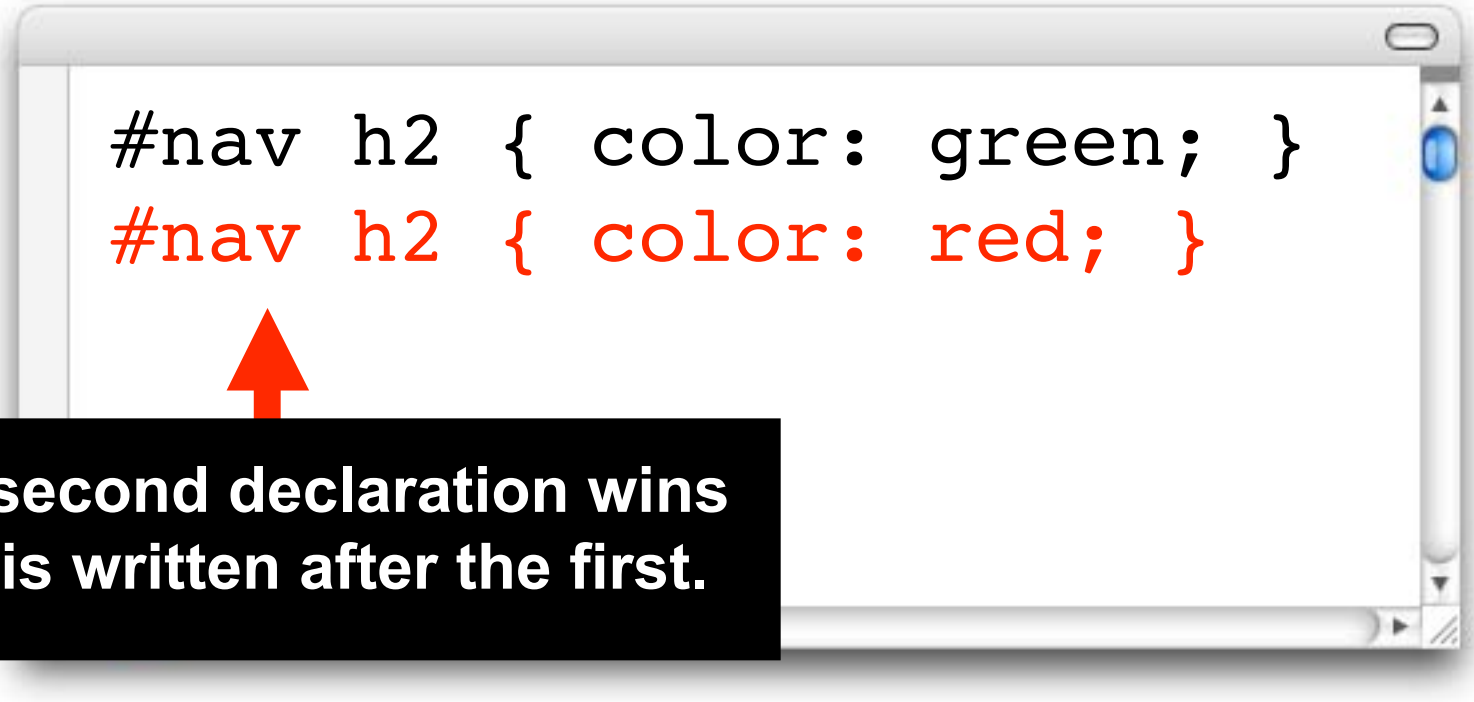


If there is still **no clear winner** then proceed to Step 4.

Step 4

If two declarations have the same importance, origin and specificity, the **latter specified** declaration wins

Equal-weight declarations



The second declaration wins as it is written after the first.

