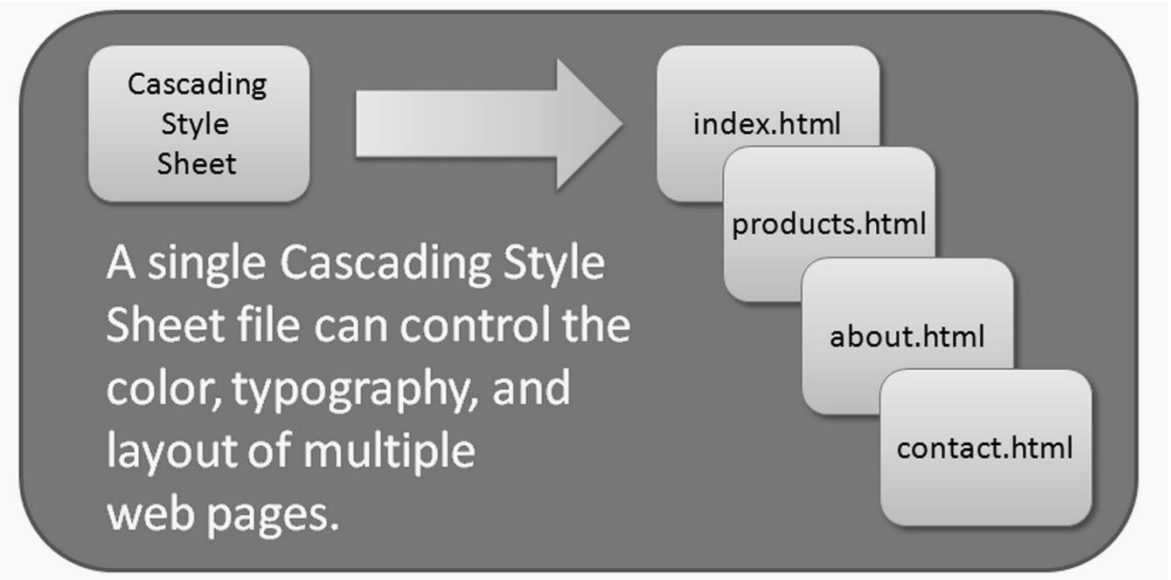
CSS3
Basics

Overview of CSS

- Cascading Style Sheets (CSS)
  - provides the functionality of style sheets *(and much more)* for web developers
  - a flexible, cross-platform, standards-based language developed by the W3C

