

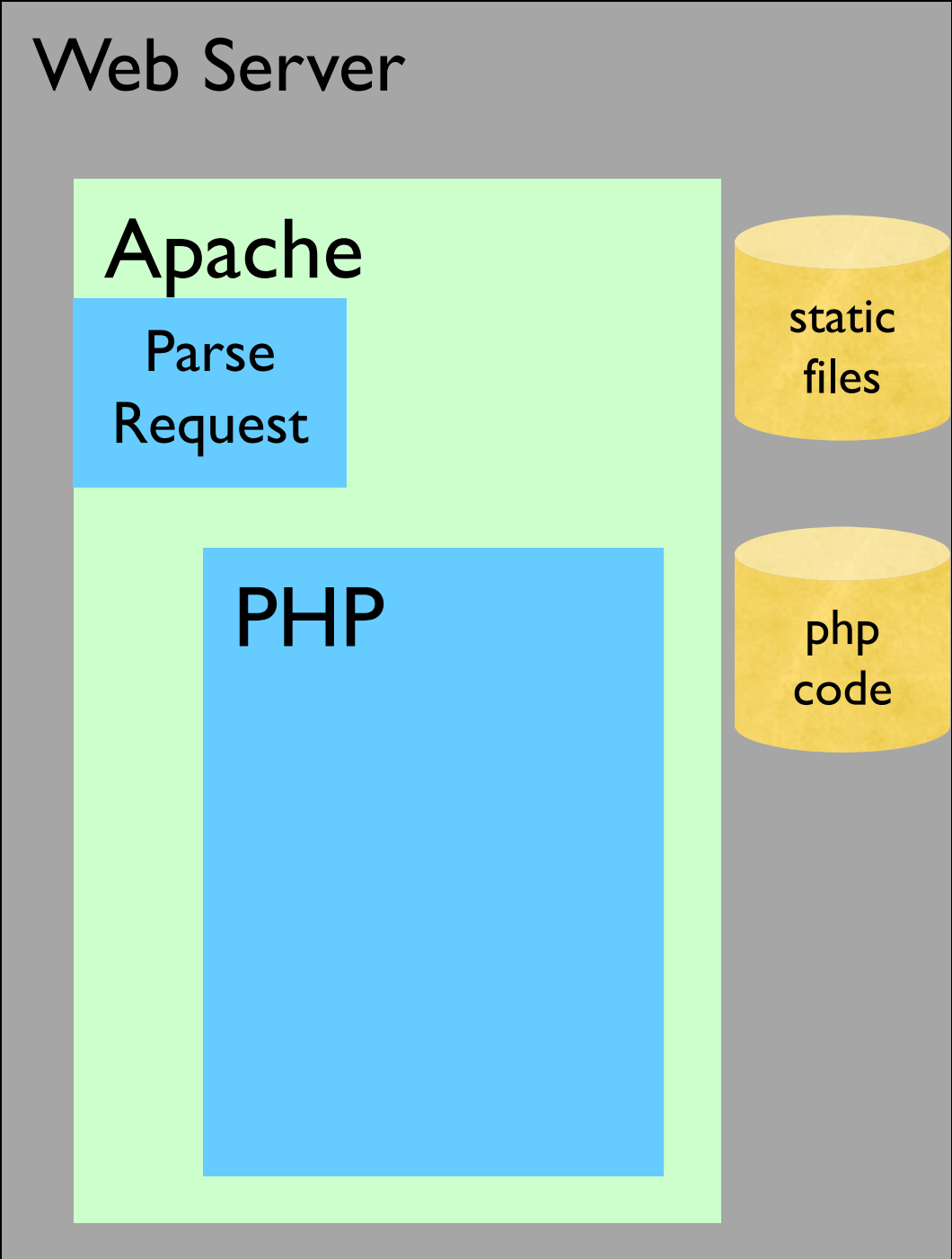
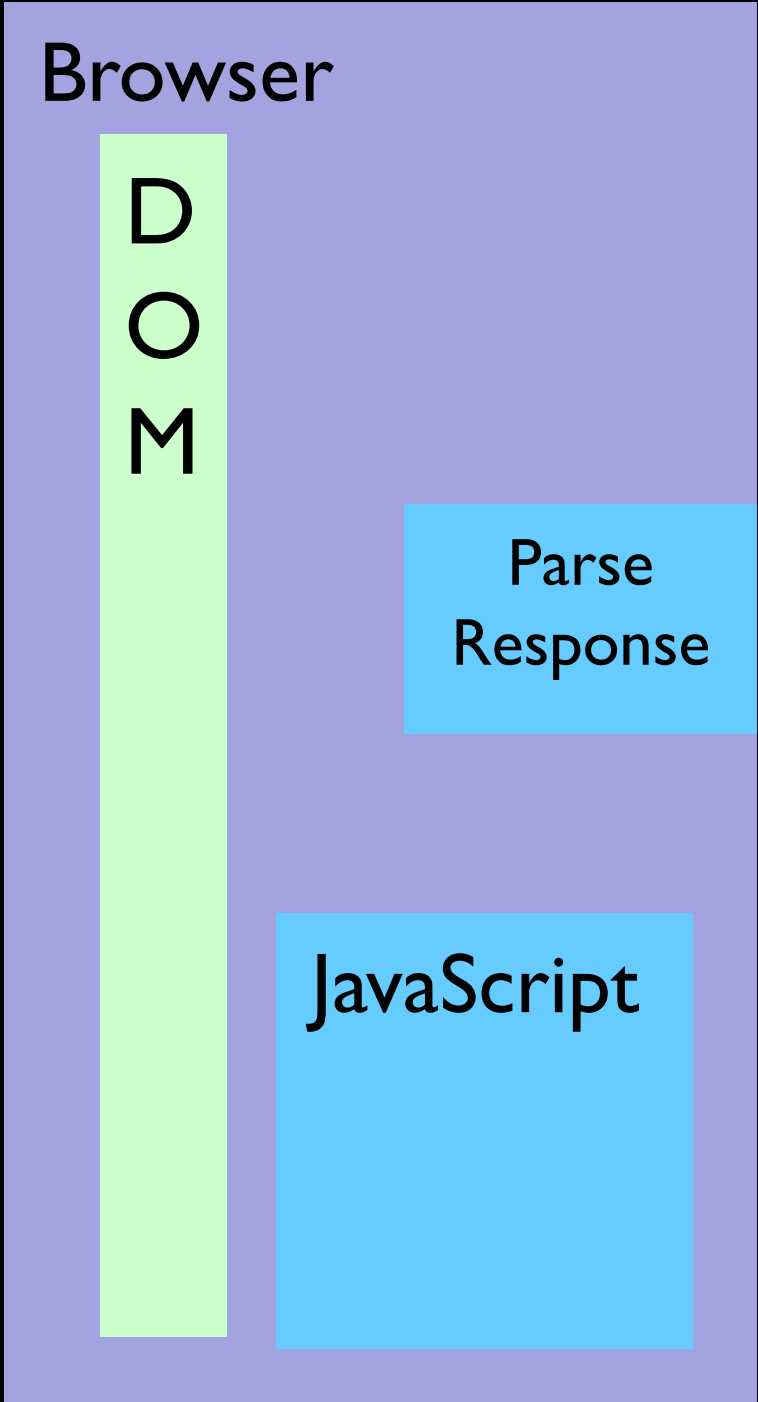
# Cascading Style Sheets

