

# HTML & CSS FOR BEGINNERS

Girl Develop IT Pittsburgh

Lindsey Bieda & Julie Pagano

# WEEK #2

## Introduction to CSS

Lindsey Bieda & Julie Pagano

# GOALS FOR TODAY

- ★ Introduction to CSS

# CSS

# Cascading Style Sheets

# CSS

Presentation and styling of the HTML document

**HTML + CSS**

**Structure + Style**

# HTML

```
1 <body class="home">
2 <header>
3 <div class="container">
4 <h1 id="logo">
5 <a href="/">Girl Develop It Pittsburgh</a>
6 </h1>
7
8 <ul class="nav">
9
10 <li>
11
12 <a href="/classes">Classes</a>
13
14 </li>
15
16 <li>
17
18 <a href="/community">Community</a>
19
20 </li>
21
22 <li>
23
24 <a href="/resources">Resources</a>
25
26 </li>
27 </ul>
```

+

# CSS

```
1 /* Welcome to Compass.
2 * In this file you should write your main styles. (or central
3 * Import this file using the following HTML or equivalent:
4 * <link href="/stylesheets/screen.css" media="screen, proje
5 * line 17, ../../../../../../.rvm/gems/ruby-1.9.3-p194/girldevelopi
6 html, body, div, span, applet, object, iframe,
7 h1, h2, h3, h4, h5, h6, p, blockquote, pre,
8 a, abbr, acronym, address, big, cite, code,
9 del, dfn, em, img, ins, kbd, q, s, samp,
10 small, strike, strong, sub, sup, tt, var,
11 b, u, i, center,
12 dl, dt, dd, ol, ul, li,
13 fieldset, form, label, legend,
14 table, caption, tbody, tfoot, thead, tr, th, td,
15 article, aside, canvas, details, embed,
16 figure, figcaption, footer, header, hgroup,
17 menu, nav, output, ruby, section, summary,
18 time, mark, audio, video {
19   margin: 0;
20   padding: 0;
21   border: 0;
22   font: inherit;
23   font-size: 100%;
24   vertical-align: baseline;
25 }
26
27 /* line 22, ../../../../../../.rvm/gems/ruby-1.9.3-p194/girldevelopi
28 html {
```

=





# CSS ZEN GARDEN



## ONE HTML PAGE, MANY STYLESHEETS

# ADDING CSS TO HTML: INLINE

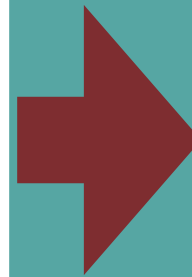
```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <h1 style="color: red;">red title!</h1>
  </body>
</html>
```

# ADDING CSS TO HTML: EMBEDDED

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1 { color: red; }
    </style>
  </head>
  <body>
    <h1>>red title!</h1>
  </body>
</html>
```

# ADDING CSS TO HTML: LINKED

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet"
          type="text/css"
          href="style.css" />
  </head>
  <body>
    <h1>red title!</h1>
  </body>
</html>
```

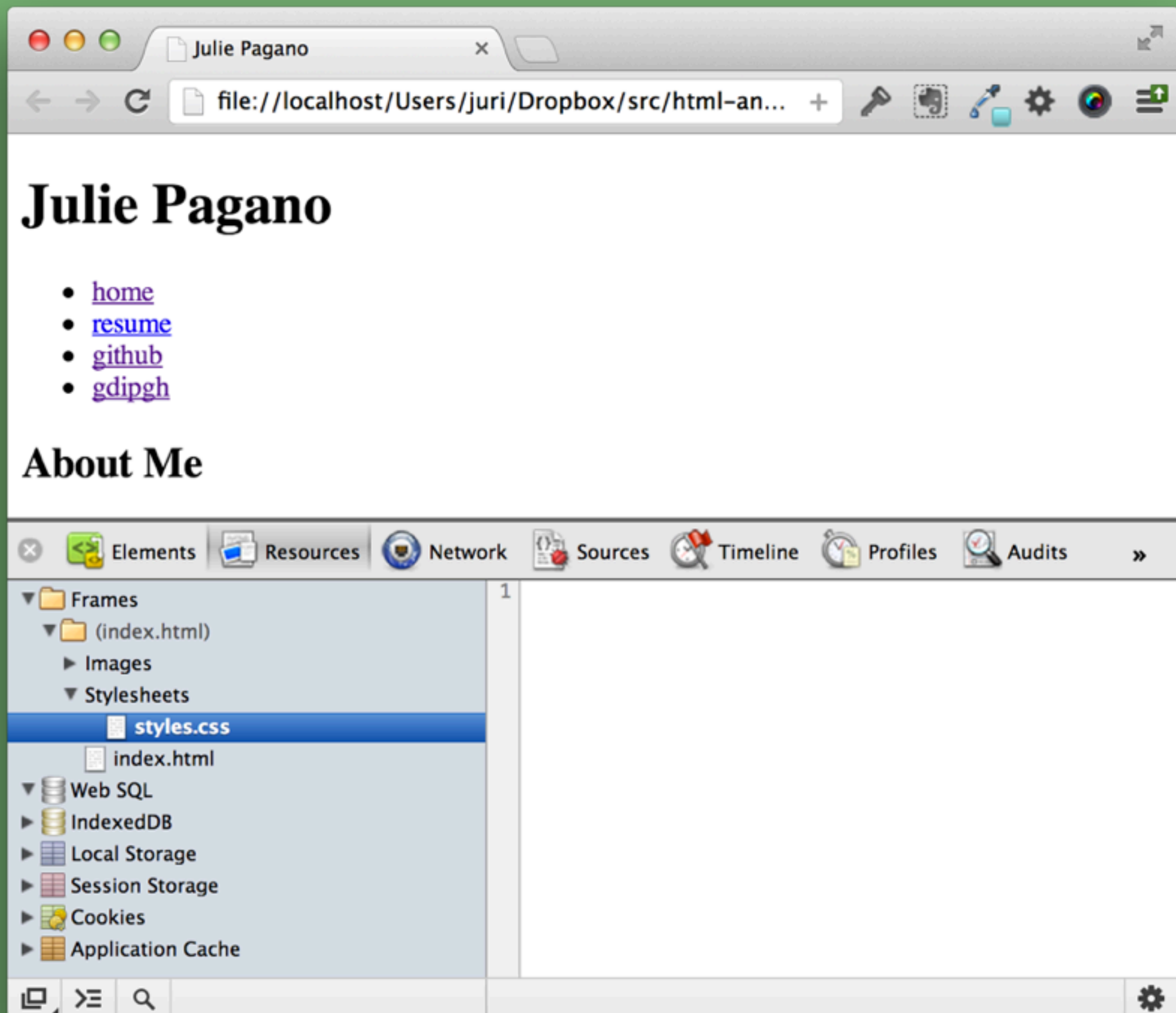


style.css

```
h1 {
  color: red;
}
```

# ACTIVITY: SET UP SITE CSS

1. Create a folder named “styles” in your working directory to hold stylesheets
2. Create *styles.css* in the styles directory
3. Link to *styles.css* in <head>
4. Use the developer tools to check your work



# ID ATTRIBUTE

```
<div id="header">  
  some content  
</div>
```

- › Unique identifier
- › A document can only have one of each id
- › Begin with a letter and followed by any number of letters, numbers, hyphens, underscores