And now... a guessing game

Exercise 1

browser, user, author

Part 1: Which one **wins?**

Browser style sheet

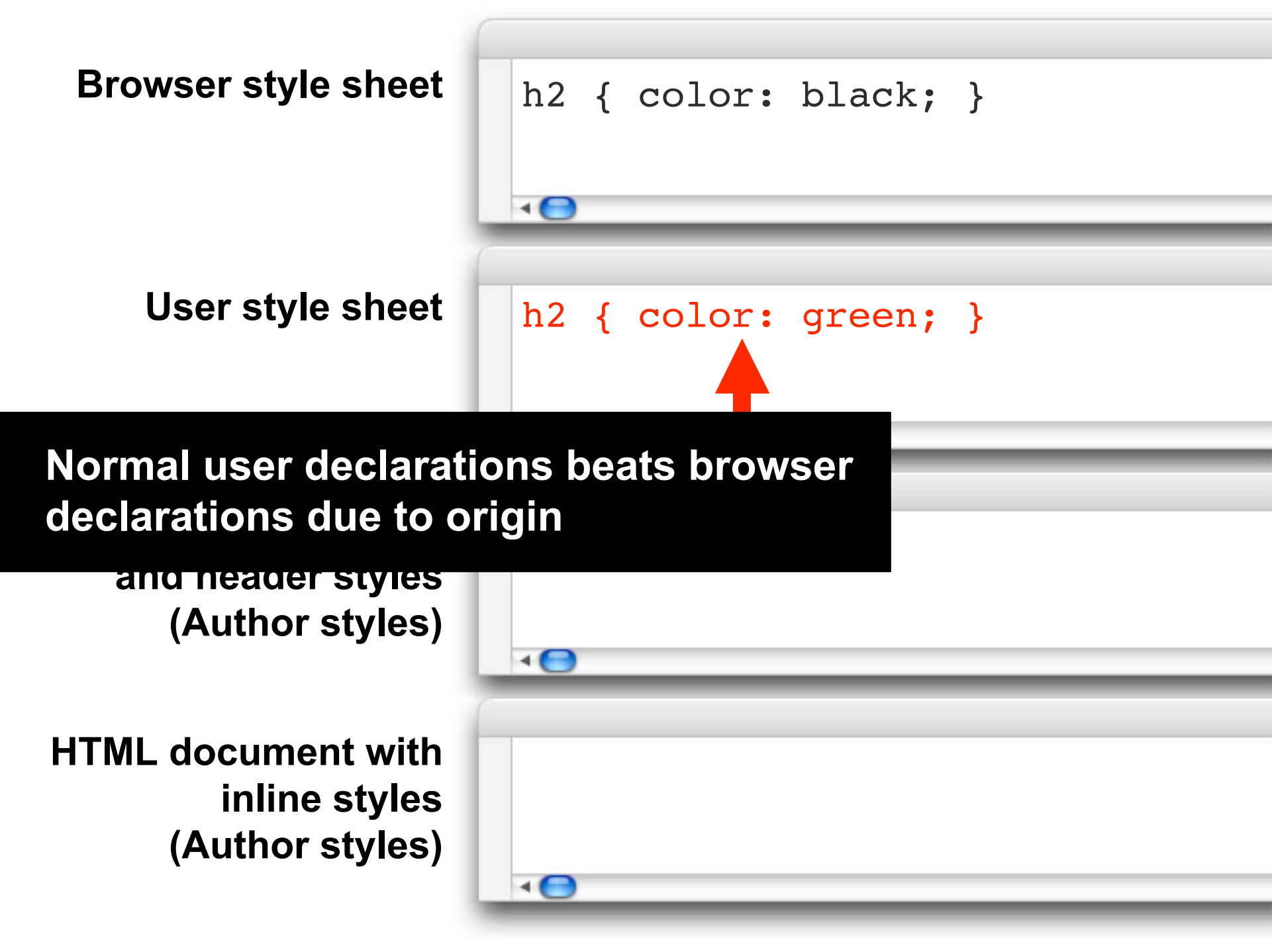
```
h2 { color: black; }
```

User style sheet

```
h2 { color: green; }
```

External style sheets and header styles (Author styles)

HTML document with inline styles (Author styles)


Browser style sheet
h2 { color: black; }
User style sheet
h2 { color: green; }
Normal user declarations beats browser declarations due to origin
and header styles
(Author styles)
HTML document with
inline styles
(Author styles)

