Telerik Academy

CSS Overview

Cascading Style Sheets


```
#content .article img.left.border {
    padding: 0 9px 9px 0;
    border-right: 1px dotted #999;
    border-bottom: 1px dotted #999; }
#content .article blockquote {
    margin-left: 10px;
    padding-left: 10px;
    border-left: 3px solid #252525; }
#content .article ul {
    padding-left: 1em;
    list-style-type: circle; }
```

CSS Styling

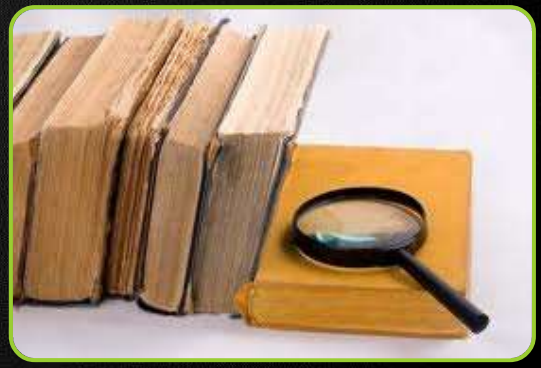
Telerik Academy

telerikacademy.com

Follow us

Telerik Academy

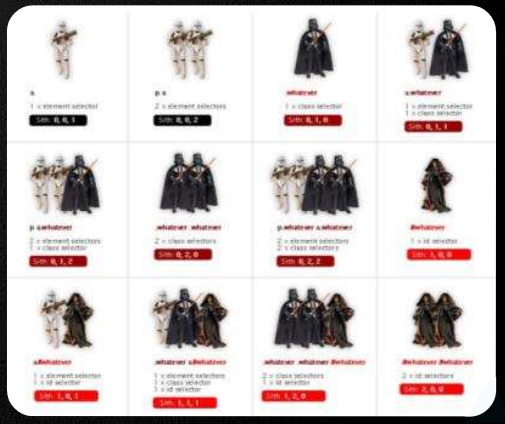
Table of Contents



Telerik Academy

Cascading Style Sheets

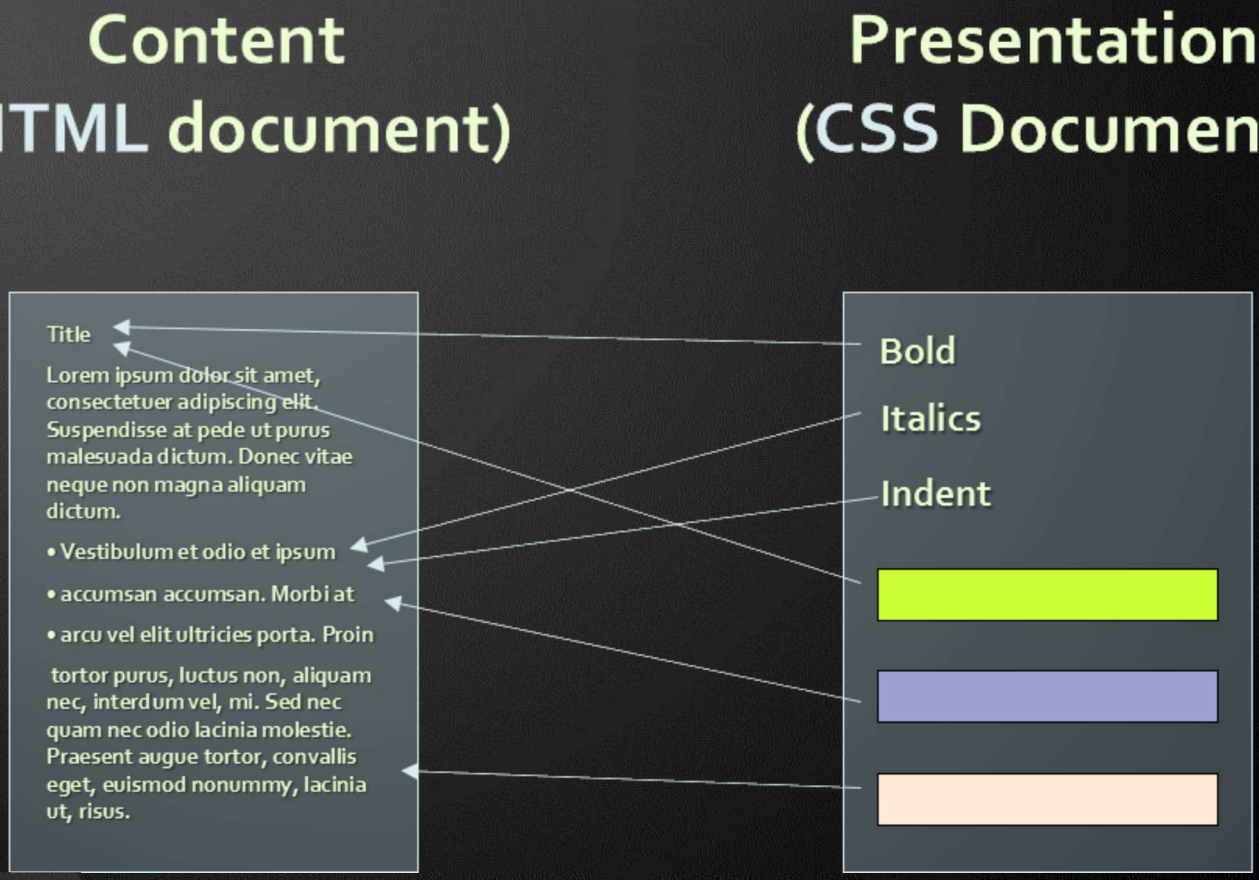
Separating Content from Presentation

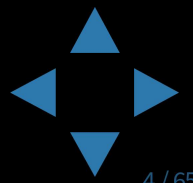


Telerik Academy

CSS: A New Philosophy

- Separate content from presentation!





Telerik Academy

The Resulting Page