# CLASS ATTRIBUTE

```
<div class="code highlighted">  
  some content  
</div>
```

- › Multiple elements can share a class
- › Element can have multiple classes separated by a space
- › Begin with a letter and followed by any number of letters, numbers, hyphens, underscores



# NAMING CLASSES AND IDS

- › **Do** describe what it is for
- › **DO NOT** describe what it looks like
- › Use a name that will make sense to you in the future

# EXAMPLES OF BAD NAMES

- › bigText
- › red
- › leftColumn
- › underlined

# EXAMPLES OF GOOD NAMES

- › header
- › footer
- › alert
- › copyright
- › navigation

# ANATOMY OF THE CSS RULE

```
selector {  
  property: value;  
}
```

# SELECTOR

```
selector {  
  property: value;  
}
```

# PROPERTY

```
selector {  
  property: value;  
}
```

# VALUE

```
selector {  
  property: value;  
}
```

# DECLARATION

```
selector {  
  property: value;  
}
```



# RULE

```
selector {  
  property: value;  
}
```

# SELECTOR: ELEMENT

```
div {  
  property: value;  
}
```

Selects all **div** elements

```
<div>foo</div>  
<div>bar</div>
```

Example of selected HTML elements

# SELECTOR: ID

```
#header {  
  property: value;  
}
```

Selects all elements with the id “**header**”

```
<div id="header">  
  Welcome!  
</div>
```

Example of selected HTML element

# SELECTOR: CLASS

```
.tip {  
  property: value;  
}
```

Selects all elements with the class “**tip**”

```
<span class="tip">HTML gives sites structure.</span>  
<span class="tip">CSS makes sites pretty.</span>
```

Example of selected HTML elements

# SELECTOR: POSITION

```
p span {  
  property: value;  
}
```

Selects all span elements that are in a paragraph.

```
<p>  
  This, <span>that</span>, and the <span>other</span>.  
</p>
```

Example of selected HTML elements

# PROPERTY VALUES

Properties have one or more comma separated values

```
p {  
  font-size: 14px;  
  font-weight: normal;  
  font-family: Arial, sans-serif;  
}
```

# PROPERTY: FONT-FAMILY

Defines which font is used

```
p {  
  font-family: 'Open Sans';  
  font-family: serif;  
  font-family: "Arial", sans-serif;  
}
```

- Specific font name
- Generic name
- Comma separated list

# PROPERTY: FONT-WEIGHT

Defines the weight or boldness of the font

```
p {  
  font-weight: normal;  
  font-weight: 400;  
  font-weight: bold;  
  font-weight: 700;  
}
```

- Named weight
- Numeric weight



# PROPERTY: FONT-SIZE

Specifies the size of the font

```
p {  
  font-size: 12px;  
}
```

# SHORTHAND PROPERTY: FONT

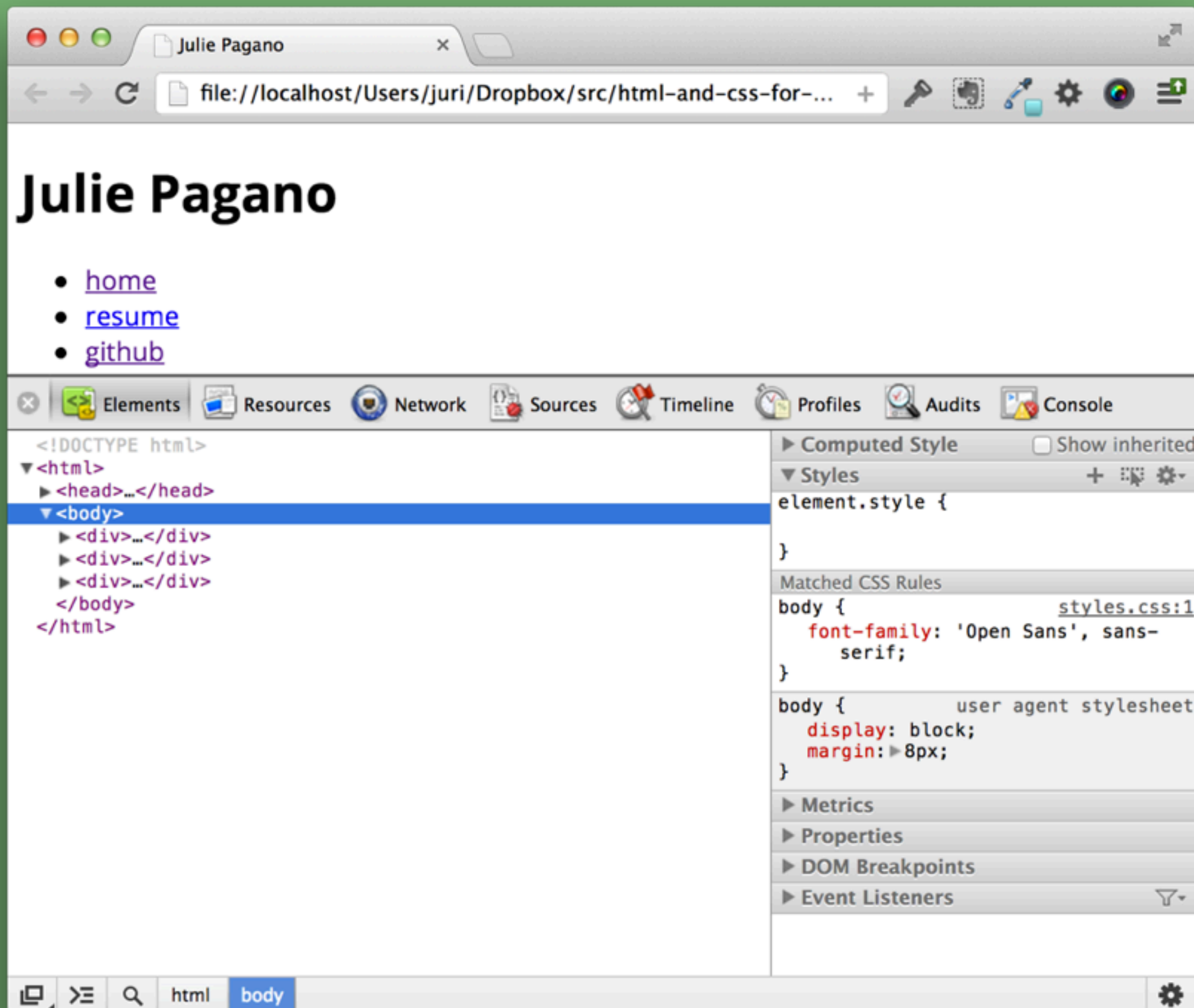
```
p {  
  font-style: italic;  
  font-weight: bold;  
  font-size: 14px;  
  font-family: sans-serif;  
}
```

can also be written as

```
p {  
  font: italic bold 14px sans-serif;  
}
```

# ACTIVITY: PICK A FONT

1. Go to <http://www.google.com/webfonts>
2. Select a font and click **“quick use”**
3. Check the font styles you want
4. Add the standard code to `<head>` in *index.html*
5. Set the font-family on the body in *style.css*
6. Use the developer tools to check your work



# PROPERTY: COLOR

Defines the color of the text

```
p {  
  color: blue;  
  color: #0000FF;  
  color: rgb(0, 0, 255);  
}
```