Charles Severance  
[www.wa4e.com](http://www.wa4e.com)

<https://www.wa4e.com/code/css.zip>



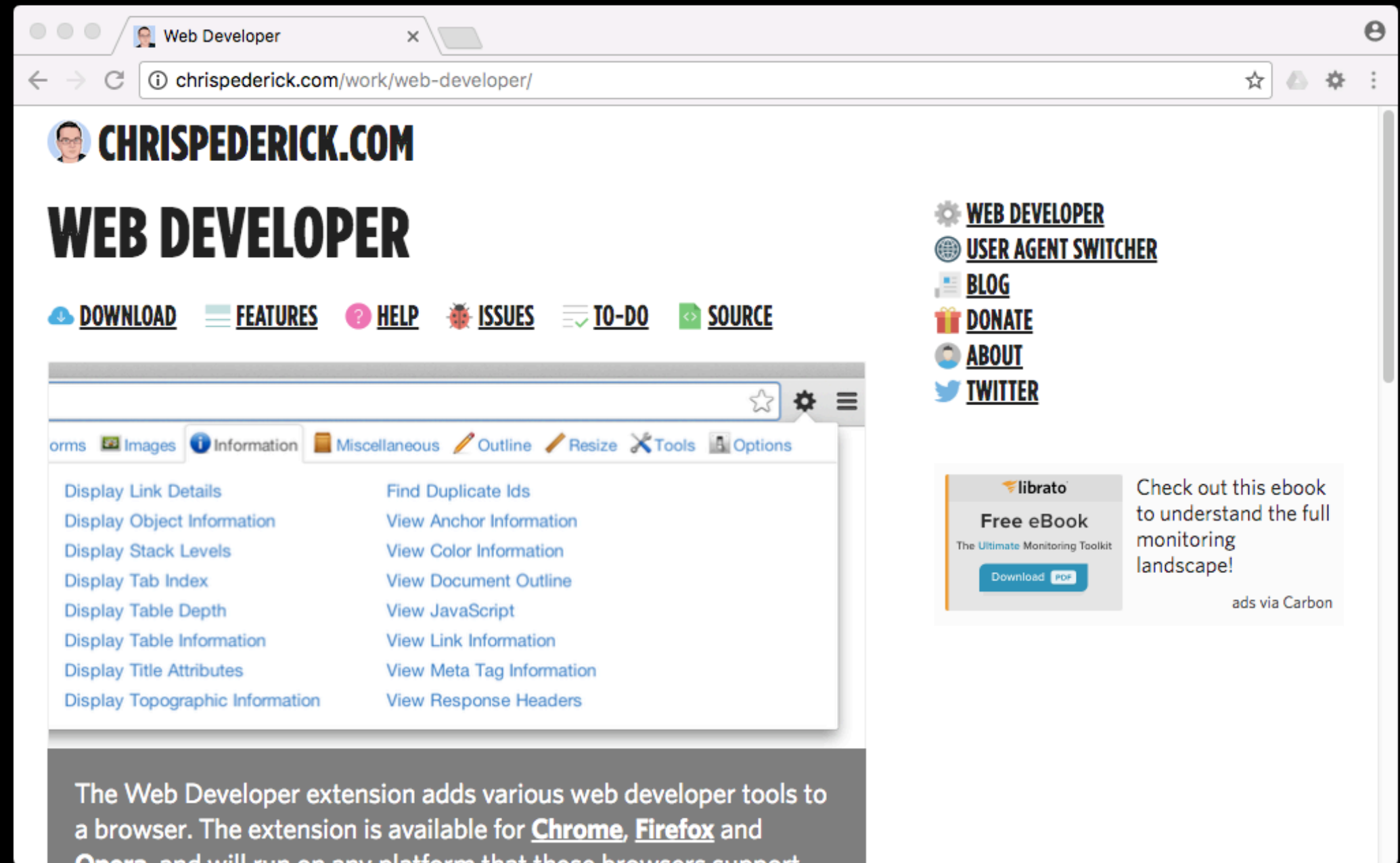
Time



RRC/HTTP

SQL

# More than Developer Console

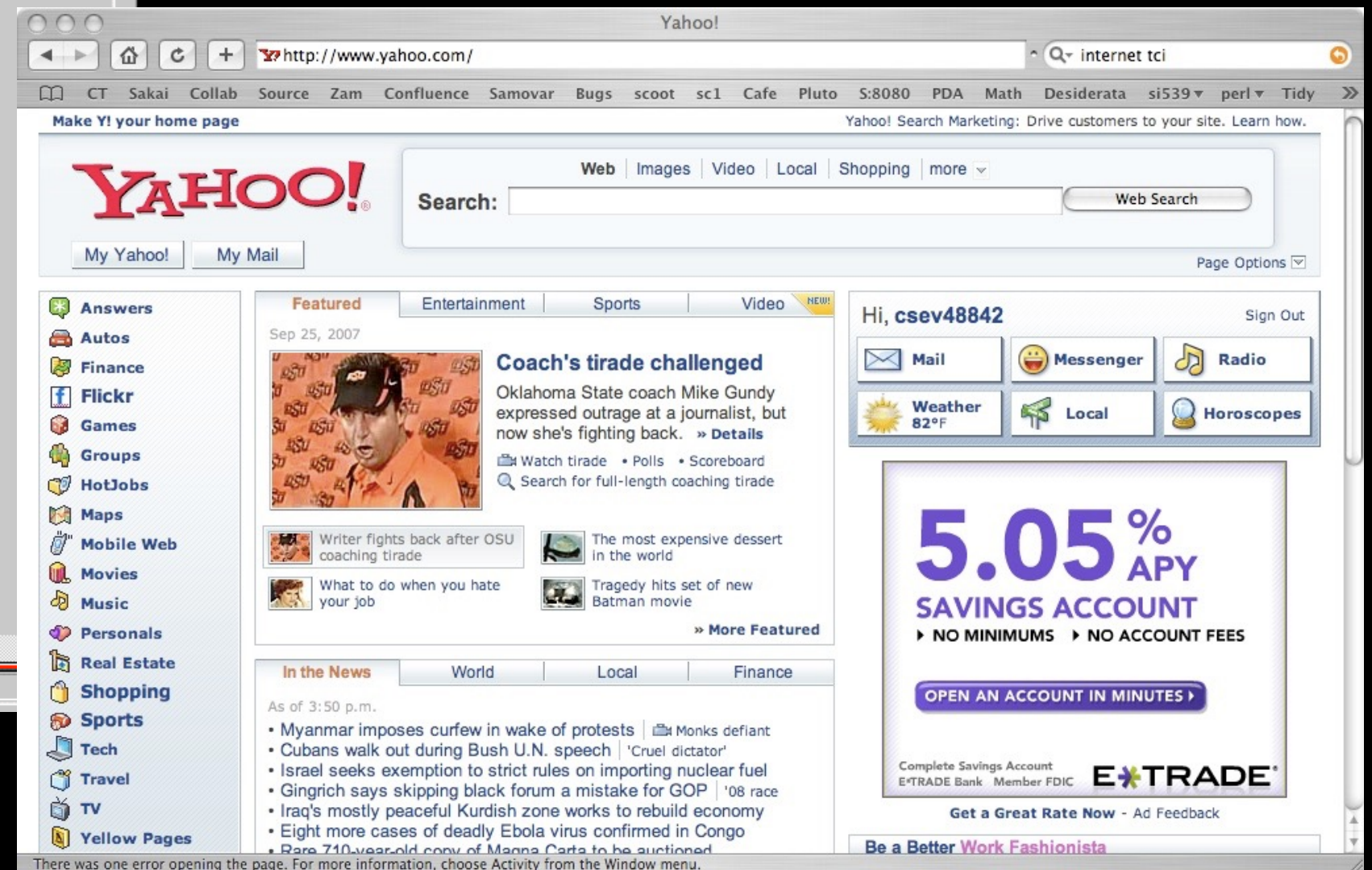


<http://chrispederick.com/work/web-developer/>

HTML has evolved a *\*lot\** over the years - as computers and networks have gotten faster