- See what is possible with CSS:
  - Visit http://www.csszengarden.com

CSS Advantages

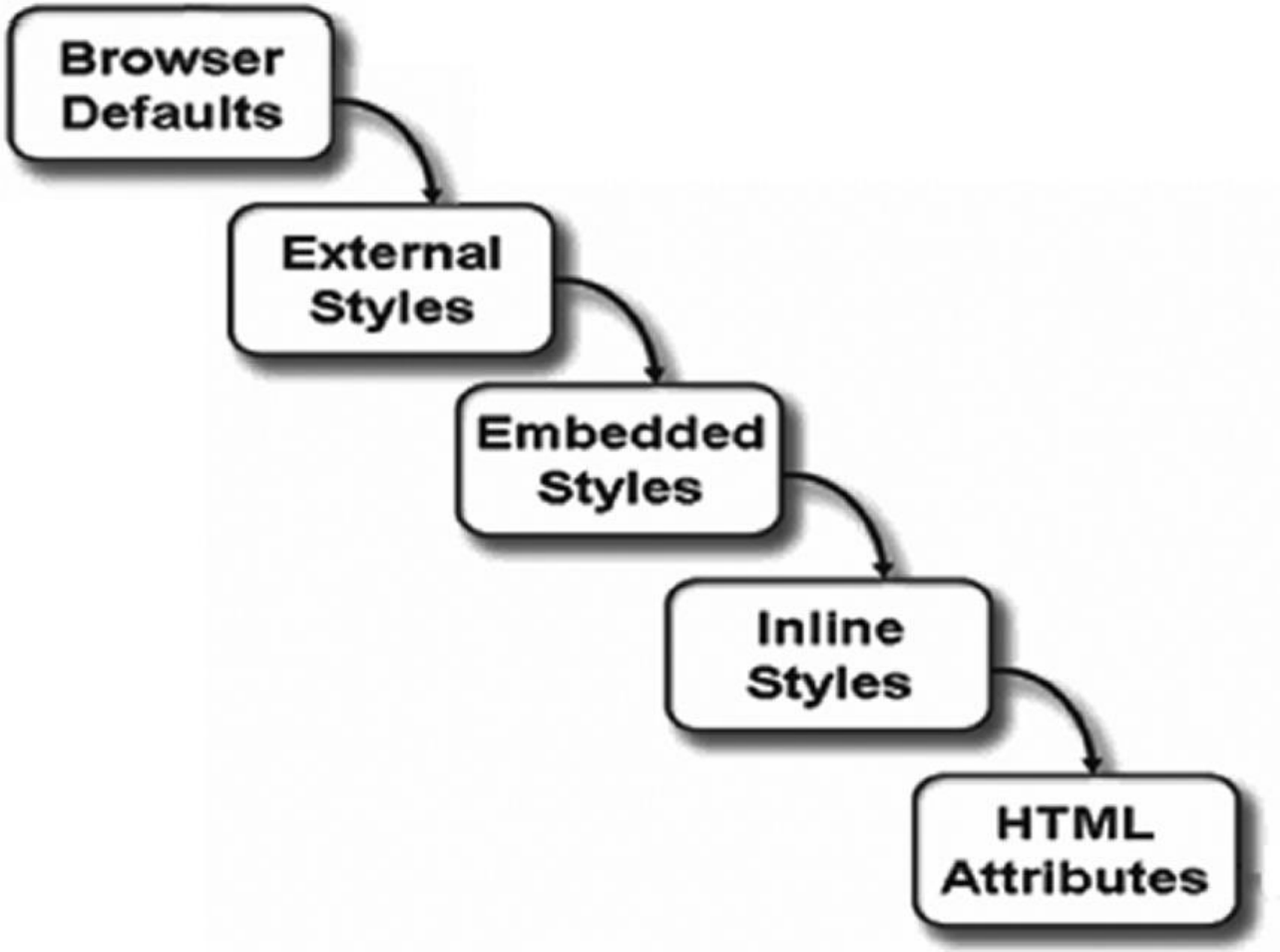


- Greater typography and page layout control
- Style is separate from structure
- Styles can be stored in a separate document and associated with the web page
- Potentially smaller documents
- Easier site maintenance

4 Places for CSS in a HTML Document

- Inline Styles
  - body section
  - HTML style attribute
  - apply only to the specific element
- Embedded Styles
  - head section
  - HTML style element
  - apply to the entire web page document
- External Styles
  - Separate text file with .css file extension
  - Associate with a HTML link element in the head section of a web page
- Imported Styles
  - Similar to External Styles
  - We'll concentrate on the other three types of styles

The “Cascade”


Browser Defaults
External Styles
Embedded Styles
Inline Styles
HTML Attributes

CSS Syntax

- Style sheets are composed of "Rules" that describe the styling to be applied
- Each Rule contains a Selector and a Declaration

CSS Syntax Sample

- Configure a web page to display blue text and yellow background

```
body { color:  blue;
       background-color:  yellow; }
```

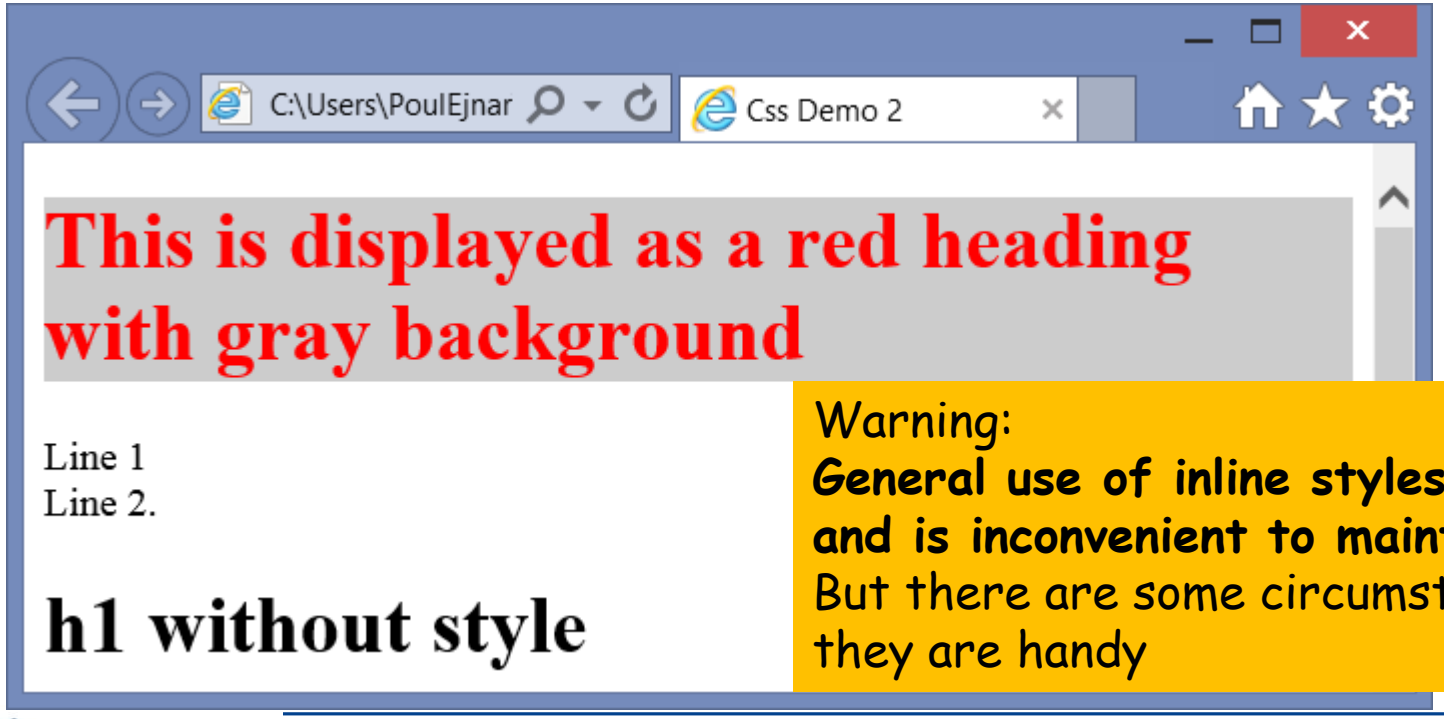
- This could also be written using hexadecimal color values as shown below

```
body { color:  #0000FF;
       background-color:  #FFFF00; }
```