- Named color
- Hexadecimal (hex) color
- Red Green Blue (rgb) color

# PROPERTY: BACKGROUND-COLOR

Defines the color of the background

```
p {  
  background-color: red;  
  background-color: #FF0000;  
  background-color: rgb(255, 0, 0);  
}
```

- Named color
- Hexadecimal (hex) color
- Red Green Blue (rgb) color

# ACTIVITY: COLOR SCHEMES

1. Go to <http://colorschemedesigner.com/>
2. Pick a color you like and a scheme (e.g. triad)
3. Take a look at the light/dark page examples
4. Add color styles to *style.css*
  - a) Add ids or classes to HTML, as needed
  - b) Set the background color for the body, header, footer
  - c) Set the text color
  - d) Set the color for links
5. Inspect the color styles in the developer tools

# Color Scheme Designer

2002-2010 © Petr Stanicek • v3.51 • [Blog & News](#)



mono



complement



triad



tetrad



analogic



accented  
analogic

Undo

Redo

Random

Colorblind

Color space

Export

Help



Scheme ID: [3n51Tw0w0w0w0](#)

CREATE  
FREE  
STUNNING  
WEBSITES  
GO >>

Hue: 215°

[opposite](#)

Angle: 30°



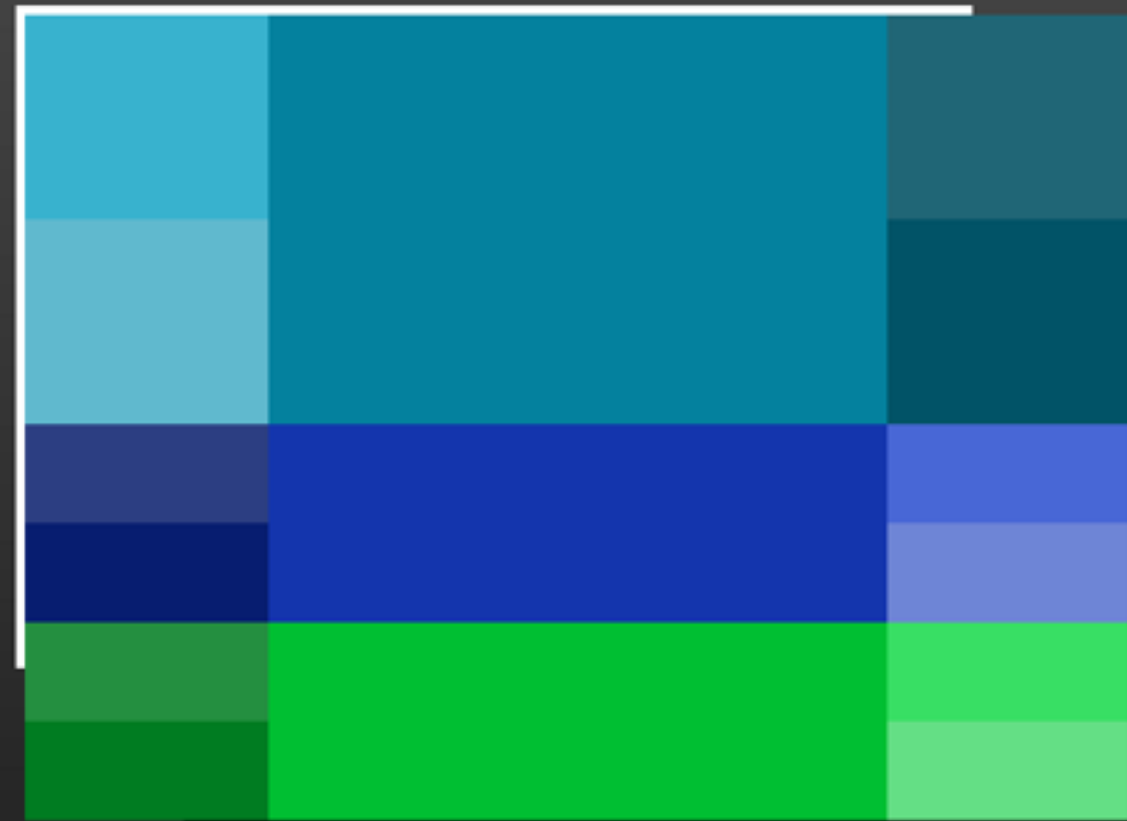
R: 2 %  
G: 51 %  
B: 62 %

RGB: 04819E

Hues

Adjust Scheme

Color List



Show sample text

Preview

Light page example

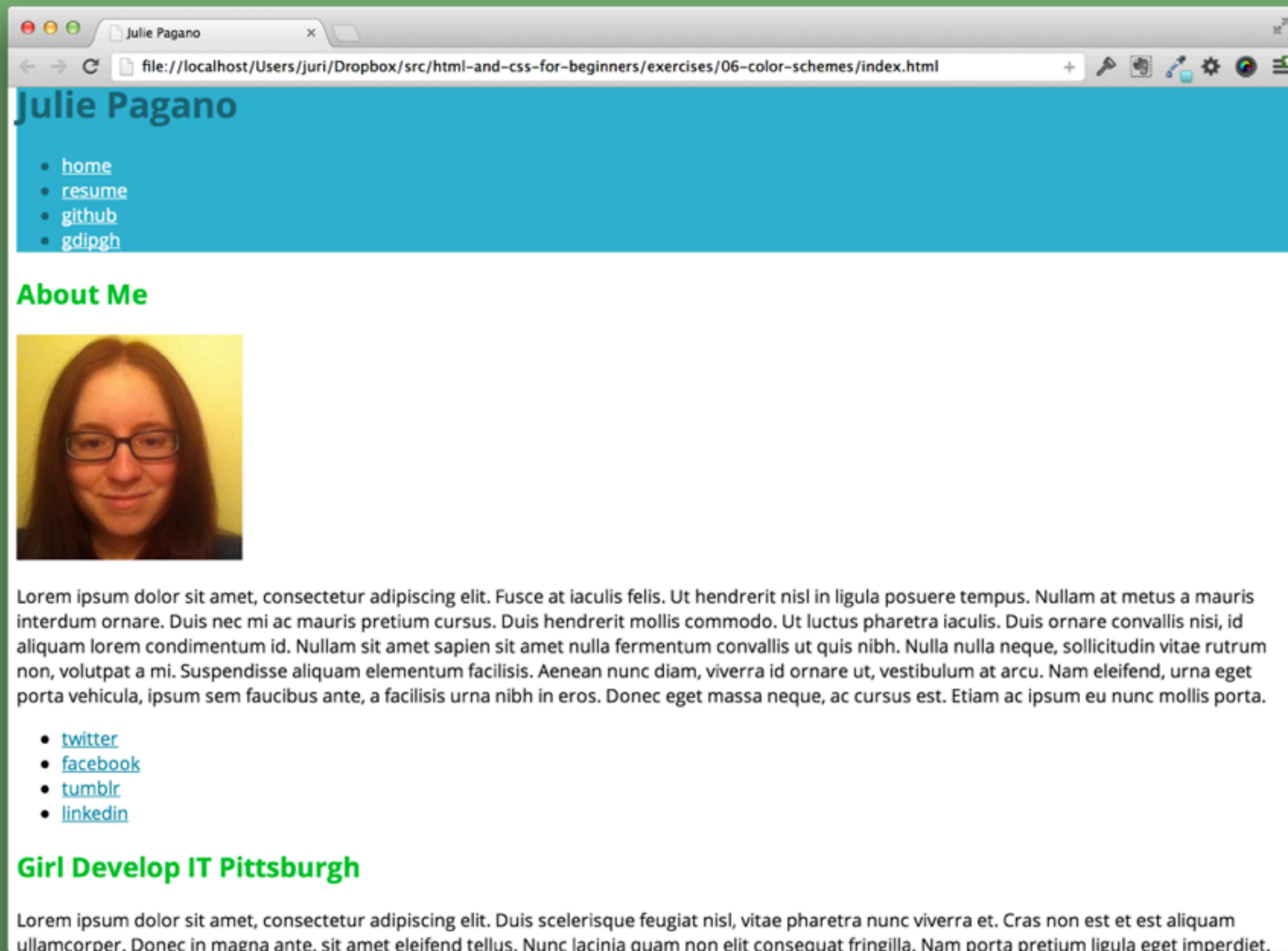
Dark page example

Do you like this app?