1995

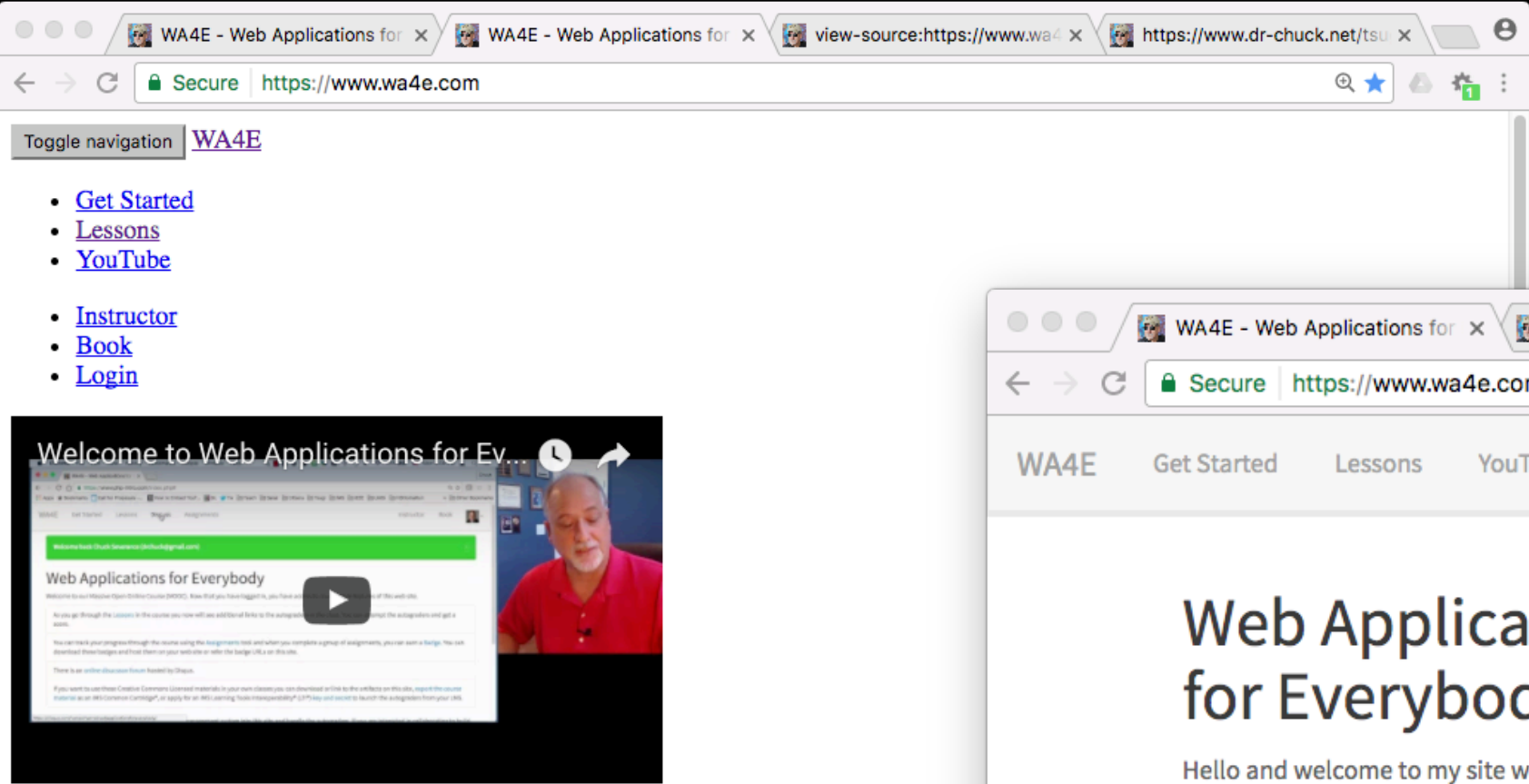


2007



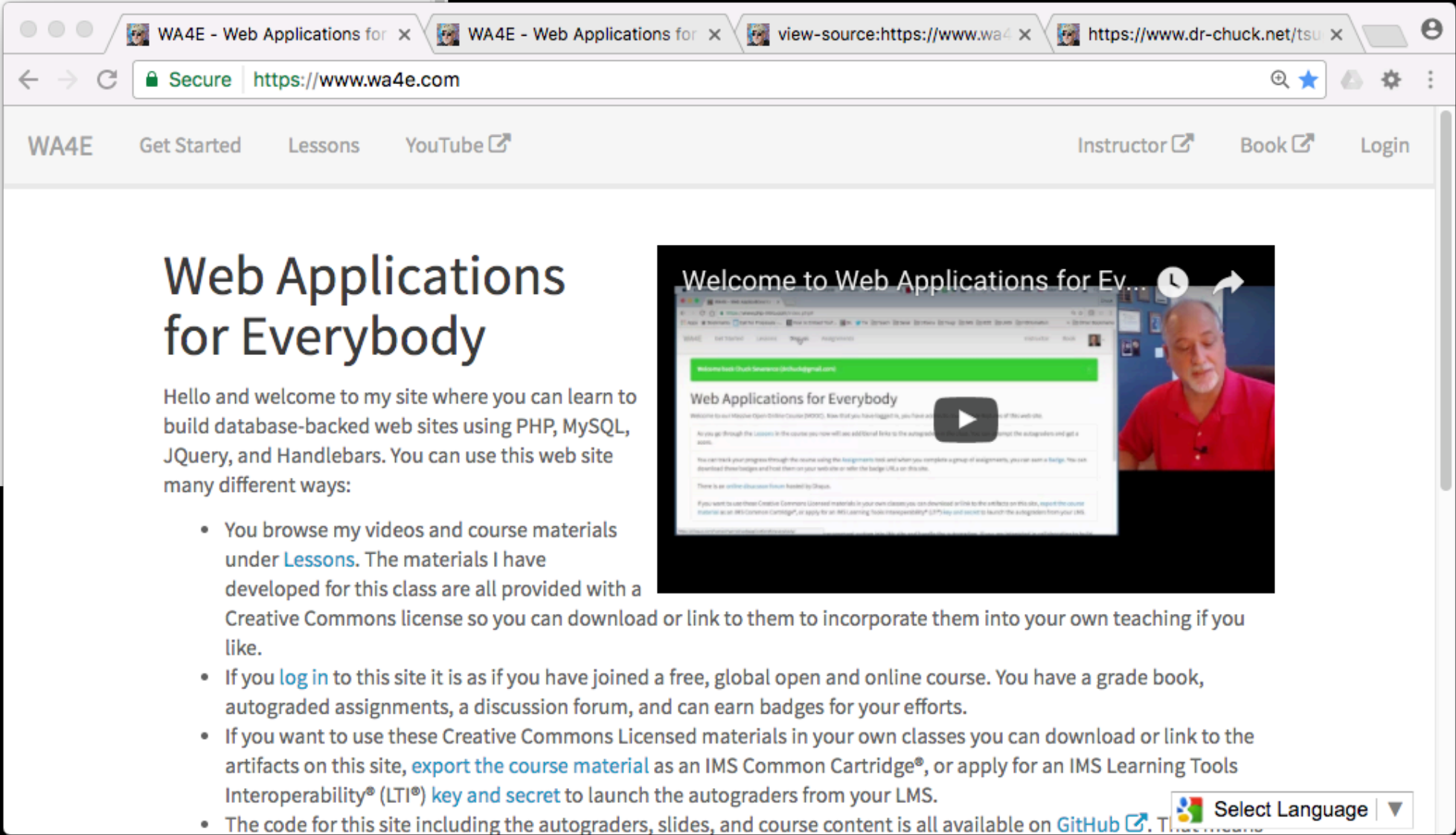


Without CSS



Web Applications for Everybody