Inline CSS

- Example:
  - configure the red text in the heading
  - configure a gray background in the heading

```
<h1 style="color:#FF0000;background-color:#cccccc">This is displayed as a red heading with gray background</h1>
```



Warning:
General use of inline styles is inefficient and is inconvenient to maintain!
But there are some circumstances where they are handy

Embedded CSS

- Configured in the header section of a web page
- Use the HTML <style> element
- Apply to the entire web page document
- Style declarations are contained between the opening and closing <style> tags
- Example:
  - Configure a web page with white text on a black background

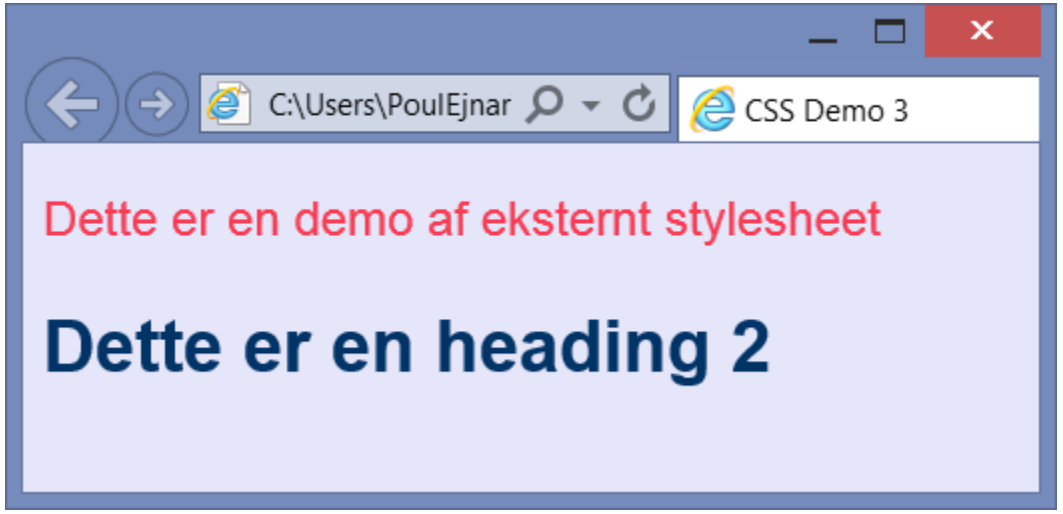
```
<head>
    <title>CSS demo</title>
    <style>
        body { background-color: #000000;
               color: #FFFFFF;
             }
    </style>
</head>
```

External Style Sheets

- CSS style rules are contained in a text file separate from the HTML documents
- The External Style Sheet text file:
  - extension ".css"
  - contains only style rules
- A HTML link Element is used to associates the external style sheet file with the web page
- Multiple web pages can associate with the same external style sheet file

External Style Sheet Example

```
<head>
    <title>CSS demo</title>
    <link rel="stylesheet" href="site.css">
</head>
<body>
    <p>Dette er en demo af eksternt stylesheet</p>
    <h2>Dette er en heading 2</h2>
</body>
```



site.css

```
body {background-color:#E6E6FA;
      color:#f93c53;
      font-family:Arial, sans-serif;
      font-size:120%; }
h2 {  color: #003366; }
```

Common Formatting Properties

- background-color
- color
- font-family
- font-size
- font-style
- font-weight
- line-height
- margin
- text-align
- text-decoration
- width