Donate

Related: [FREE Logo Design](#) | [Wedding Invitations](#) | [Save the Dates](#)





# PROPERTY: TEXT-DECORATION

Used to set some text formatting

```
p {  
  text-decoration: overline underline;  
}
```

- › Values: none, underline, overline, line-through
- › Multiple values allowed

# PROPERTY: TEXT-ALIGN

Determines the alignment of text within an element

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

› Values: left, right, center, justify

# CSS COMMENTS

Comments are added inside `/* */`

```
/* Header */  
#header {  
    color: red;  
}
```

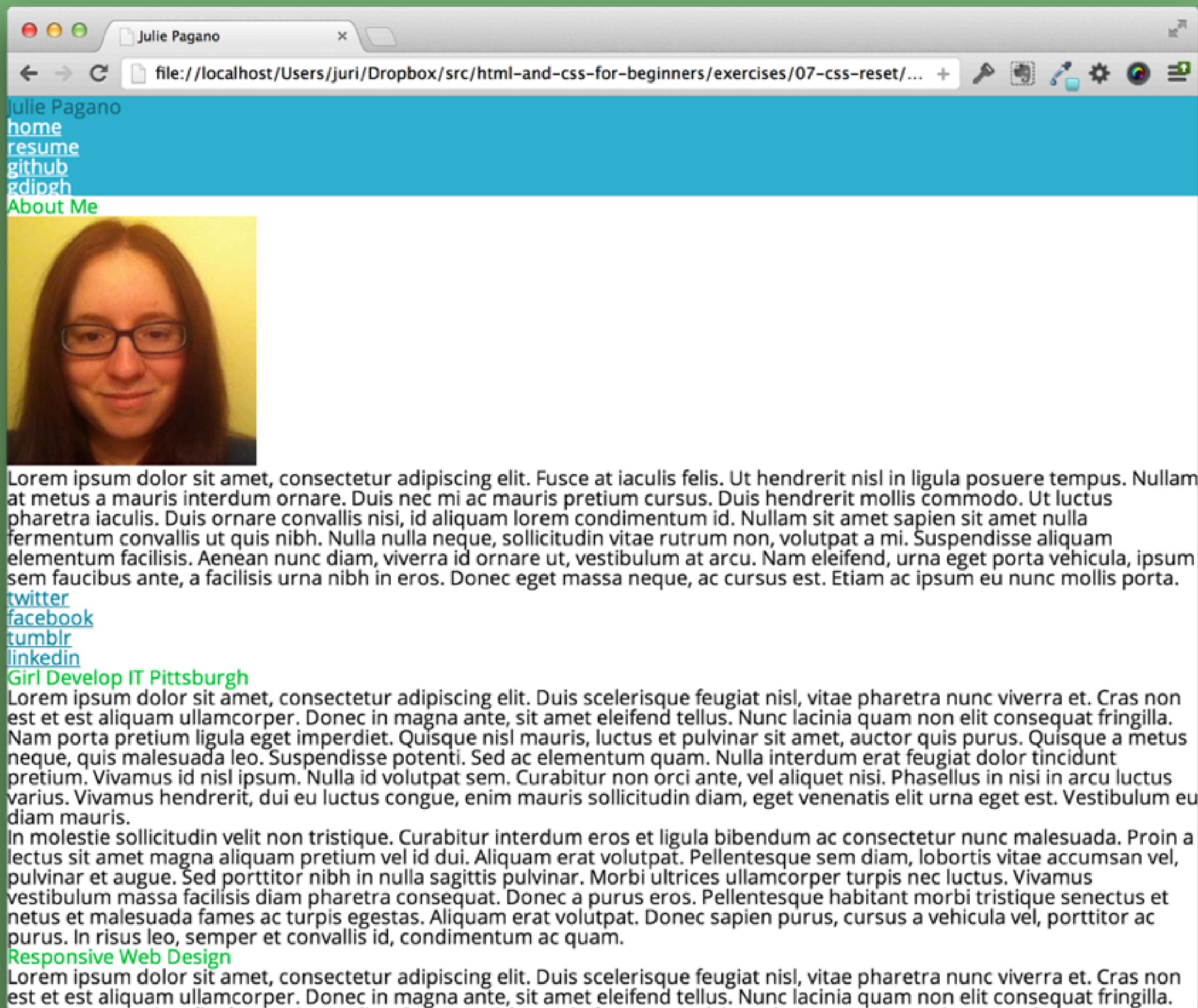
- › Used for separating sections of styles
- › Used to explain complicated behavior

# CSS RESET

- › CSS resets are used to set all styles to the same baseline
- › This is useful for eliminating inconsistencies across browsers

# ACTIVITY: SET UP CSS RESET

1. Go to <http://meyerweb.com/eric/tools/css/reset/reset.css>
2. Add css reset to your styles directory
3. Link to reset in <head>
4. View the page in your browser and note the differences



# MULTIPLE SELECTORS

```
h1 { font-weight: bold; }  
h2 { font-weight: bold; }
```

Can be combined into

```
h1, h2 {  
    font-weight: bold;  
}
```



# PSEUDO-CLASS SELECTORS

```
a { color: blue; }  
a:visited { color: green; }  
a:hover { color: red; }
```

- **:hover** selects when user hovers over an element
- **:visited** selects when user has visited a link