With CSS

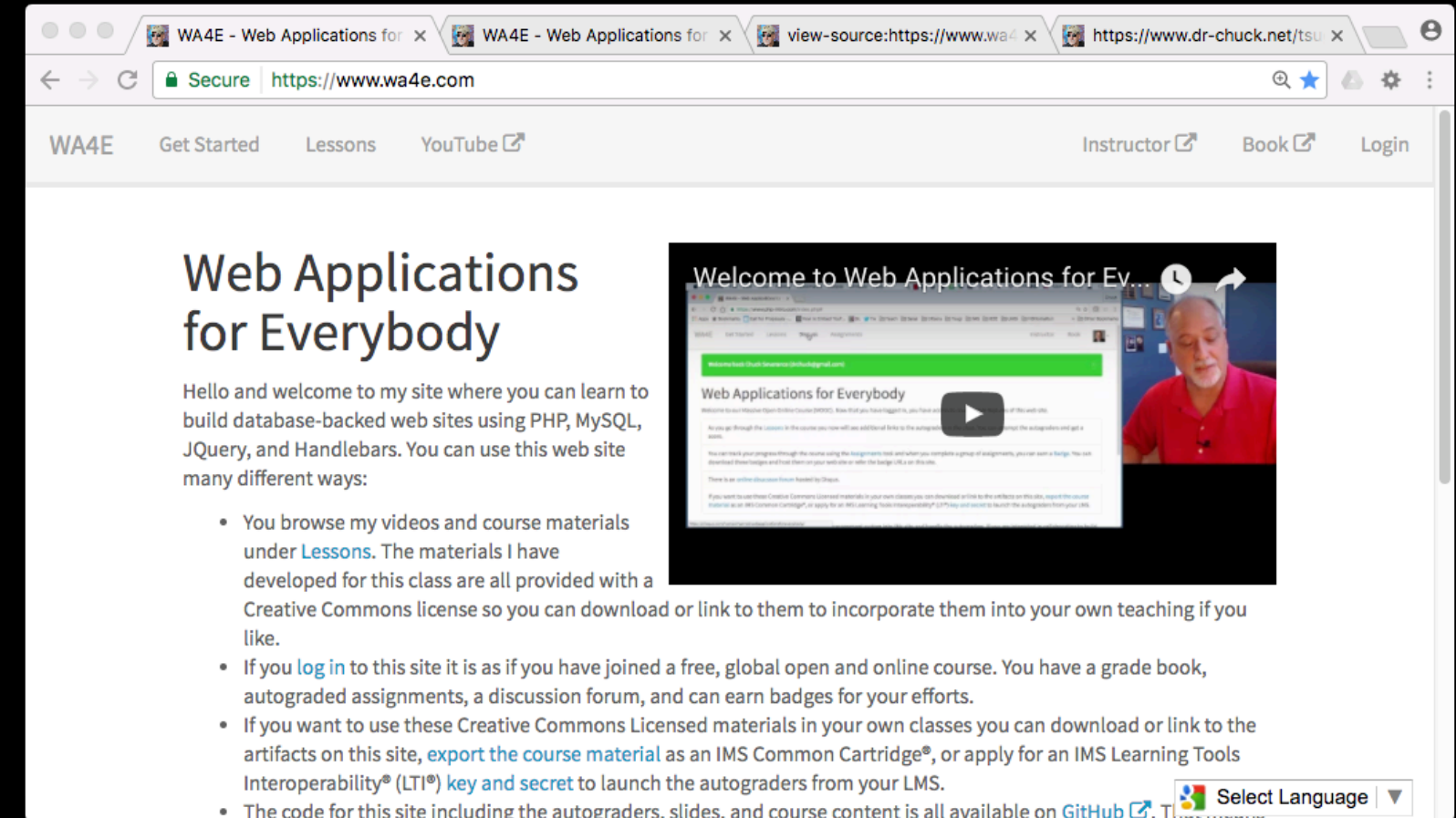
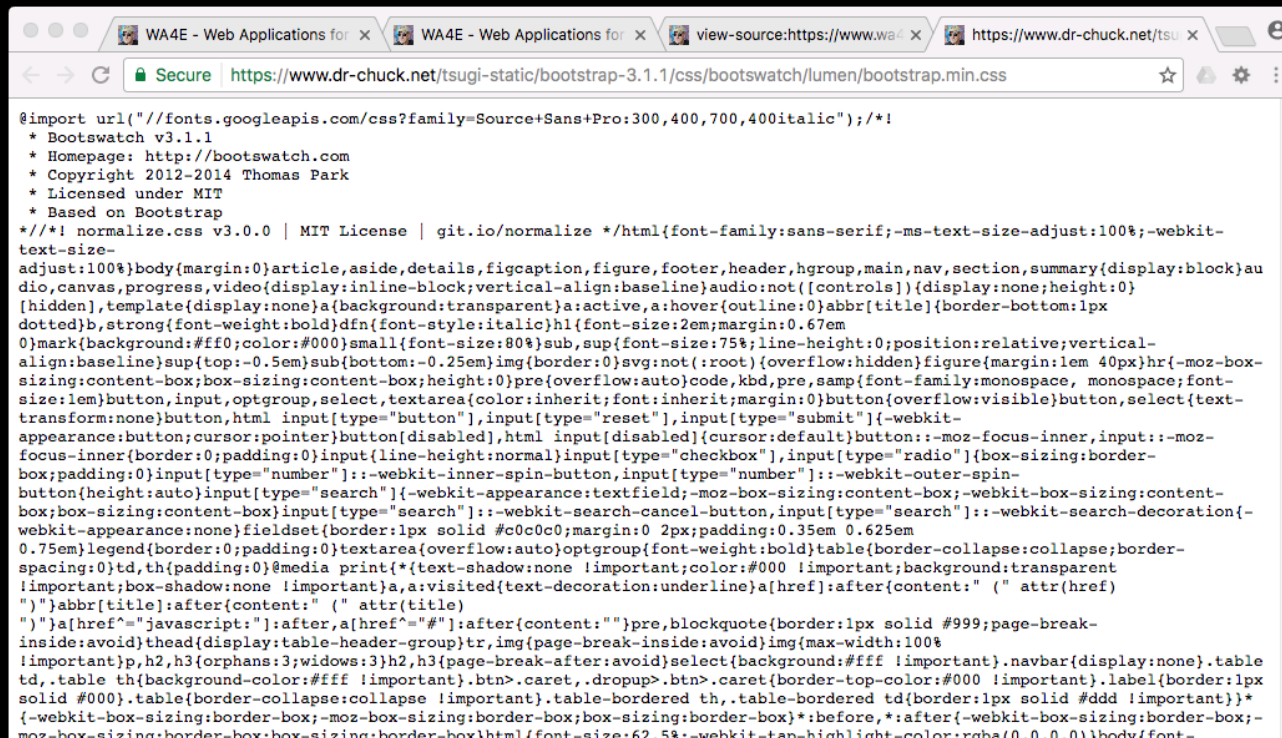
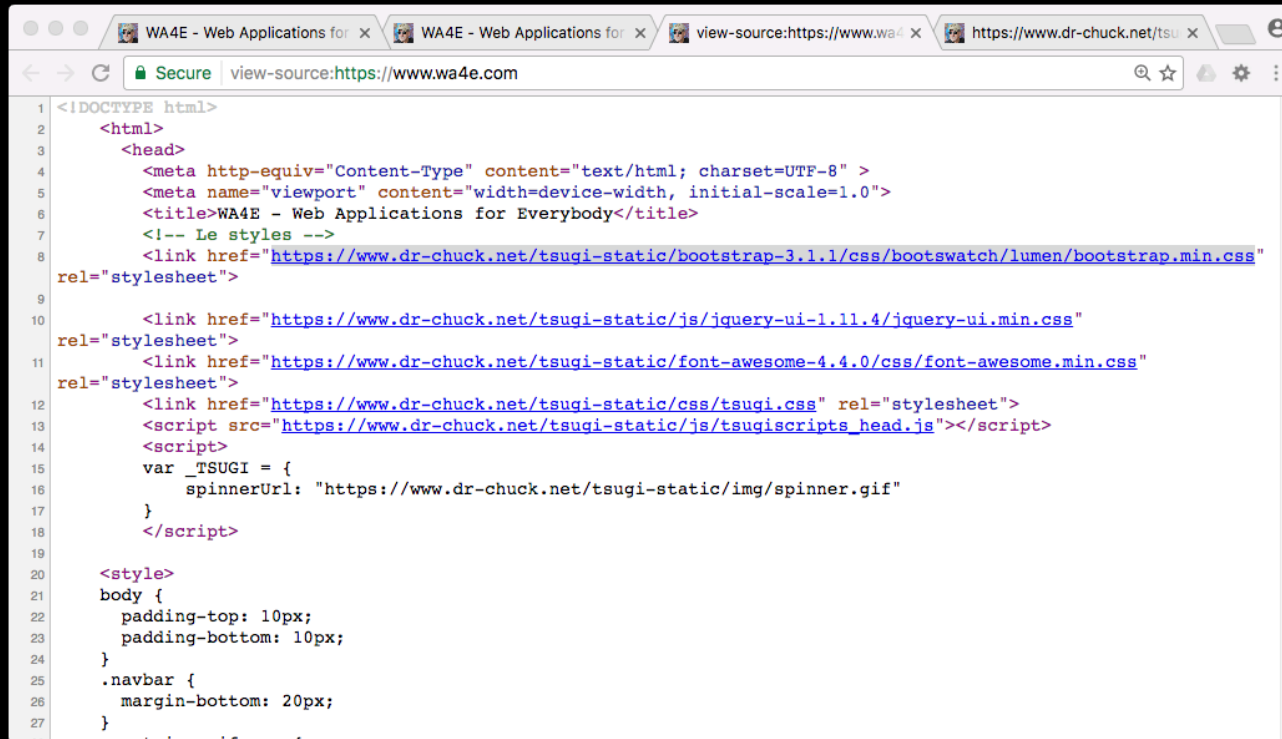