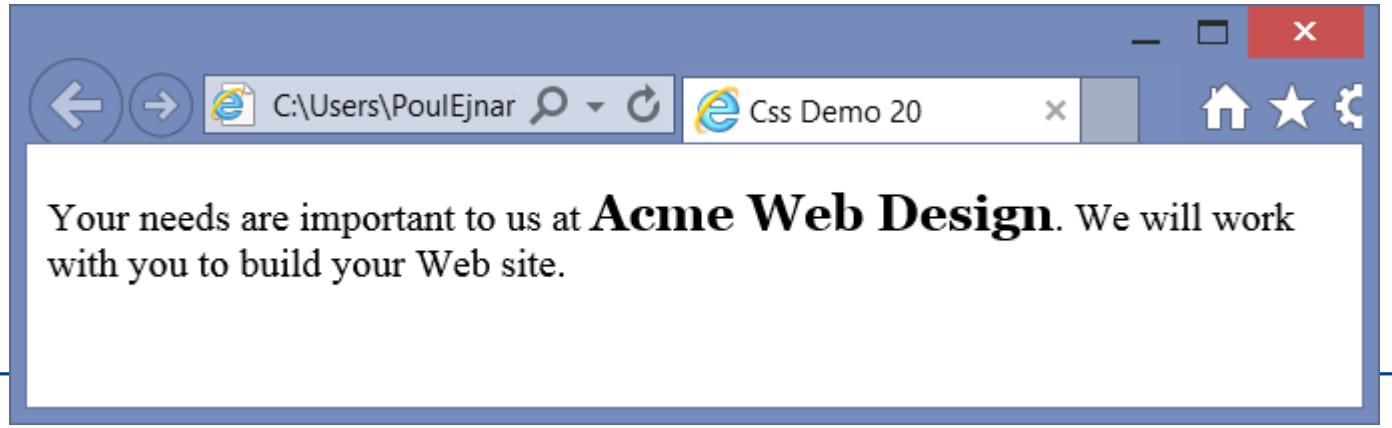
span Element Example

- CSS:

```
<style>
.companyname { font-weight: bold;
               font-family: Georgia, "Times New Roman",
                            serif;
               font-size: 1.25em;
             }
 </style>
```

- HTML:

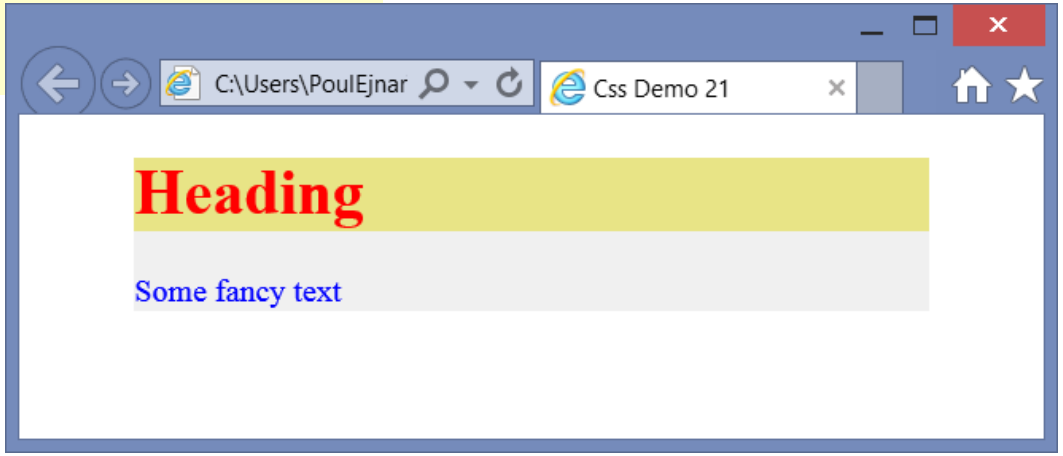
```
<p>Your needs are important to us at <span
class="companyname">Acme Web Design</span>.
We will work with you to build your Web site.</p>
```



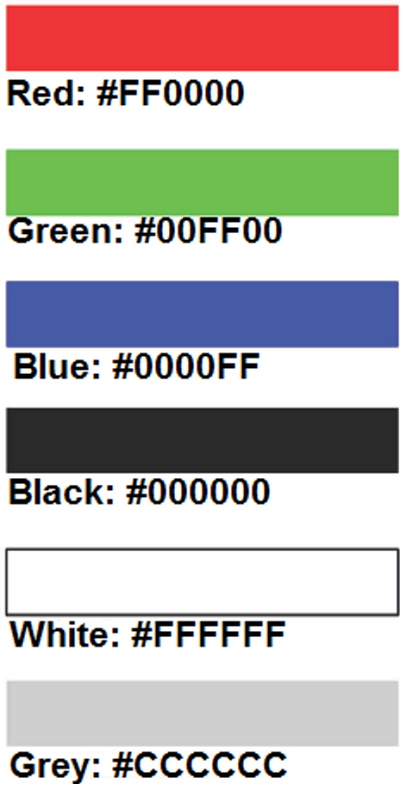
Centering Page Content with CSS and div

```
#wrapper { margin-left: auto;
           margin-right: auto;
           width:80%;
           color: blue;
           background-color: #F0F0F0;}
```

```
<body>
    <div id="wrapper">
        <h1>Heading</h1>
        <p>Some fancy text</p>
    </div>
</body>
```