Title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Suspendisse at pede ut purus malesuada dictum. Donec vitae neque non magna aliquam dictum.

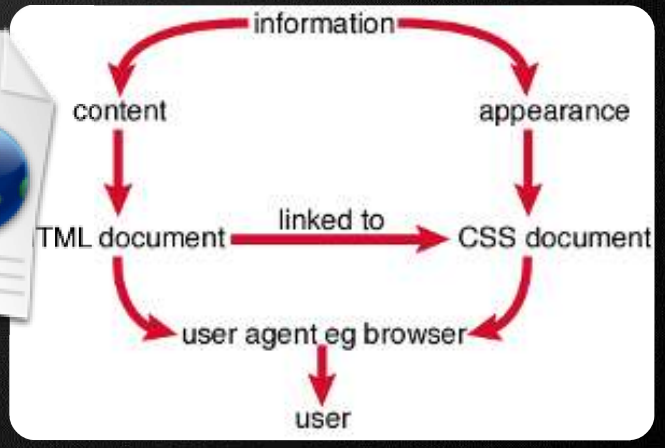
- *Vestibulum et odio et ipsum*
- *accumsan accumsan. Morbi at*
- *arcu vel elit ultricies porta. Proin*

Tortor purus, luctus non, aliquam nec, interdum vel, mi. Sed nec quam nec odio lacinia molestie. Praesent augue tortor, convallis eget, euismod nonummy, lacinia ut, risus.

Telerik Academy

CSS Intro

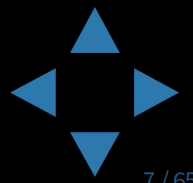
Styling with Cascading Stylesheets



Telerik Academy

CSS Introduction

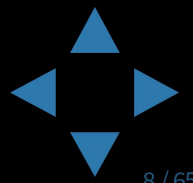
- Cascading Style Sheets (CSS)
  - Used to describe the presentation of documents
  - Define sizes, spacing, fonts, colors, layout, etc.
  - Improve content accessibility
  - Improve flexibility
- Designed to separate presentation from content
- Due to CSS, all HTML presentation tags and attributes are deprecated, e.g. font, center, etc.