Part 2: Which one **wins?**

Browser style sheet

```
h2 { color: black; }
```

User style sheet

```
h2 { color: green; }
```

External style sheets and header styles (Author styles)

```
h2 { color: blue; }
```

HTML document with inline styles (Author styles)

Browser style sheet

```
h2 { color: black; }
```

User style sheet

```
h2 { color: green; }
```

External style sheets
and header styles
(Author styles)

```
h2 { color: blue; }
```

Normal author declarations beat browser and normal user declarations due to origin

Inline styles
(Author styles)

Part 3: Which one **wins?**

Browser style sheet

```
h2 { color: black; }
```

User style sheet

```
h2 { color: green; }
```

External style sheets and header styles (Author styles)

```
h2 { color: blue; }
```

HTML document with inline styles (Author styles)

```
<h2 style="color: purple;">
```

Browser style sheet

```
h2 { color: black; }
```

Normal inline declarations beat normal external and header declarations due to specificity: 1,0,0,0 beats 0,0,0,1

External style sheets and header styles (Author styles)

```
h2 { color: blue; }
```

HTML document with inline styles (Author styles)

```
<h2 style="color: purple;">
```

Part 4: Which one **wins?**

Browser style sheet

```
h2 { color: black; }
```

User style sheet

```
h2 { color: green; }
```

External style sheets and header styles (Author styles)

```
h2 { color: blue; }
h2 { color: lime !important; }
```

HTML document with inline styles (Author styles)

```
<h2 style="color: purple;">
```

Browser style sheet

```
h2 { color: black; }
```

User style sheet

```
h2 { color: green; }
```

!important author declarations beat normal browser, user and author declarations

External style sheets and header styles (Author styles)

```
h2 { color: blue; }
h2 { color: lime !important; }
```

HTML document with inline styles (Author styles)

```
<h2 style="color: purple;">
```

Part 5: Which one **wins?**

Browser style sheet

```
h2 { color: black; }
```

User style sheet

```
h2 { color: green; }
```

External style sheets and header styles (Author styles)

```
h2 { color: blue; }
h2 { color: lime !important; }
```

HTML document with inline styles (Author styles)

```
<h2 style="color: purple
!important;">
```

Browser style sheet

```
h2 { color: black; }
```

!important inline author declarations beat !important external author and header declarations due to specificity: 1,0,0,0 beats 0,0,0,1

External style sheets and header styles (Author styles)

```
h2 { color: blue; }
h2 { color: lime !important; }
```

HTML document with inline styles (Author styles)

```
<h2 style="color: purple
!important;">
```

Part 6: Which one **wins?**

Browser style sheet

```
h2 { color: black; }
```

User style sheet

```
h2 { color: green; }
h2 { color: gray !important; }
```

External style sheets and header styles (Author styles)

```
h2 { color: blue; }
h2 { color: lime !important; }
```

HTML document with inline styles (Author styles)

```
<h2 style="color: purple
!important;">
```

!important user declarations beat !important author declarations (regardless of whether they are external, header or inline)

User style sheet

```
h2 { color: green; }
h2 { color: gray !important; }
```

External style sheets and header styles (Author styles)

```
h2 { color: blue; }
h2 { color: lime !important; }
```

HTML document with inline styles (Author styles)