# ACTIVITY: PSEUDO-CLASS SELECTORS

1. Add a css rule to modify the color of links on hover
2. Add a css rule to modify the color of visited links

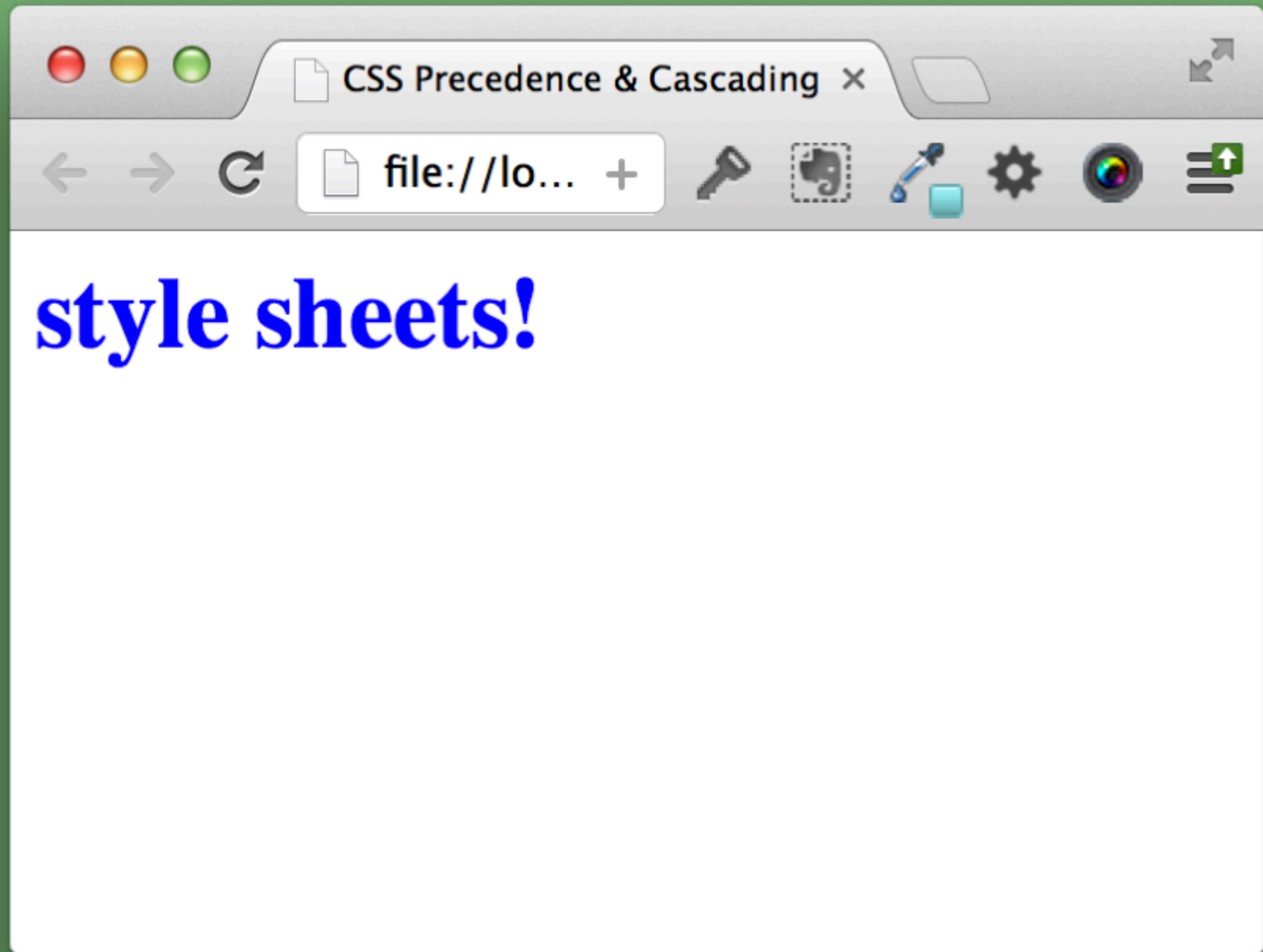
# INHERITANCE

Elements can inherit some CSS rules (e.g. font, color) from their parents

# WHAT COLOR IS THE TEXT?

```
<div id="main" class="foo">  
  <h1 id="heading" class="bar baz">  
    style sheets!  
  </h1>  
</div>
```

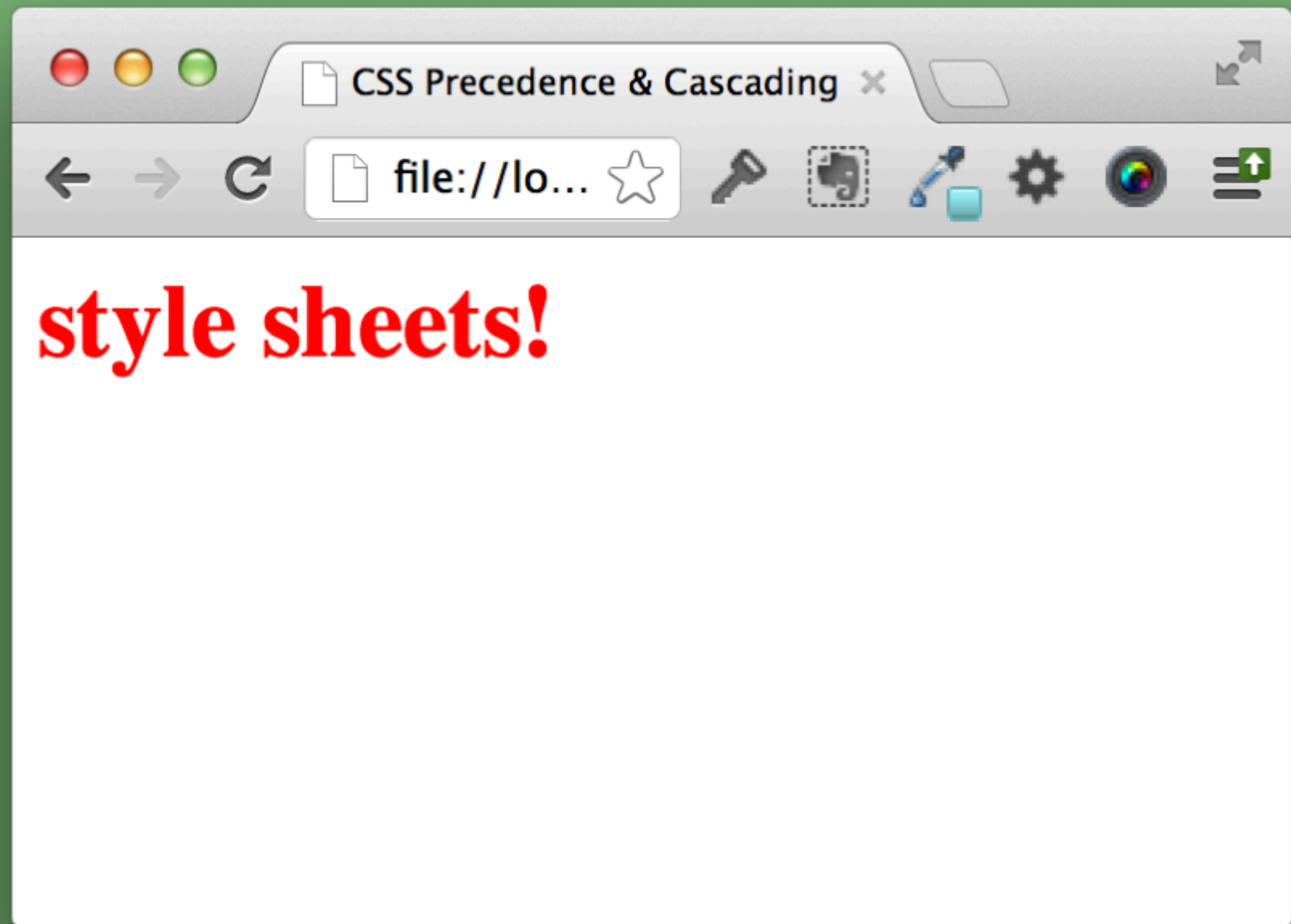
```
#main {  
  color: blue;  
}
```



# WHAT COLOR IS THE TEXT?

```
<div id="main" class="foo">  
  <h1 id="heading" class="bar baz">  
    style sheets!  
  </h1>  
</div>
```

```
div {  
  color: blue;  
}  
h1 {  
  color: red;  
}
```