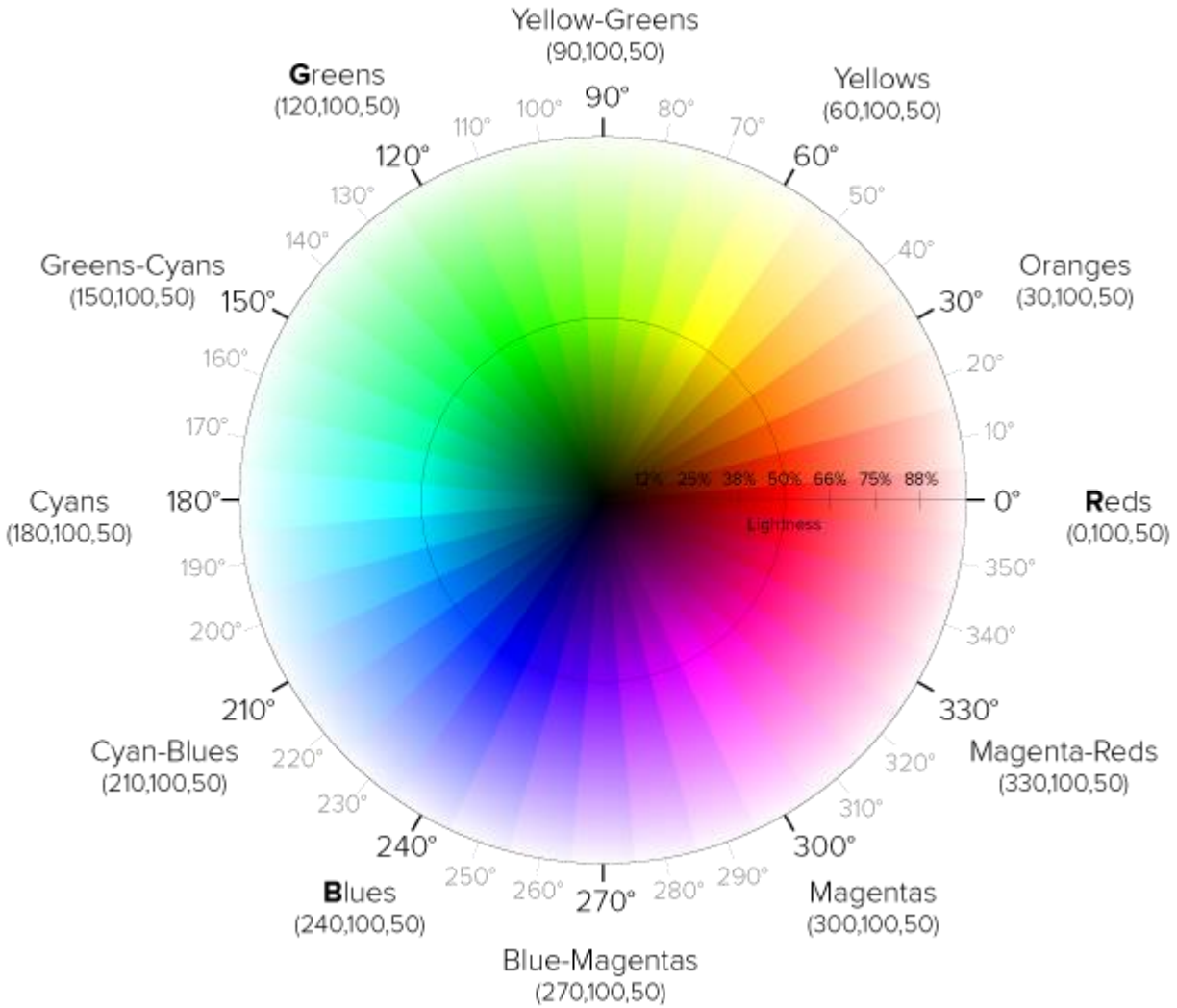
Using Color on Web Pages



- Computer monitors display color as intensities of red, green, and blue light
- The values of red, green, and blue vary from 0 to 255
  - **But often expressed in hexadecimal numbers**
  - # is used to indicate a hexadecimal value
- There are several alternatives for specifying a color:
  - Rgb in hex or decimal
  - Color name http://www.w3schools.com/colors/colors_names.asp
  - Hsl (only in decimal)

HSL and HSLA

- HSL stands for hue, saturation, and lightness



- **Hue**
  Is a degree on the color wheel from 0 to 360
  0 is red
  120 is green
  240 is blue

- **Saturation**
  Is a percentage value
  0% means a shade of gray
  100% is the full color

- **Lightness**
  Is a percentage value
  0% is black
  100% is white

Different Options for Color

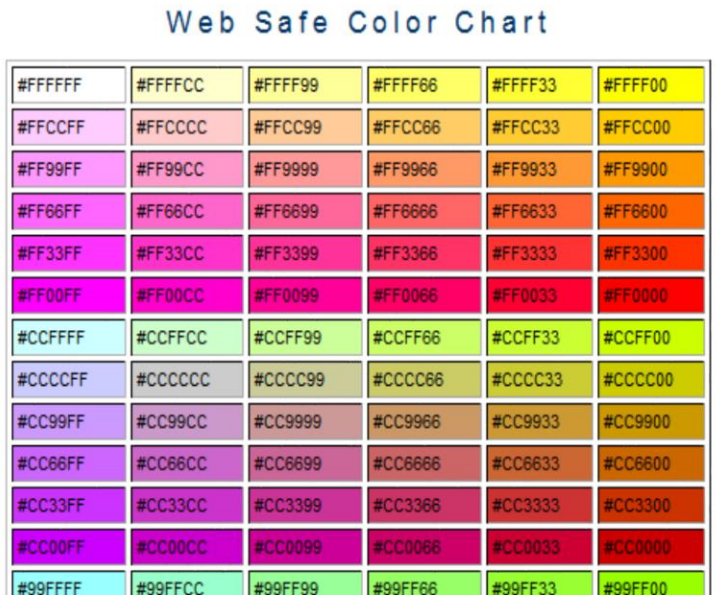
```
.color1 {
    background-color: maroon;
    color: white;
}
.color2 {
    background-color: #800;
    color:#FFF;
}
.color3 {
    background-color: #800000;
    color: #FFFFFF;
}
.color4 {
    background-color: rgb(128,0,0);
    color: rgb(255,255,255);
}
```

```
.color5 {
    background-color: rgba(128,0,0,1.0);
    color: rgba(255,255,255,1.0);
}
.color6 {
    background-color: hsl(0,100%,13%);
    color:hsl(0,100%,100%);
}
.color7 {
    background-color:
                    hsla(0,100%,13%,1.0);
    color: hsla(0,100%,100%,1.0);
}
```