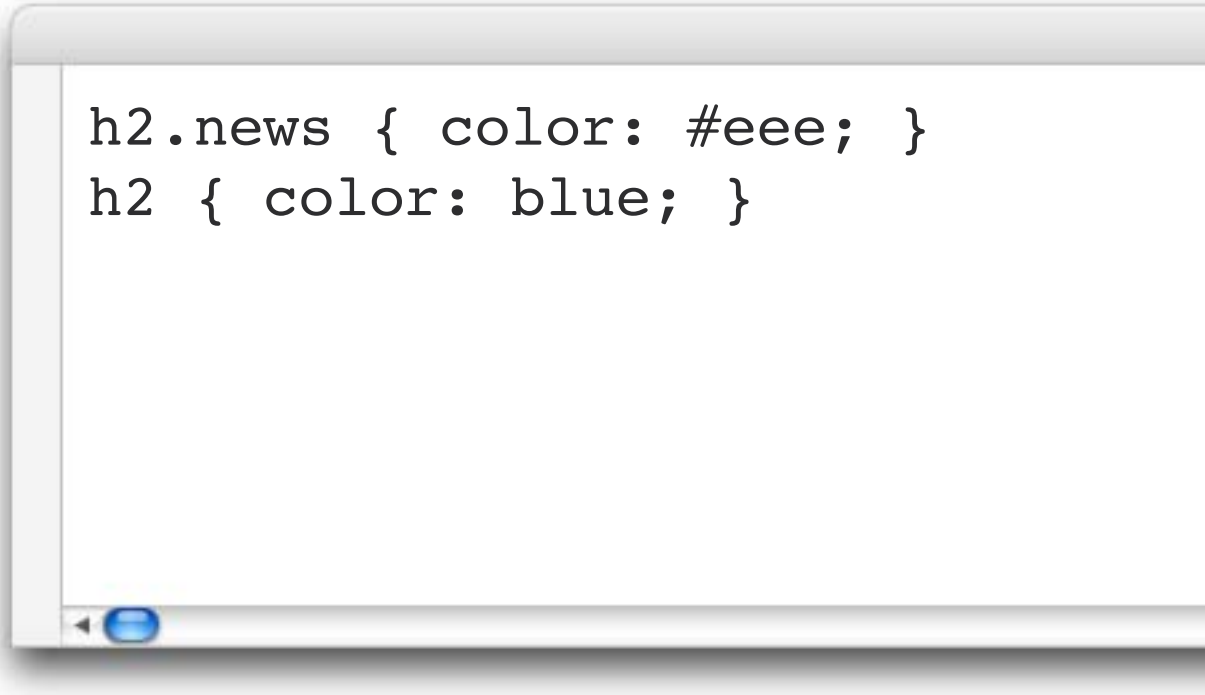
```
<h2 style="color: purple
!important;">
```

Exercise 2

author external, header and inline CSS

Part 1: Which one **wins?**

External style sheets and header styles (Author styles)