Telerik Academy

CSS Introduction

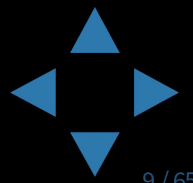
- CSS can be applied to any XML document
  - Not just to HTML / XHTML
- CSS can specify different styles for different media
  - On-screen
  - In print
  - Handheld, projection, etc.
  - … even by voice or Braille-based reader



Telerik Academy

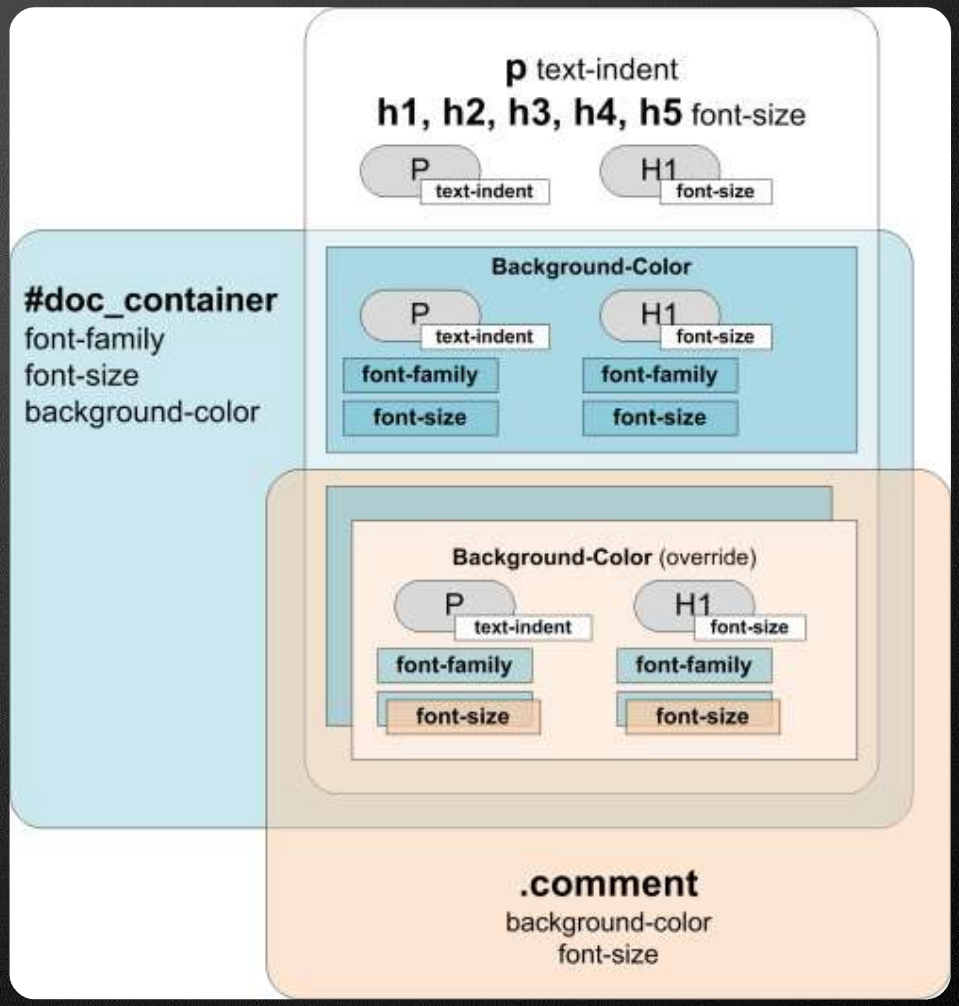
Why "Cascading"?

- Priority scheme determining which style rules apply to element
  - Cascade priorities or specificity (weight) are calculated and assigned to the rules
  - Child elements in the HTML DOM tree inherit styles from their parent
    - Can override them
    - Control via !important rule



Telerik Academy

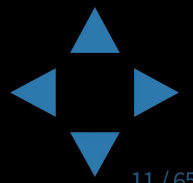
Why "Cascading"?