# SPECIFICITY

How the browser picks a value when there are multiple rules



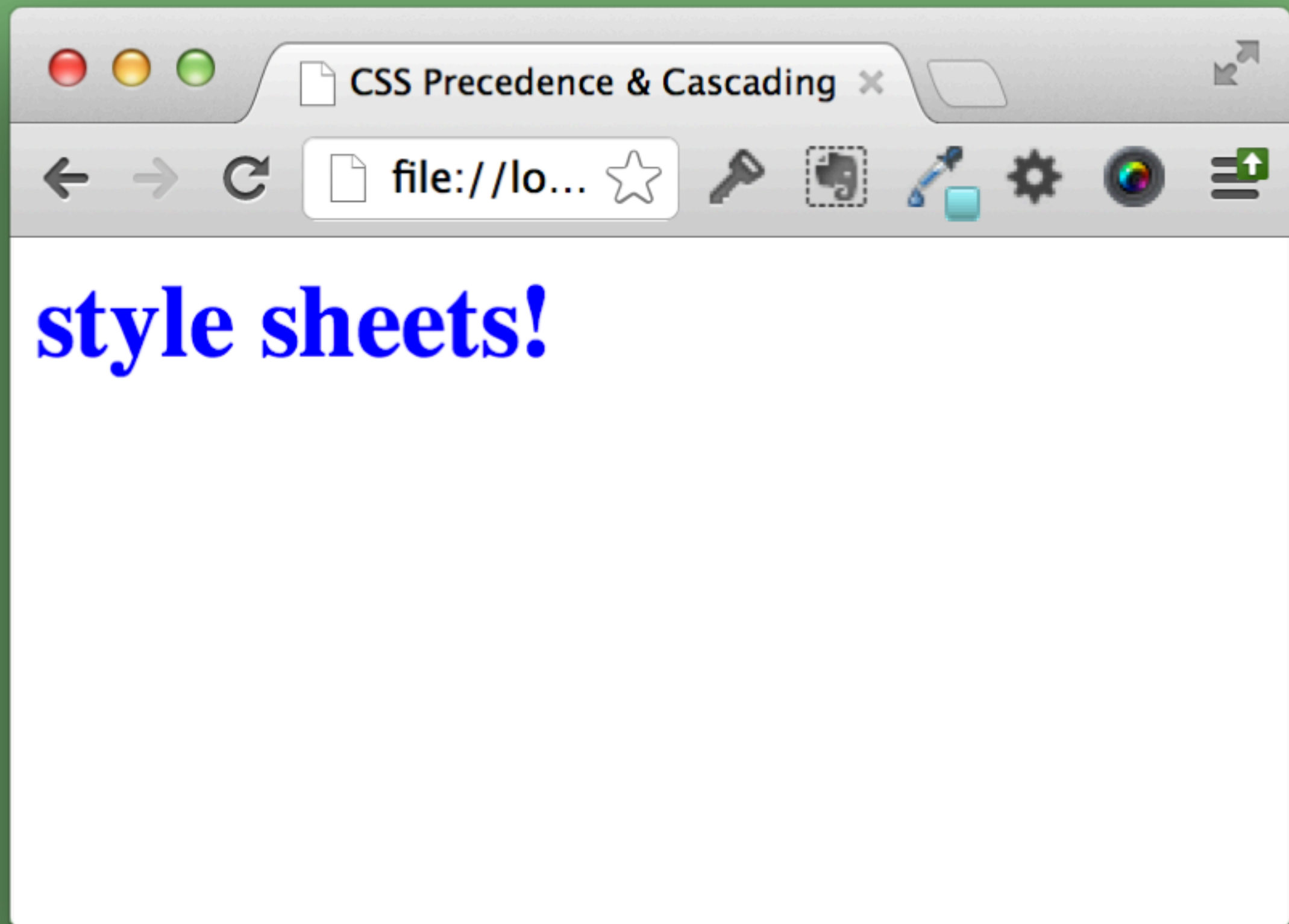
# SPECIFICITY

- › Element selectors (least)
- › Class selectors
- › Pseudo-class selectors
- › ID selectors
- › Inline style (most)

# WHAT COLOR IS THE TEXT?

```
<div id="main" class="foo">  
  <h1 id="heading" class="bar baz">  
    style sheets!  
  </h1>  
</div>
```

```
#heading {  
  color: blue;  
}  
.bar {  
  color: red;  
}
```



CSS Rules Quiz - Specificity

file:///localhost/Users/juri/Dro...

# style sheets!

Elements Resources Network Sources Timeline

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <div id="main" class="foo">
      <h1 id="heading" class="bar baz">style
      sheets!</h1>
    </div>
  </body>
</html>
```

Computed Style

Styles

element.style {

Matched CSS Rules

#heading { specificity1.html:6

color: blue;

.bar { specificity1.html:9

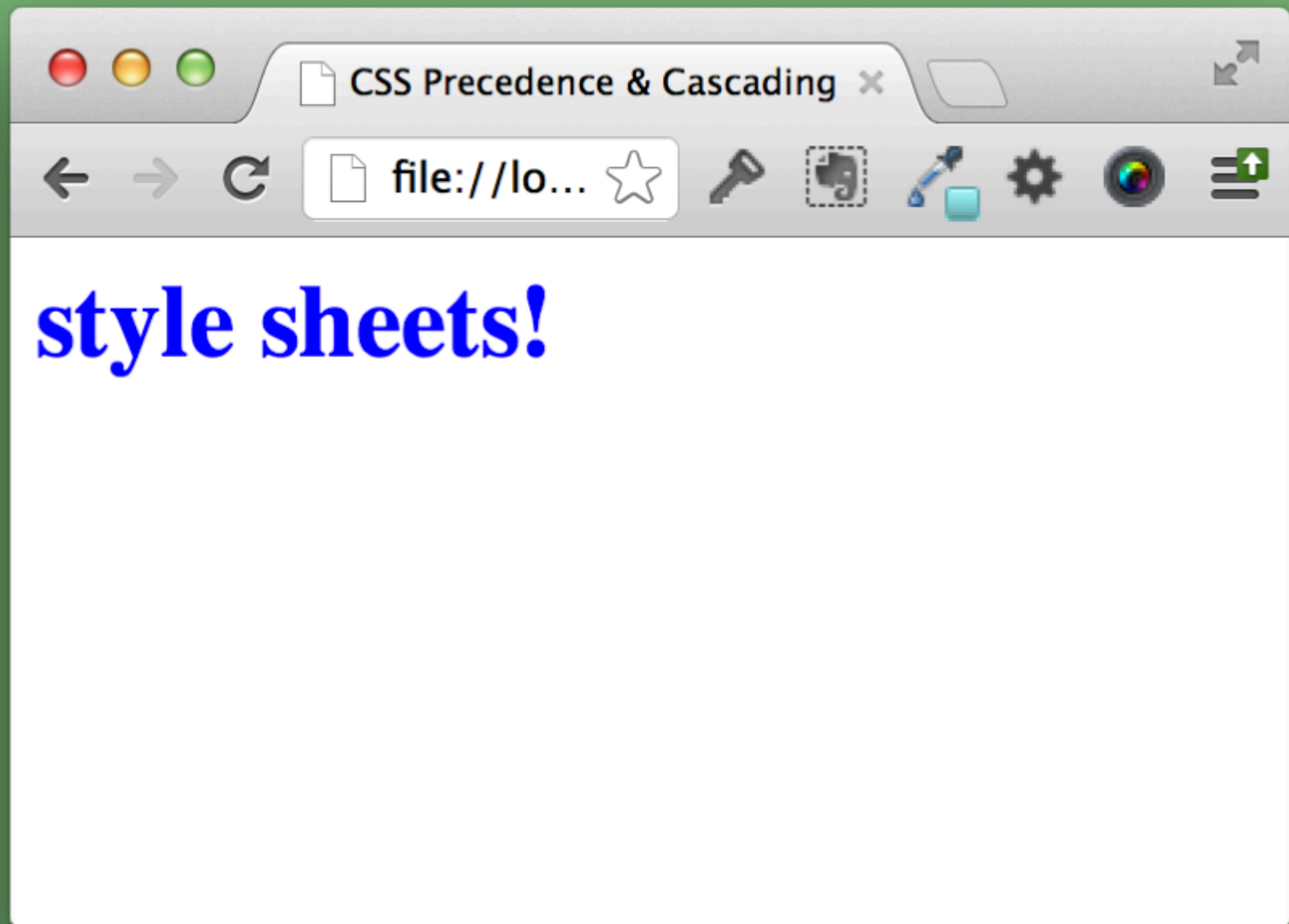
color: red;

html body div#main.foo h1#heading.bar.baz

# WHAT COLOR IS THE TEXT?