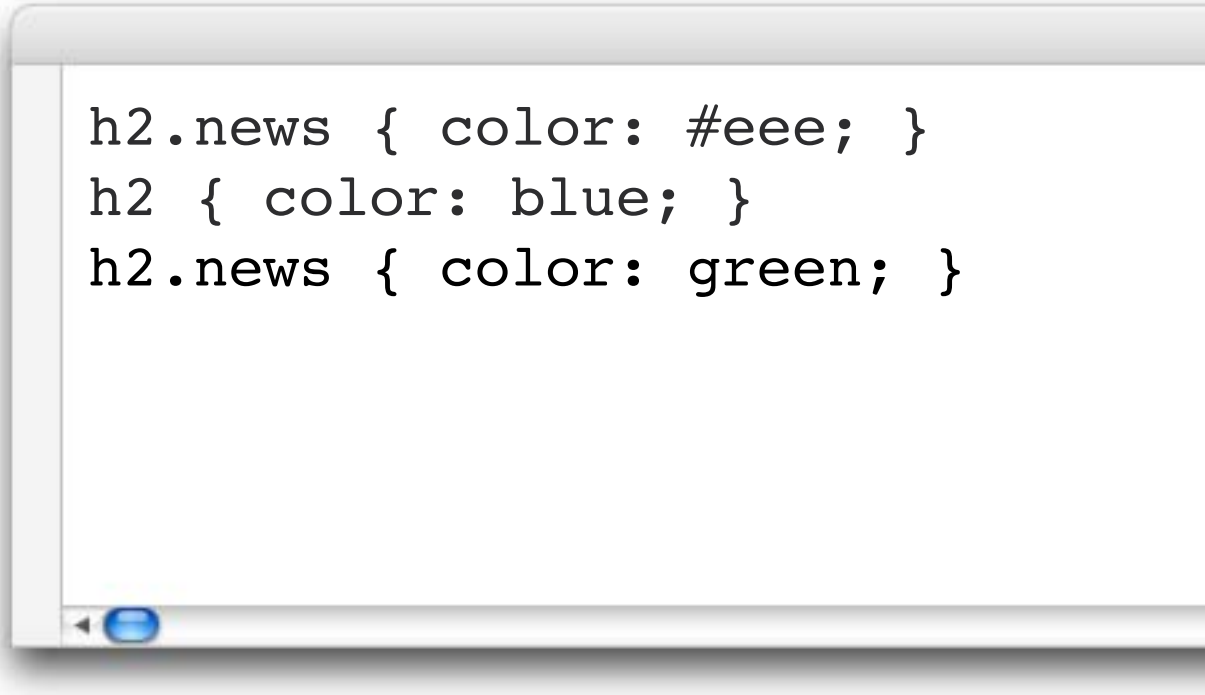


The highlighted declaration wins due to specificity - 0,0,1,1 beats 0,0,0,1

External style sheets and header styles (Author styles)

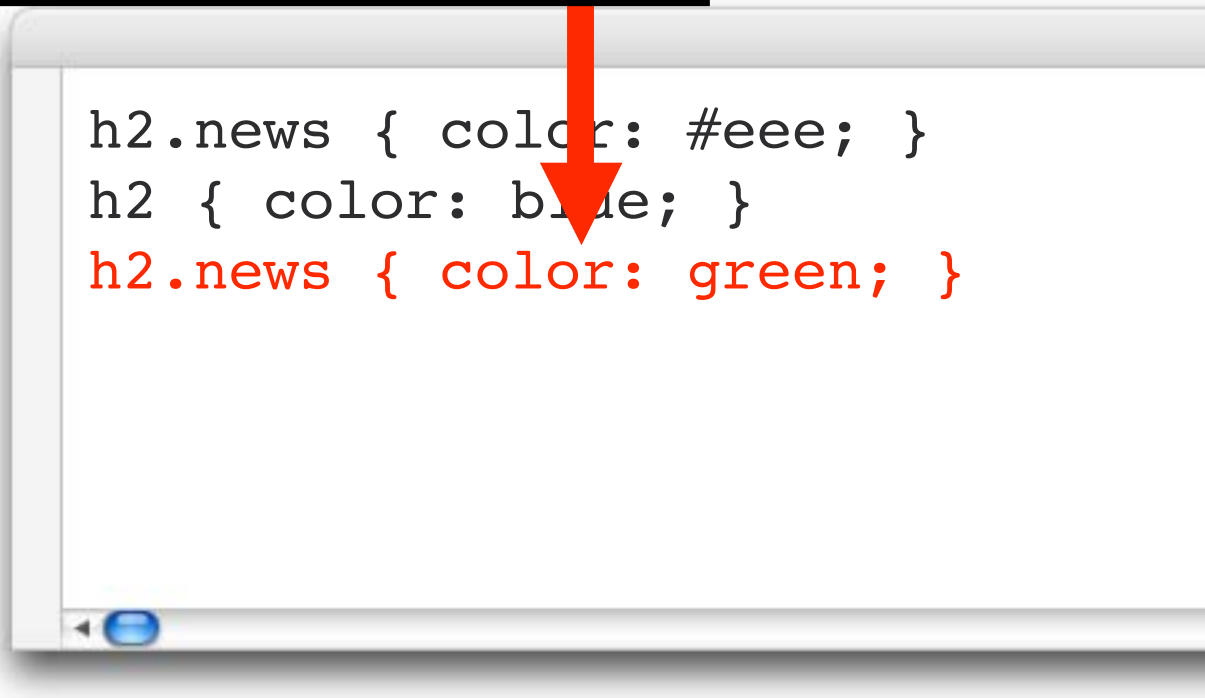
Part 2: Which one **wins?**

External style sheets and header styles (Author styles)