# Cascading Style Sheets



# HTML

# CSS



# Separation of Concerns / Specialization

```
<html>
<head>
  <title>Including CSS From a File</title>
  <link type="text/css" rel="stylesheet" href="rules.css">
</head>
<body>
<h1>A Header</h1>
<p>
By putting the CSS rules into a separate file,
it can be included in many different web pages
with a single "link" tag, usually in the
"head" of the document.
</p>
```

Developer

```
body {
  font-family: arial, sans-serif;
}
h1 {
  color: blue;
}
p {
  border-style: solid;
  border-color: red;
  border-width: 5px;
}
a {
  color: green;
  background-color: lightgray;
  text-decoration: none;
}
```

Designer

# CSS Syntax

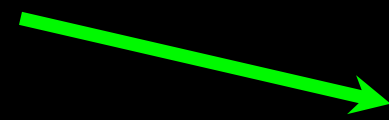
- CSS Syntax is very different than HTML.
- CSS is a set of “rules” which include a “**selector**” and one or more “**properties**” and “**values**” as well as some punctuation...

```
body {  
    font-family: arial, sans-serif;  
}
```



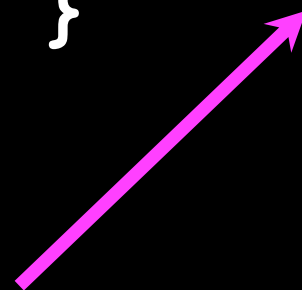
# Anatomy of a CSS Rule

selector - which part of the document this rule applies to

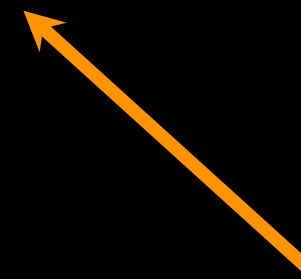


```
body {  
    font-family: arial, sans-serif;  
    font-size: 100%;  
}
```

property - which aspect of CSS we are changing



value – what we are setting the property to





CSS CHEAT SHEET

Shorthand\*

background  
border  
border-bottom  
border-left  
border-right  
border-top  
font  
list-style  
margin  
padding

Comments

/\* Comment \*/

Pseudo  
Selectors

:hover  
:active  
:focus  
:link  
:visited  
:first-line  
:first-letter

Media  
Types

all  
braille  
embossed  
handheld  
print  
projection  
screen  
speech  
tty  
tv

Units

Length %  
em  
pt

SYNTAX

**Syntax**  
selector {property: value;}  
**External Style Sheet**  
<link rel="stylesheet" type="text/css" href="style.css" />  
**Internal Style**  
<style type="text/css">  
selector {property: value;}  
</style>  
**Inline Style**  
<tag style="property: value">

GENERAL

Class	String preceded by a period
ID	String preceded by a hash mark
div	Formats structure or block of text
span	Inline formatting
color	Foreground color
cursor	Appearance of the cursor
display	block; inline; list-item; none
overflow	How content overflowing its box is handled visible, hidden, scroll, auto
visibility	visible, hidden

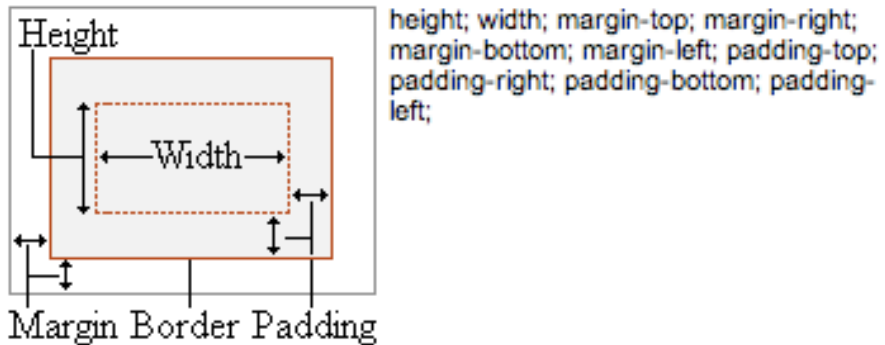
FONT

font-style	Italic, normal
font-variant	normal, small-caps
font-weight	bold, normal, lighter, bolder, integer (100-900)
font-size	Size of the font
font-family	Specific font(s) to be used