1. Color name
2. Shorthand hexadecimal
3. Hexadecimal color value
4. RGB Decimal color value
5. RGB Decimal color value with transparency
6. HSL Decimal color value
7. HSL Decimal color value with transparency

Making Color Choices

- How to choose a color scheme?
  - Monochromatic
    - http://meyerweb.com/eric/tools/color-blend
  - Choose from a photograph or other image
    - http://www.colr.org
  - Begin with a favorite color
    - Use one of the sites below to choose other colors
    - http://colorsontheweb.com/colorwizard.asp
    - http://kuler.Adobe.com
    - http://colorschemedesigner.com/
  - Web Color Palette
    - http://webdevfoundations.net/color



Configuring Text with CSS

- CSS properties for configuring text:
  - font-weight
    - Configures the boldness of text
  - font-style
    - Configures text to an italic style
  - font-size
    - Configures the size of the text
  - font-family
    - Configures the font typeface of the text

The font-size Property

Absolute-size keywords	em (or ex)	rem	Percentage	px
xx-small	.5em	.5rem	50%	8px
x-small	.6em	.6rem	60%	10px
small	.75em	.75rem	75%	12px
medium	1em	1rem	100%	16px
large	1.15em	1.15rem	115%	18px
x-large	1.5em	1.5rem	150%	24px
xx-large	2em	2rem	200%	32px
xxx-large				

em/ex/percentage:
Font size is relative to parents font size.

rem:
Font size is relative to the size of the font used by the <html> (root) element.

- Accessibility Recommendation: **Use rem** (or em / percentage) font sizes
  - these can be easily enlarged in all browsers by users

https://developer.mozilla.org/en-US/docs/Web/CSS/font-size

The font-family Property

- Not everyone has the same fonts installed in their computer
- Configure a list of fonts and include a generic family name

```
p {font-family: Arial, Verdana, sans-serif;}
```

Font Family Category	Description	Common Font Typeface Names
serif	Have small embellishments (serifs) on the end of letter strokes	Times New Roman, Georgia, Palatino
sans-serif	Do not have serifs	Arial, Tahoma, Helvetica, Verdana
monospace	Fixed-width font	Couier New, Lucida Console
cursive	Handwritten style	Lucida Handwriting, Brush Script, Comic Sans MS
fantasy	Exaggerated style	Jokerman, Impact, Papyrus

Which Units To Use When

- px
  - **Use for:** hairline borders, values for CSS shadow displacement and when creating fixed-width designs
  - **Don't use for:** typography
- rem and em
  - **Use for:** typography, and elements related to typography (margins, for example)
  - prefer rem
- %
  - **Use for:** making responsive images and containers
- pt
  - **Use for:** print stylesheets
  - **Don't use for:** anything else
- cm and in
  - **Use with**: print stylesheets, especially page margins
  - **Don't use for:** anything else

CSS SELECTORS

CSS Selectors

- CSS style rules can be configured for an:
  - element selector
  - class selector
  - id selector
  - Contextual Selector

Element Selector

- The element selector selects all elements with the specified element name

```
<style>
body { color:  blue;
       background-color:  yellow;
}
</style>
```

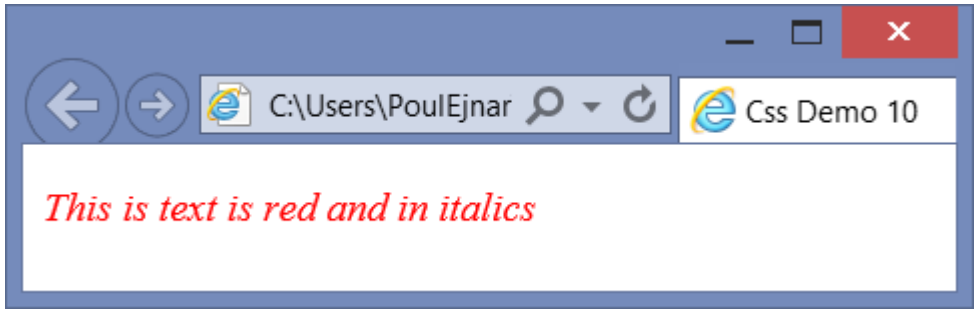
class Selector

- Apply a CSS rule to a certain "class" of elements on a web page
- Does not associate the style to a specific HTML element
- Configure with **.classname**
  - code CSS to create a class called “new” with red italic text

```
<style>
.new { color: #FF0000;
       font-style: italic;
     }
</style>
```

- Apply the class:

```
<p class="new">This is text is red and in italics</p>
```