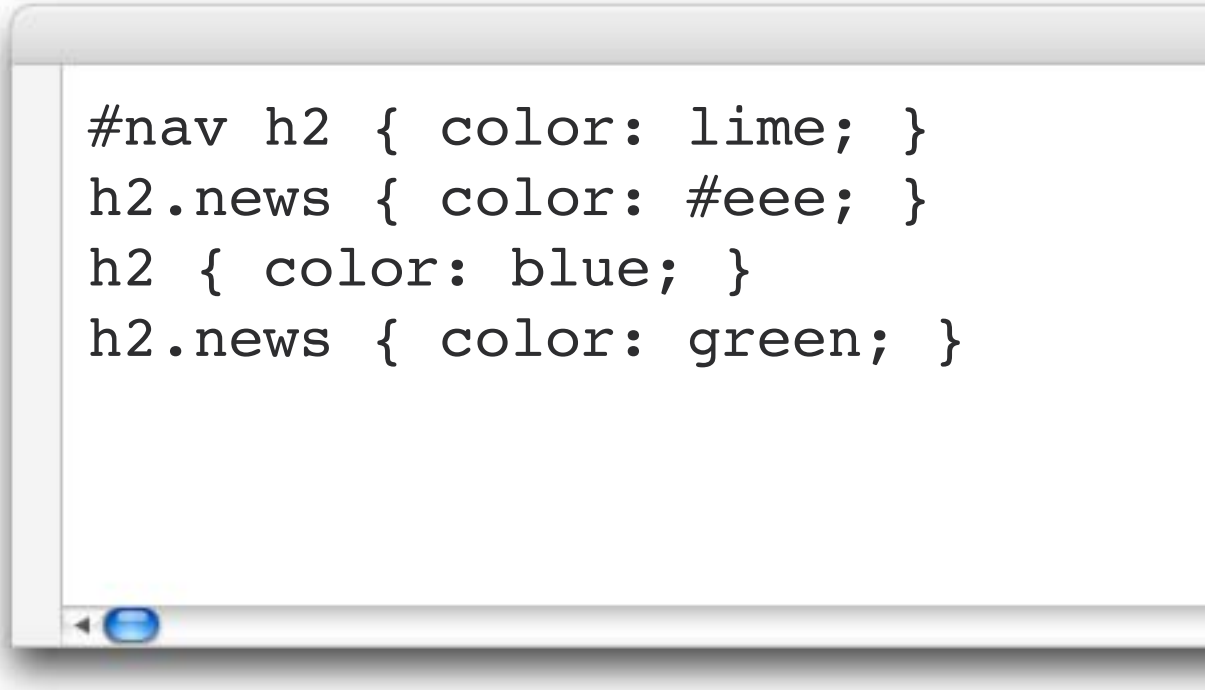
The highlighted declaration has the same specificity as the first declaration (0,0,1,1). However, as it is written later, it wins!

External style sheets and header styles (Author styles)