TEXT

letter-spacing	Space between letters
line-height	Vertical distance between baselines

BOX MODEL



BORDER

border-width	Width of the border
border-style	dashed; dotted; double; groove; inset; outset; ridge; solid; none
border-color	Color of the border

POSITION

clear	Any floating elements around the element? both, left, right, none
float	Floats to a specified side left, right, none
left	The left position of an element auto, length values (pt, in, cm, px)
top	The top position of an element auto, length values (pt, in, cm, px)
position	static, relative, absolute
z-index	Element above or below overlapping elements? auto, integer (higher numbers on top)

BACKGROUND

background-color	Background color
background-image	Background image
background-repeat	

# Partial List of CSS Properties

<code>color</code>	<code>text-decoration</code>
<code>background-color</code>	<code>border-width</code>
<code>visibility (visible/hidden)</code>	<code>border-style</code>
<code>font-family (arial, sans-serif)</code>	<code>border-color</code>
<code>font-size</code>	<code>margin</code>
<code>font-style (italic, normal)</code>	<code>border</code>
<code>font-weight (bold, normal)</code>	<code>padding</code>
<code>text-align</code>	<code>float (left, right, none)</code>
<code>vertical-align</code>	<code>left / top</code>
<code>text-transform (lowercase, etc)</code>	<code>position (static, relative, absolute)</code>
	<code>z-index</code>

<http://www.lesliefranke.com/files/reference/csscheatsheet.html>



# Styling with CSS



# Applying CSS to our HTML

- **Inline** - right on an HTML tag, using the style= attribute
- An **embedded style sheet** in the <head> of the document
- As an **external style sheet** in a separate file



And on to CSS

www.wa4e.com/code/css/

# An Overview of CSS

These pages are intended to be explored in the developer console of your browser.

We can use the **style=** attribute on HTML tags to apply various styles to text and blocks of text.

The "Cascading" in CSS means that the closest style setting (monospace) "wins" over a syle setting (arial) that is "further away".

We can put a border around a block of text. You probably want a CSS cheat sheet handy or use Google to search to find CSS values.

While the **style** attribute *works*, it leads to overly verbose HTML so it is more typical to write CSS [⇒ style rules](#) to apply styles to many tags at once. [Go to the very last page.](#)

Elements Console Sources Network Timeline Profiles Application Security Audits

```
<html>
  <head>...</head>
  <body style="font-family: arial, sans-serif">
    <h1 style="color: blue;">An Overview of CSS</h1>
    <p>...</p>
    <p>...</p>
    <p></p>
    <p style="font-family: monospace;">...</p>
    <p style="border-style: solid; border-color: red; border-width: 5px;"> == $0
      "
      We can put a border around a block of text.
      You probably want a CSS cheat sheet handy
      or use Google to search to find CSS values.
      "
    </p>
    <p>...</p>
    <p style="margin-top: 5em;">...</p>
  </body>
</html>
```

Styles Computed Event Listeners DOM Breakpoints Properties

Filter :hov .cls +

element.style {  
 border-style: solid;  
 border-color: red;  
 border-width: 5px;  
}

p {  
 display: block;  
 -webkit-margin-before: 1em;  
 -webkit-margin-after: 1em;  
 -webkit-margin-start: 0px;  
 -webkit-margin-end: 0px;  
}

Inherited from body

Style Attribute {



CSS Rules

www.wa4e.com/code/css/rules.htm

A Header

You put a set of rules in a <style> tag, and they are applied to parts of the documents that match the selectors.

After a while we end up with far too many CSS rules to put in the <head> area of each document. A simpler way to include a large number of CSS rules is to put them in a separate file and ⇒ include that file in each of the pages.

Go back to the [Starting page](#).

Elements Console Sources Network Timeline Profiles Application Security Audits

<head>

<title>CSS Rules</title>

<style>

body {

font-family: arial, sans-serif;

}

h1 {

color: blue;

}

p {

border-style: solid;

border-color: red;

border-width: 5px;

}

a {

color: green;

text-decoration: none;

background-color: lightgray;

}

</style>

</head>

<body>

<h1>A Header</h1>

Styles Computed Event Listeners DOM Breakpoints Properties

Filter

element.style {

}

title {

display: none;

}

user agent stylesheet

margin

border

padding

auto x auto

html head title

```
<html>
<head>
  <title>Including CSS From a File</title>
  <link type="text/css" rel="stylesheet" href="rules.css">
</head>
<body>
<h1>A Header</h1>
```

```
csev $ ls -l
total 32
-rw-r--r-- 1 csev staff  44 Dec 19 06:06 rules.css
-rw-r--r-- 1 csev staff 679 Dec 19 06:07 index.htm
-rw-r--r-- 1 csev staff 883 Dec 19 05:59 include.htm
-rw-r--r-- 1 csev staff 679 Dec 19 05:59 colors.htm
csev $
```





Including CSS From a File

www.wa4e.com/code/css/rules

www.wa4e.com/code/css/include.htm

A Header

By putting the CSS rules into a separate file, it can be included many different web pages with a single "link" tag, usually in the "head" of the document.

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

Now that we have seen the three ways to indicate CSS styles in your HTML, we will look some HTML tags whose sole purpose is to **mark text** so we can apply styling to the text or block.

Go back to the **Starting page**.

Elements Console Sources Network Timeline Profiles Application Security Audits

<html>  
<head>  
 <title>Including CSS From a File</title>  
 <link type="text/css" rel="stylesheet" href="rules.css">  
</head>  
<body>  
 <h1>A Header</h1>  
 <p>...</p>  
 <pre> <link type="text/css" rel="stylesheet" href="rules.css">  
 </pre>  
 <p>...</p>  
 <!-- Ignore this for now :) -->  
 <p style="border-style: none; position: fixed; bottom: 10px; right: 10px;">...</p>  
</body>  
</html>

Styles

Computed

Event Listeners

DOM Breakpoints

Properties

Filter :hov .cls +

element.style {  
}  
  
h1 { rules.css:4  
 color: blue;  
}  
  
h1 { user agent stylesheet  
 display: block;  
 font-size: 2em;  
 -webkit-margin-before: 0.67em;  
 -webkit-margin-after: 0.67em;  
 -webkit-margin-start: 0px;  
 -webkit-margin-end: 0px;  
 font-weight: bold;  
}  
  
Inherited from body

html body h1

```
body {  
    font-family: arial, sans-serif;  
}  
h1 {  
    color: blue;  
}  
p {  
    border-style: solid;  
    border-color: red;  
    border-width: 5px;  
}  
a {  
    color: green;  
    background-color: lightgray;  
    text-decoration: none;  
}
```

# span and div Tags

```
<p style="border: 1px green solid;">
```

With CSS we wanted some tags that had no pre-existing **style**. So the **<span style="color: green;">span</span>** tag was invented as the new "inline" tag with no styling.

```
</p>
```

```
<div style="border: 1px blue solid;">
```

And the **<strong>div</strong>** tag is a new unstyled block tag with no padding, margin, background-color, or anything else. So you could **mark** blocks with the **div** tag and not inherit any default **style**.

```
<div style="border: 1px orange solid;">
```

And the **<strong>div</strong>** tags can be nested as well. Adding the 1-pixel borders does take up a pixel of space.

```
</div>
```

You can add some **text** in the outer **div**.

```
</div>
```

```
<p style="border: 1px green solid;">
```

With CSS we wanted some tags that had no pre-existing **Style**. So the `<span style="color: green;">span</span>` tag was invented as the new "inline" tag with no styling.

```
</p>
```

```
<div style="border: 1px blue solid;">
```

And the `<strong>div</strong>` tag is a new unstyled block tag with no padding, margin, background-color, or anything else. So you could **mark** blocks with the **div** tag and not inherit any default **style**.

```
<div style="border: 1px orange solid;">
```

And the `<strong>div</strong>` tags can be nested as well. Adding the 1-pixel borders does take up a pixel of space.

```
</div>
```

You can add some **text** in the outer **div**.

```
</div>
```

With CSS we wanted some tags that had no pre-existing style. So the **span** tag was invented as the new "inline" tag with no styling.