```
<div id="main" class="foo">  
  <h1 id="heading" class="bar baz">  
    style sheets!  
  </h1>  
</div>
```

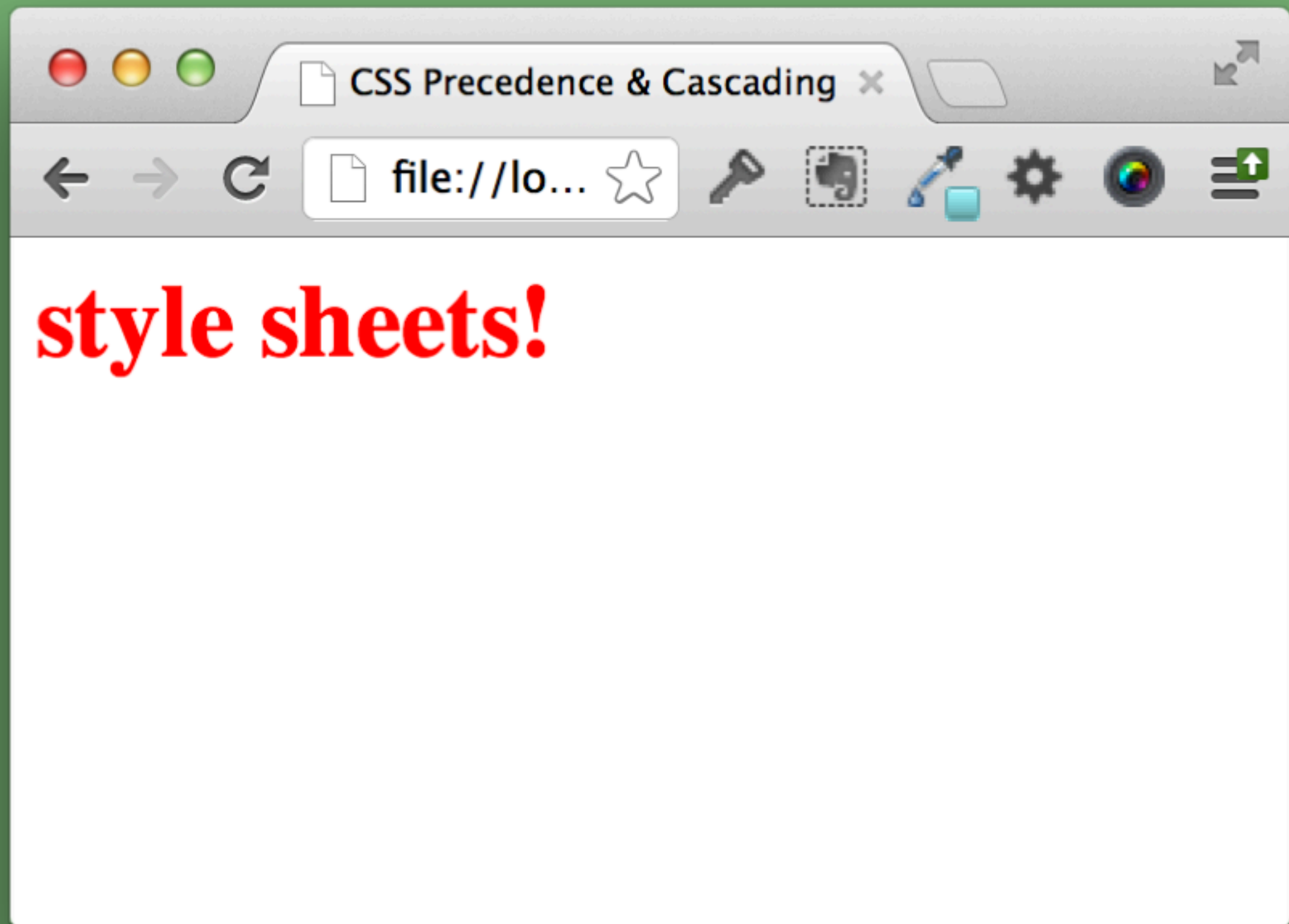
```
.bar.baz {  
  color: blue;  
}  
.bar {  
  color: red;  
}
```



# WHAT COLOR IS THE TEXT?

```
<div id="main" class="foo">  
  <h1 id="heading" class="bar baz">  
    style sheets!  
  </h1>  
</div>
```