Part 3: Which one **wins?**

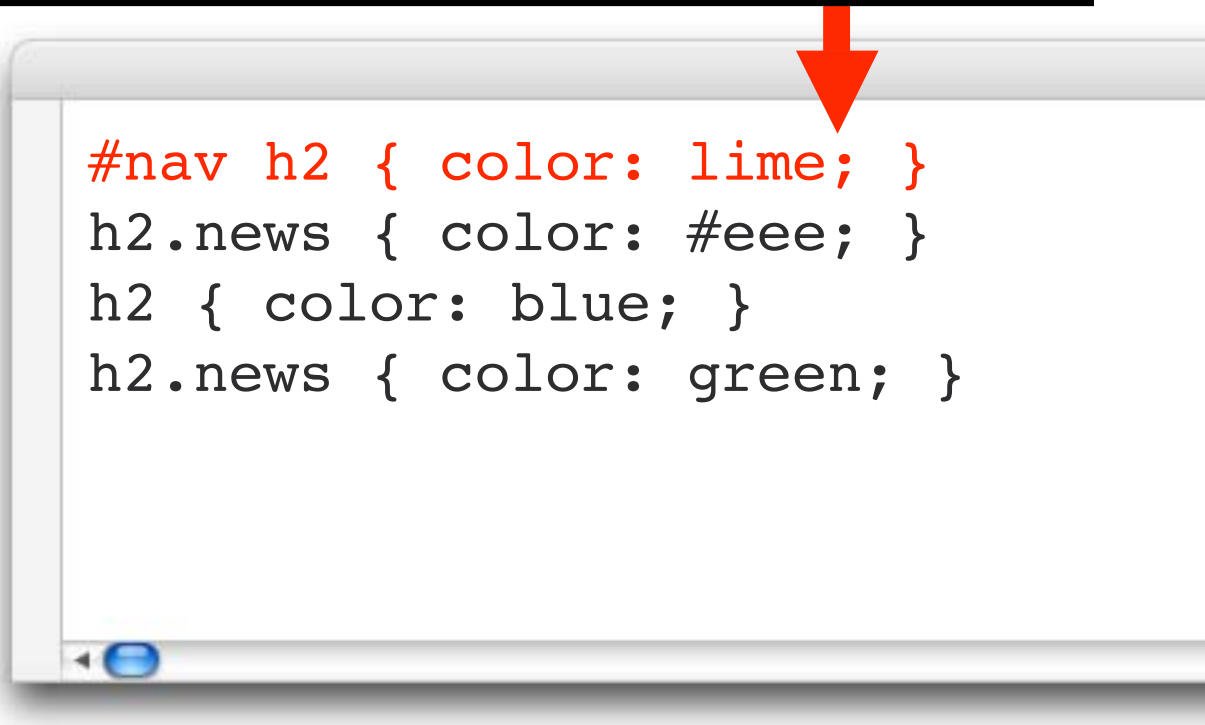
External style sheets and header styles (Author styles)



```
#nav h2 { color: lime; }
h2.news { color: #eee; }
h2 { color: blue; }
h2.news { color: green; }
```

The highlighted selector wins due to specificity - 0,1,0,1 beats 0,0,1,1 and 0,0,0,1

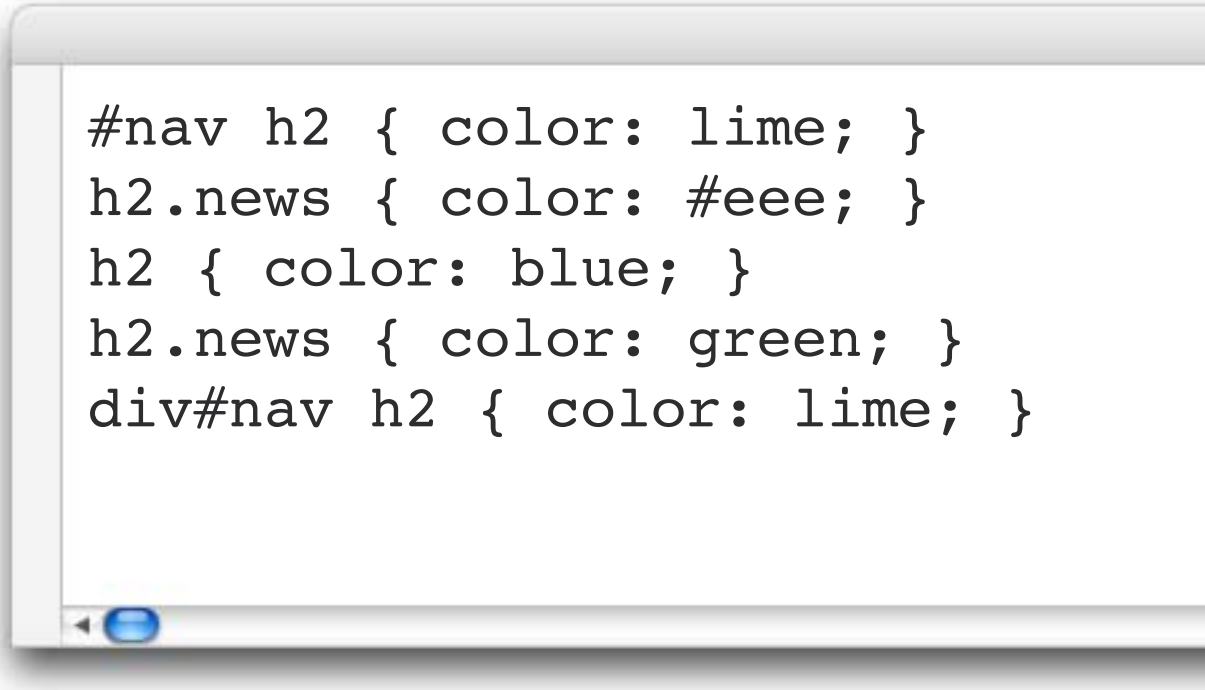
External style sheets and header styles (Author styles)



```
#nav h2 { color: lime; }
h2.news { color: #eee; }
h2 { color: blue; }
h2.news { color: green; }
```