And the **div** tag is a new unstyled block tag with no padding, margin, background-color, or anything else. So you could mark blocks with the div tag and not inherit any default style.

And the **div** tags can be nested as well. Adding the 1 pixel borders does take up a pixel of space.

You can add some text in the outer div.

Wow - there really is no margin or padding on an unstyled **div** tag.



```
1 <html>
2 <head>
3   <title>Gettin' Classy</title>
4   <style>
5     /* I wish this were the default */
6     body {
7       font-family: arial, sans-serif;
8     }
9     #first {
10      font-family: monospace;
11    }
12    #second {
13      color: green;
14    }
15    .morespace {
16      margin-left: 20px;
17      margin-right: 20px;
18    }
19    .shout {
20      text-transform: uppercase;
21    }
22    .loud {
23      color: red;
24    }
25    /* Paragraph tags within a tag with id */
26    #third p {
27      background-color: yellow;
28    }
29  </style>
30 </head>
31 <body>
32 <h1>Selecting and Styling Tags</h1>
33 <div id="first">
34 <p>
35 To avoid putting too much into the
36 <strong>style</strong> attribute,
37 we can use the <strong>class</strong> and
```

```
31 <body>
32 <h1>Selecting and Styling Tags</h1>
33 <div id="first">
34 <p>
35 To avoid putting too much into the
36 <strong>style</strong> attribute,
37 we can use the <strong>class</strong> and
38 <strong>id</strong> attributes to style
39 a subset of the tags.
40 </p>
41 <p class="morespace">
42 An <strong>id</strong> tag must be unique
43 through the document while the
44 <strong>class</strong> tag can be used on
45 many tags throughout the document.
46 </div>
47 <div id="second">
48 <p>
49 We can use the <strong>class</strong>
50 <span class="shout">all</span> throughout
51 the <span class="shout">entire</span>
52 document.
53 </p>
54 <p class="morespace">
55 It is <strong class="shout">very cool</strong>
56 to use classes. The can be used on any tag.
57 </p>
58 <p class="shout loud">
59 A tag can have more than one class!.
60 </p>
61 </div>
62 <div id="third">
63 <p>
64 You can even use a hierarchical selection string
65 to style tags.
66 </p>
67 <p>
```

```
1 <html>
2 <head>
3   <title>Gettin' Classy</title>
4   <style>
5     /* I wish this were the default */
6     body {
7       font-family: arial, sans-serif;
8     }
9     #first {
10      font-family: monospace;
11    }
12    #second {
13      color: green;
14    }
15    .morespace {
16      margin-left: 20px;
17      margin-right: 20px;
18    }
19    .shout {
20      text-transform: uppercase;
21    }
22    .loud {
23      color: red;
24    }
25    /* Paragraph tags within a tag with id */
26    #third p {
27      background-color: yellow;
28    }
29  </style>
30 </head>
31 <body>
32 <h1>Selecting and Styling Tags</h1>
33 <div id="first">
```

## Selecting and Styling Tags

To avoid putting too much into the **style** attribute, we can use the **class** and **id** attributes to style a subset of the tags.

An **id** tag must be unique through the document while the **class** tag can be used on many tags throughout the document.

We can use the **class** ALL throughout the ENTIRE document.

It is **VERY COOL** to use classes. The can be used on any tag.

**A TAG CAN HAVE MORE THAN ONE CLASS!**

You can even use a hierarchical selection string to style tags.

In general it is nice to keep your HTML as clean and easily understood as possible as in this [⇒ simple navigation bar example](#).

Go back to the [Starting page](#).

```
40 </p>
41 <p class="morespace">
42 An <strong>id</strong> tag must be unique
43 through the document while the
44 <strong>class</strong> tag can be used on
45 many tags throughout the document.
46 </div>
47 <div id="second">
48 <p>
49 We can use the <strong>class</strong>
50 <span class="shout">all</span> throughout
51 the <span class="shout">entire</span>
52 document.
53 </p>
54 <p class="morespace">
55 It is <strong class="shout">very cool</strong>
56 to use classes. The can be used on any tag.
57 </p>
58 <p class="shout loud">
59 A tag can have more than one class!.
60 </p>
61 </div>
62 <div id="third">
63 <p>
64 You can even use a hierarchical selection string
65 to style tags.
66 </p>
67 <p>
68 In general it is nice to keep your HTML as clean
69 and easily understood as possible as in this
70 <a href="navbar.htm">
71 &rArr; simple navigation bar
72 </a> example.
73 </p>
```



# Images, Colors, and Links




Images Page x view-source:www.wa4e.com/...

www.wa4e.com/code/css/images.htm



← Styled Nav Bar Cascading Styles ⇒

## CSS and Images

It is fun to float an image to the left or right and watch the text wrap around it. You might want to add a bit of margin so the text does not crowd it. An `em` is the "height of the current font" so it scales when the size of the font is changed.



Sometimes we need to clear the "hanging wrap" when we want the next paragraph to go all the way to the margin - in effect "clearing the float".

Images can be  right in the middle of text like a character. And we can even make an image a clickable  link to the starting page.

Images Page x view-source:www.wa4e.com/...

view-source:www.wa4e.com/code/css/images.htm

```
</a></li>
11 </ul>
12 </nav>
13 
14 <h1>CSS and Images</h2>
15 <p>
16 It is fun to float an image to the left or
17 right and watch the text wrap around it.
18 You might want to add a bit of margin so the text
19 does not crowd it. An <strong>em</strong> is the
20 "height of the current font" so it scales when the size
21 of the font is changed.
22 </p>
23 <br clear="all"/>
24 <p>
25 Sometimes we need to clear the "hanging wrap"
26 when we want the next paragraph to go all the way
27 to the margin - in effect "clearing the float".
28 </p>
29 <p>
30 Images can be  right
31 in the middle of text like a character.
32 And we can even make an image a clickable
33 <a href="index.htm"></a>
34 link to the starting page.
35 </p>
36 </body>
37
```

# Color Names

- W3C has listed 16 official color names that will validate with an HTML validator.
- The color names are: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.

## Color Values

HTML colors can be defined as a hexadecimal notation for the combination of Red, Green, and Blue color values (RGB).

The lowest value that can be given to one light source is 0 (hex #00) and the highest value is 255 (hex #FF).

The table below shows the result of combining Red, Green, and Blue light sources:.