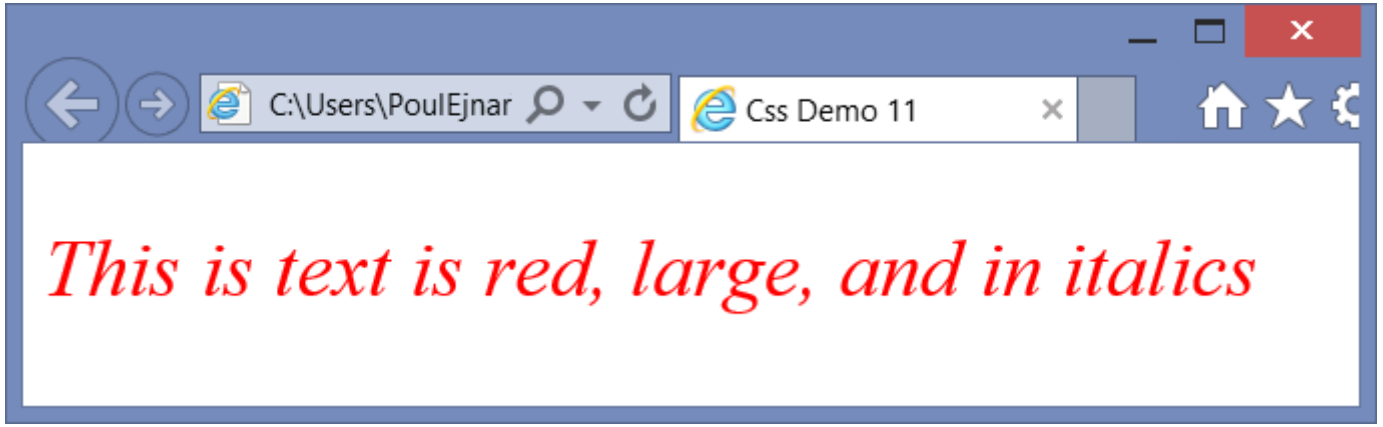


id Selector

- Apply a CSS rule to **ONE** element on a web page
- Configure with **#idname**
  - Code CSS to create an id called “new” with red, large, italic text.
- Apply the id:

```
<style>
#new { color: #FF0000;
       font-size:2em;
       font-style: italic;
     }
</style>
```

```
<p id="new">This is text is red, large, and in italics</p>
```



CSS Contextual Selector

- Specify an element within the context of its container (parent) element
  - Aka descendent selector or **relational** selectors
- The example configures a green text color only for anchor tags located within the footer id:

```
<style>
#footer a { color: #00ff00; }
</style>
```

- Advantage of contextual selectors:
  Reduce the number of classes and ids you need to apply in the HTML

Contextual Selector Overview

- **Descendant combinator (E F)**
  - Targets any element F that is a descendant (child, grandchild, great grandchild, and so on) of an element E
- **Child combinator (E > F)**
  - Matches any element F that is a *direct child* of element E
  - Any further nested elements will be ignored
- **Next sibling selector (E + F)**
  - Will match any element F that shares the same parent as E, and comes *directly after* E in the markup
- **Following sibling selector (E ~ F)**
  - Will match any element F that shares the same parent as any E and comes after it in the markup

```
main > div {
  float: left;
  overflow: hidden;
}
```

Attribute Selectors

- Allow for matching elements based on their attributes
- **`E[attr]`**
  - Matches any element `E` that has the attribute `attr` regardless of the attribute's value
- **`E[attr=val]`**
  - Matches any element E that has the attribute `attr` with the exact value val
- **`E[attr|=val]`**
  - Matches any element E whose attribute `attr` either has the value val or begins with val-

```
p[lang|="en"]
```

- **`E[attr~=val]`**
  - Matches any element E whose attribute `attr` has within its value the full word val, surrounded by whitespace

```
.info[title~=more]
```

- **`E[attr^=val]`**
  - Matches any element E whose attribute `attr` starts with the value val

Obs: There are more attribute selectors than shown here!

Pseudo-classes

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

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

Anchor Pseudo-classes

- Unvisited link
```
a:link {
    color: #FF0000;
}
```

- Visited link
```
a:visited {
    color: #00FF00;
}
```

- Mouse over link
```
a:hover {
    color: #FF00FF;
}
```

- Selected link
```
a:active {
    color: #0000FF;
}
```

Obs.:
Order matters!

Match elements based on attributes, user interaction, and form control state

- :enabled
- :disabled
- :checked
- :indeterminate
- :target
- :default
- :valid
- :invalid
- :in-range
- :out-of-range
- :required
- :optional
- :read-only
- :read-write

Structural Pseudo-classes

- Enables us to target elements based on their location in the markup
- **`:first-child`** examples
  - Match any <p> element that is the first child of any element

```
p:first-child {
    color: blue;
}
```

  - Match the first <li> element in all <ul> elements

```
ul li:first-child {
    color: blue;
}
```

Structural Pseudo-classes

- **:root**
- **E:nth-child(n)**
- **E:nth-last-child(n)**
- **E:nth-of-type(n)**
- **E:nth-last-of-type(n)**
- **E:first-child**
- **E:last-child**
- **E:only-child**
- **E:only-of-type**
- **E:empty**
- **E:not(exception)**

The *n*th child, **regardless of type**

Specify a background color for every <p> element that is the second p element of its parent:

```
p:nth-of-type(2) {
    background: #ff0000;
}
```

Odd and even are keywords that can be used to match child elements:

```
p:nth-of-type(odd) {
    background: #ff0000;
}
```

Using a formula (*an* + *b*). Description: *a* represents a cycle size, *n* is a counter (starts at 0), and *b* is an offset value:

```
p:nth-of-type(3n+1) {
    background: #ff0000;
}
```

Obs: There are more structural pseudo-classes than shown here!