Part 4: Which one **wins?**

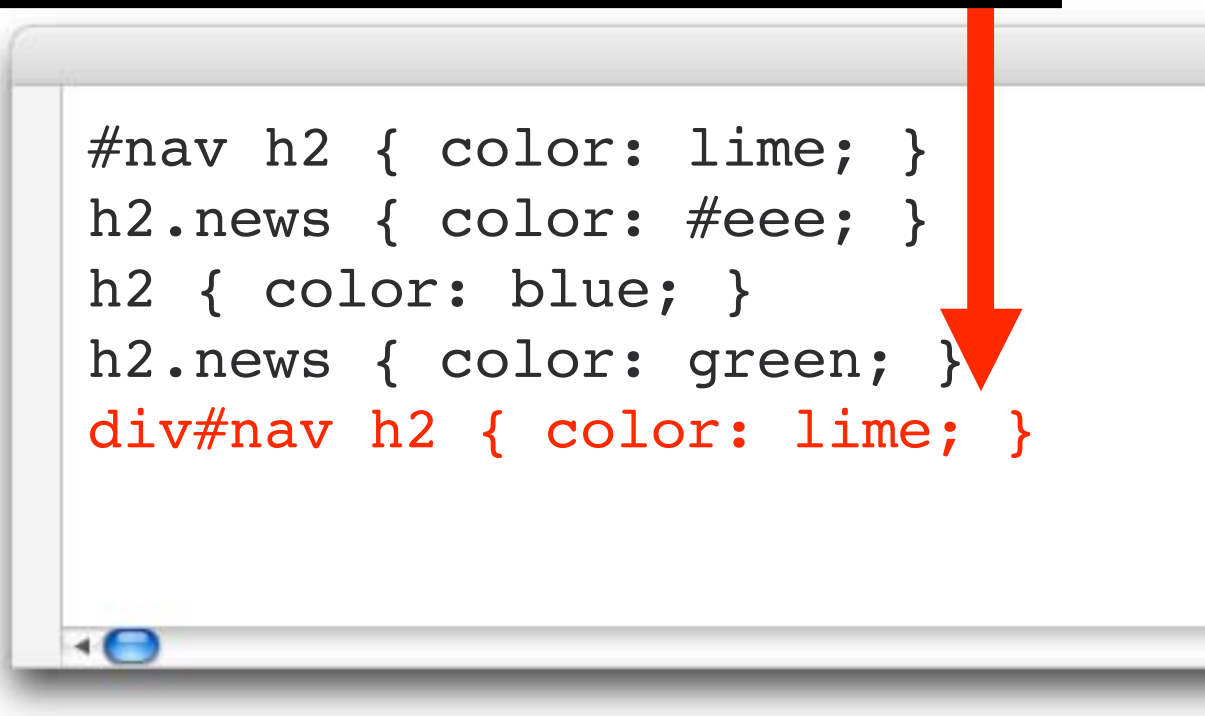
External style sheets and header styles (Author styles)



```
#nav h2 { color: lime; }
h2.news { color: #eee; }
h2 { color: blue; }
h2.news { color: green; }
div#nav h2 { color: lime; }
```

The highlighted selector wins due to specificity - 0,1,0,2 beats 0,1,0,1 and 0,0,1,1 and 0,0,0,1

External style sheets and header styles (Author styles)



We’re done!