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

## W3C Standard Color Names

W3C has listed 16 color names that will validate with an HTML validator.

The color names are: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.

[http://www.w3schools.com/html/html\\_colors.asp](http://www.w3schools.com/html/html_colors.asp)



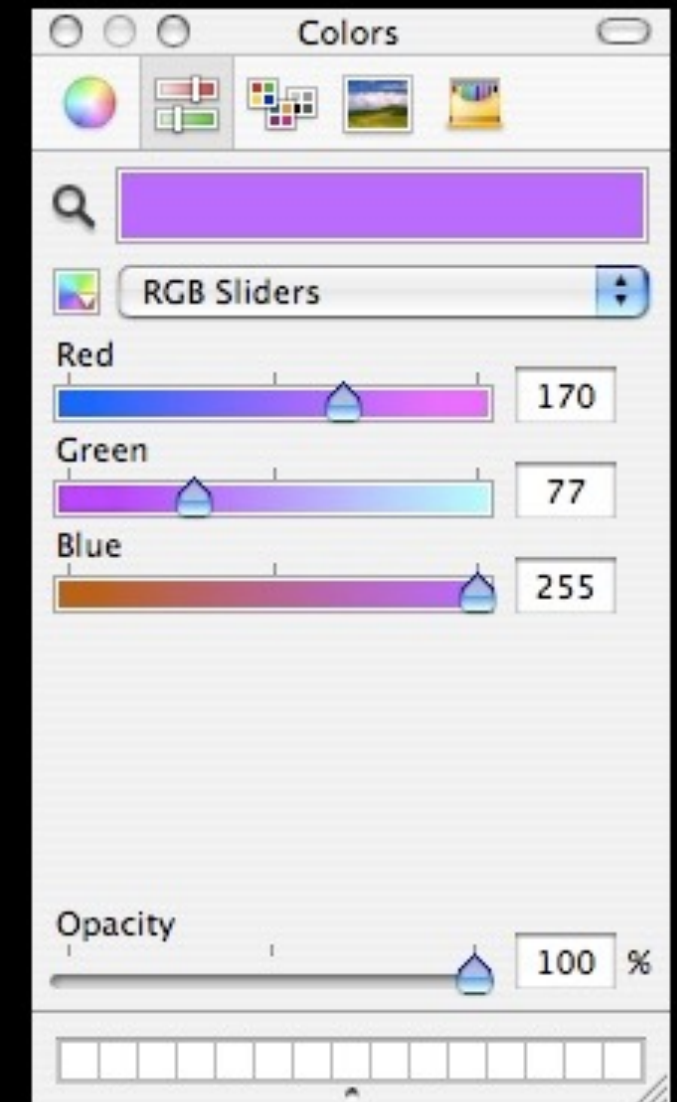
# Advanced Colors...

Three numbers,  
Red, Green, and  
Blue - each from  
00 - FF  
(Hexidecimal)

#ffffff = white  
#000000 = black  
#ff0000 = red  
#00ff00 = green  
#0000ff = blue

#e2edff

Color Name	Color HEX	Color
<a href="#">AliceBlue</a>	<a href="#">#F0F8FF</a>	
<a href="#">AntiqueWhite</a>	<a href="#">#FAEBD7</a>	
<a href="#">Aqua</a>	<a href="#">#00FFFF</a>	
<a href="#">Aquamarine</a>	<a href="#">#7FFFD4</a>	
<a href="#">Azure</a>	<a href="#">#F0FFFF</a>	
<a href="#">Beige</a>	<a href="#">#F5F5DC</a>	
<a href="#">Bisque</a>	<a href="#">#FFE4C4</a>	
<a href="#">Black</a>	<a href="#">#000000</a>	
<a href="#">BlanchedAlmond</a>	<a href="#">#FFEBCD</a>	
<a href="#">Blue</a>	<a href="#">#0000FF</a>	
<a href="#">BlueViolet</a>	<a href="#">#8A2BE2</a>	
<a href="#">Brown</a>	<a href="#">#A52A2A</a>	



Web-safe  
colors

[http://www.w3schools.com/css/css\\_colornames.asp](http://www.w3schools.com/css/css_colornames.asp)

# Fonts

- Default fonts are ugly and they have serifs - which make them harder to read on a screen
- So the first thing I usually want to do is override the fonts in my document
- And I want to do this everywhere

**Header One**

Paragraph about ugly default fonts.

Figure 3.5. Highlighting the serifs of a serif font (Georgia)




# Fonts

Most Favorite

Least Favorite

```
body {  
  font-family: "Trebuchet MS", Helvetica, Arial, sans-serif;  
  font-size: x-large;  
}
```



Fallback fonts: serif, sans-serif, monospace, cursive, and fantasy

# Font Factors

**font-size:**

xx-small

x-small

small

medium

large

x-large

xx-large

14px

**font-weight:** bold or normal

**font-style:** normal or italic

**text-decoration:** none, underline, overline, or  
line-through

# Styling for Links



Post-Click:

Hello there my name is Chuck.  
Go ahead and click on [here](#).

Browser default styling for  
links is downright ugly!



```
a {  
  font-weight: bold;  
}  
a:link {  
  color: black;  
}  
a:visited {  
  color: gray;  
}  
a:hover {  
  text-decoration: none;  
  color: white;  
  background-color: navy;  
}  
a:active {  
  color: aqua;  
  background-color: navy;  
}
```

# Styling Links

link - before a visit

visited - after it has been visited

hover - when your mouse is over it  
but you have not clicked

active - you have clicked it and you  
have not yet seen the new page

# Many More Samples

wa4e.com

The screenshot shows a web browser window with the address bar displaying `www.wa4e.com/code/css/inherit.htm`. The page title is `*Cascading*`. The page content is titled "Putting the 'Cascade' in CSS" and contains three paragraphs illustrating CSS inheritance and the `!important` rule:

- Order matters. The CSS rule that is "closest" to the tag wins unless a "further away" tag (i.e. border-color) marks itself as **important**.
- Even using the **style** attribute *right on the tag* fades in the face of the **!important** tag.
- But you can always take local control using **!important** "closer" to the tag.

The browser's developer tools are open, showing the `Elements` panel on the left and the `Styles` panel on the right. The `Elements` panel shows the following HTML structure:

```
<html>
  <head>...</head>
  <body>
    <nav>...</nav>
    <h1>Putting the "Cascade" in CSS</h1>
    <p>...</p>
    <p style="border-style: dashed; border-color: blue;">...</p>
    <p style="border-style: dashed; border-color: blue !important;">
      "
      But you can always take local control using
      "
      <strong>!important</strong>
      " "closer" to the tag.
    </p>
  </body>
</html>
```

The `Styles` panel shows the following CSS rules:

```
element.style {
  border-style: dashed;
  border-color: blue !important;
}

p {
  border-style: solid;
  border-color: green;
}

p {
  border-style: dotted;
  border-color: red !important;
  border-width: 5px;
}
```

# CSS Summary

- CSS layout is its own art and science.
- CSS basics are well established and well supported in all modern browsers.
- Site layout and markup is further evolving - mostly to make it increasingly possible to support desktop-like experiences on the web and mobile.
- These innovations will naturally cause incompatibilities - which make things interesting and frustrating at times.

# Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance ([www.dr-chuck.com](http://www.dr-chuck.com)) as part of [www.wa4e.com](http://www.wa4e.com) and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information

Insert new Contributors and Translators here including names and dates

Continue new Contributors and Translators [here](#)