Pseudo-elements

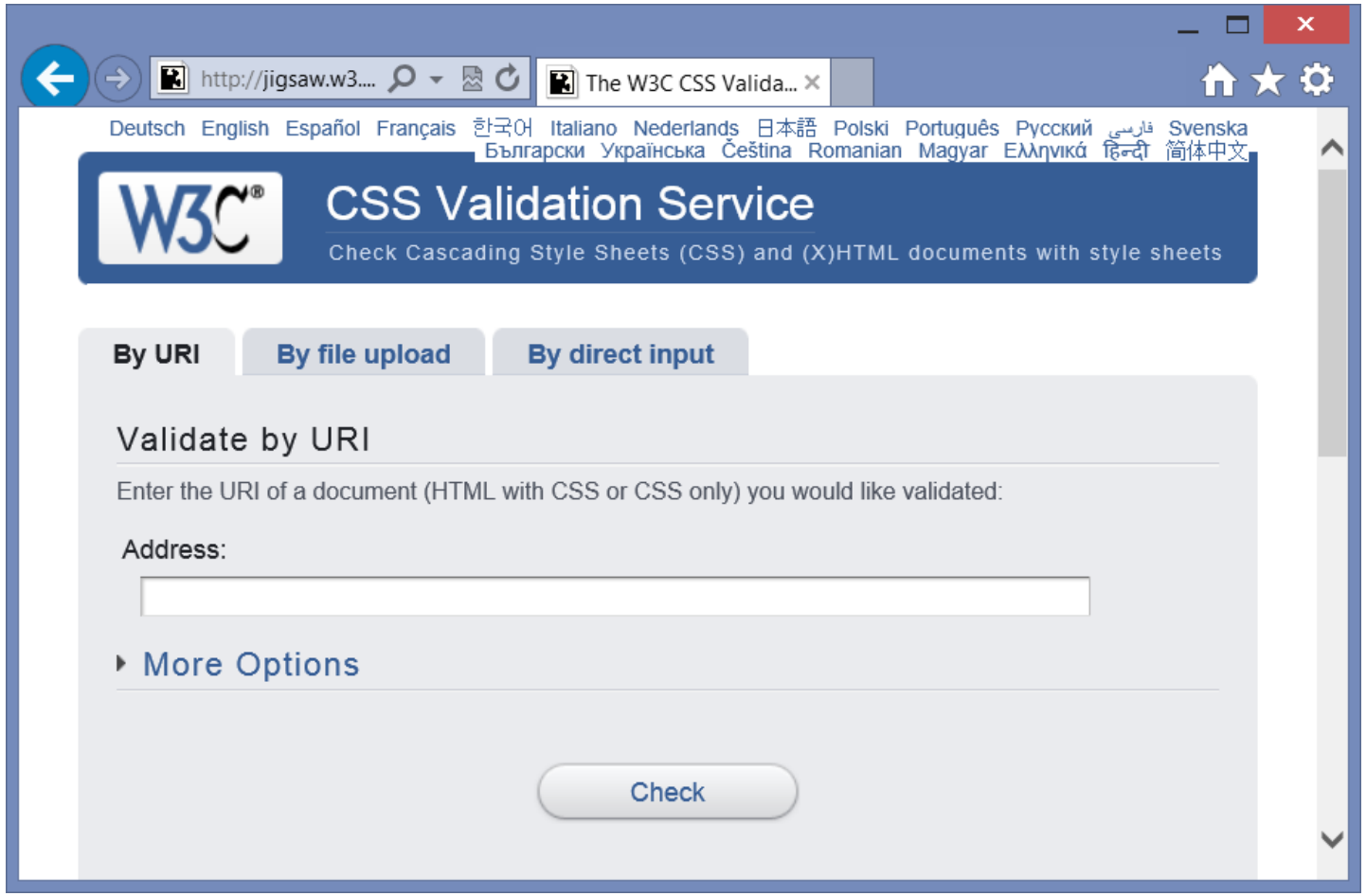
- Allow you to target text that is part of the document, but not otherwise targetable in the document tree
- ::first-letter
  - Matches the first letter of a text node
- ::first-line
  - Match the first line of a text node
- ::before
  - Inserts something before the content of each selected element(s)
- ::after
  - Inserts something after the content of each selected element(s)
- ::selection
  - Matches user-selected or highlighted text

```
p::first-letter {
    font-size: 150%;
}
```

```
p::after {
    content: " - Remember
this";
}
```

W3C CSS Validation

- http://jigsaw.w3.org/css-validator/



References & Links

- Web Development and Design Foundations with HTML5
- HTML5 & CSS3 for the Real World

- CSS3 Click Chart
  http://css3clickchart.com
- A visual CSS editor
  http://enjoycss.com/
- Testing tools for web developers from Microsoft
  http://modern.ie/en-us