```
.bar.baz {  
  color: blue;  
}  
#heading {  
  color: red;  
}
```





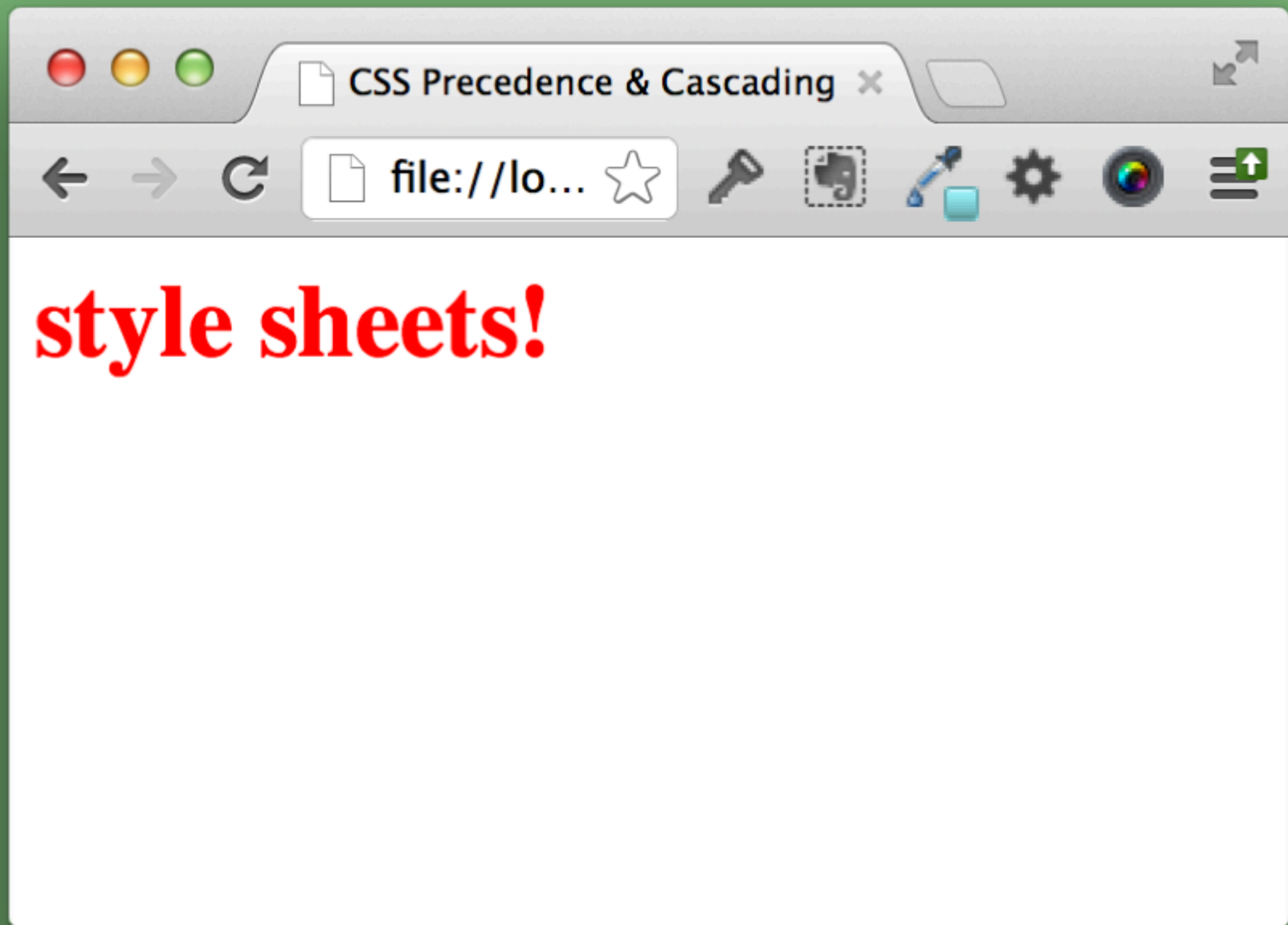
# ORDER

Later rules trump earlier rules (assuming equal specificity)

# WHAT COLOR IS THE TEXT?

```
<div id="main" class="foo">  
  <h1 id="heading" class="bar baz">  
    style sheets!  
  </h1>  
</div>
```

```
.bar {  
  color: blue;  
}  
.baz {  
  color: red;  
}
```



# !IMPORTANT

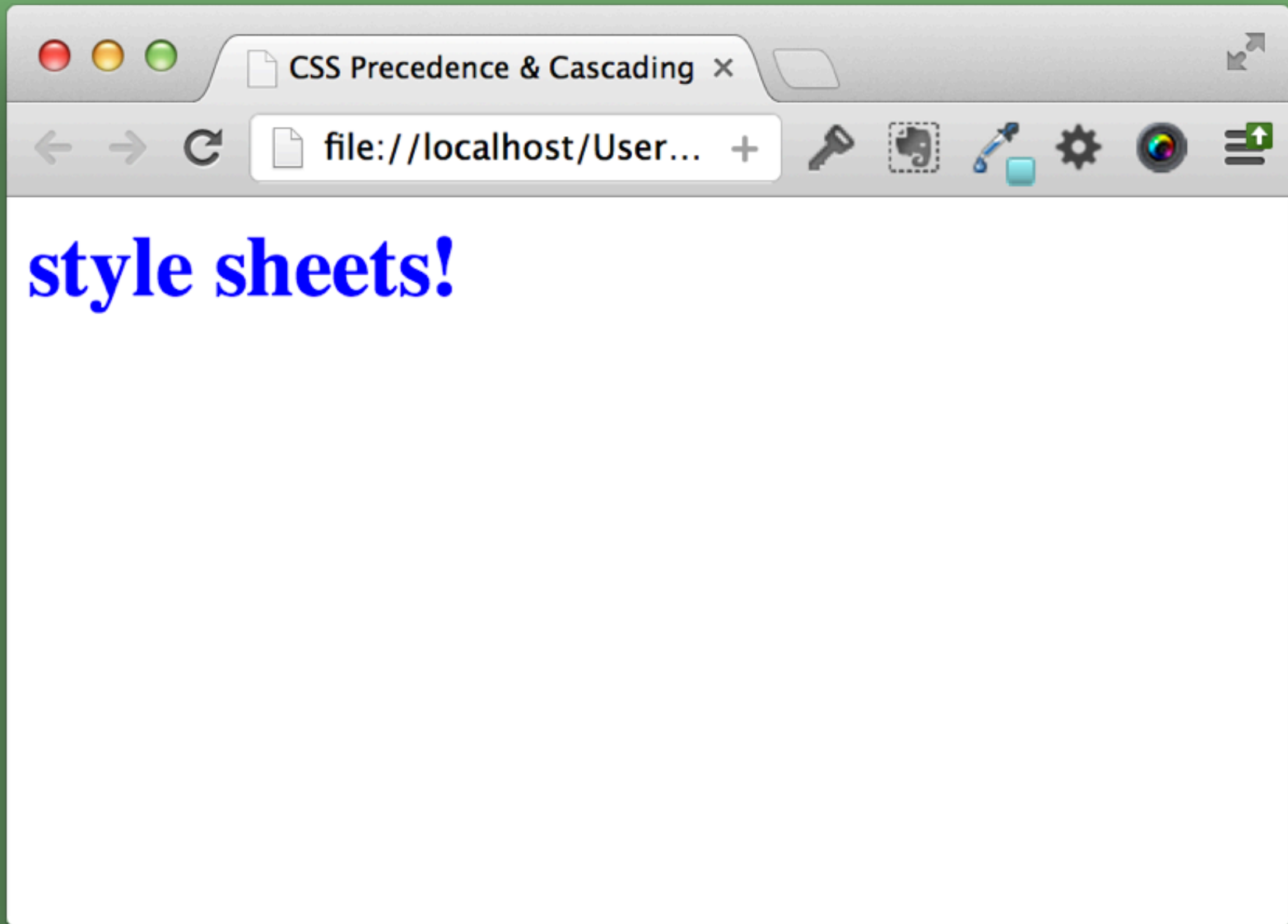
```
h2 {  
  color: red !important;  
}
```

- › Trumps other rules
- › Use sparingly - method of last resort

# WHAT COLOR IS THE TEXT?

```
<div id="main" class="foo">  
  <h1 id="heading" class="bar baz">  
    style sheets!  
  </h1>  
</div>
```

```
.bar {  
  color: blue !important;  
}  
.baz {  
  color: red;  
}
```



# CSS REFERENCES

- Mozilla Developer Network
- HTML Dog

# ACTIVITY: CONTINUE SITE

1. Focus on styling
2. Adjust HTML, as needed
3. Use CSS & HTML references (Mozilla Developer Network)