Telerik Academy

Style Inheritance

- Some CSS styles are inherited and some are not
  - Text-related and list-related properties are inherited: color, font-size, font-family, line-height, text-align, list-style, etc.
  - Box-related and positioning styles are not inherited: width, height, border, margin, padding, position, float, etc
  - <a> elements do not inherit color and text-decoration

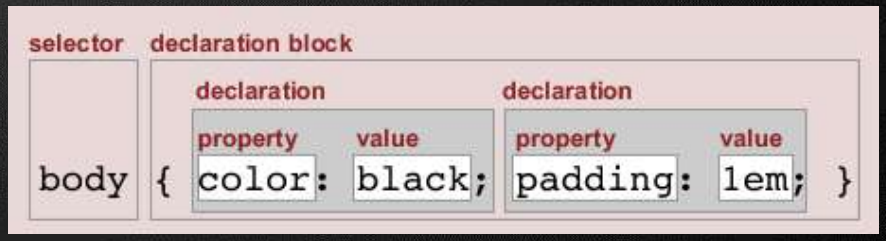


Telerik Academy

Style Sheets Syntax

- Stylesheets consist of rules, selectors, declarations, properties and values
- Selectors are separated by commas
- Declarations are separated by semicolons
- Properties and values are separated by colons

```
h1,h2,h3 { color: green; font-weight: bold; }
```



Telerik Academy

Common Selectors

Select the Elements to Apply a Style

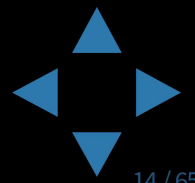
Telerik Academy

Selectors

- Selectors determine which element the rules apply to:
  - All elements of specific type (`tag`)
  - Those that match a specific attribute (`id`, `class`)
  - Elements may be matched depending on how they are nested in the document tree (HTML)
- *Examples*:

```
.header a { color: green }
```

```
#menu>li { padding-top: 8px }
```

Telerik Academy

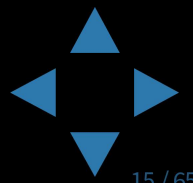
Primary Selectors

- Three primary kinds of selectors:
- By tag (type selector):

```
h1 { font-family: verdana,sans-serif; }
```

- By element id:

```
#element_id { color: #ff0000; }
```



Telerik Academy

Primary Selectors

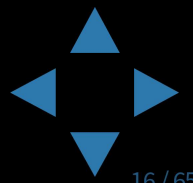
- By element class name (only for HTML):

```
.myClass {border: 1px solid red}
```

- Selectors can be combined with commas:

```
h1, .link, #top-link {font-weight: bold}
```

- This will match <h1> tags, elements with class link, and the element with id top-link



Telerik Academy

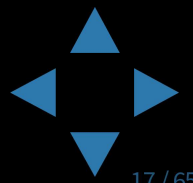
Nested Selectors

- Match relative to element placement:

```
p a {text-decoration: underline}
```

- This will match all <a> tags that are inside of <p>
- * – universal selector (avoid or use with care!):

```
p * {color: black}
```

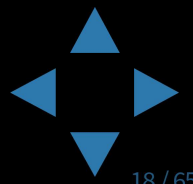
Telerik Academy

Nested Selectors

- This will match all descendants of <p> element
- + selector – used to match “next sibling”:

```
img + .link {float:right}
```

- This will match all siblings with class name link that appear immediately after <img> tag



Telerik Academy

Nested Selectors

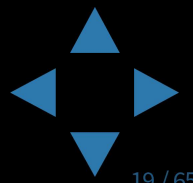
- > selector – matches direct child nodes:

```
p > .error {font-size: 8px}
```

- This will match all elements with class error, direct children of <p> tag
- .class1.class2 (no space!)

```
p.post-text.special {font-weight: bold}
```

- Matches elements with both (all) classes applied at the same time



Telerik Academy

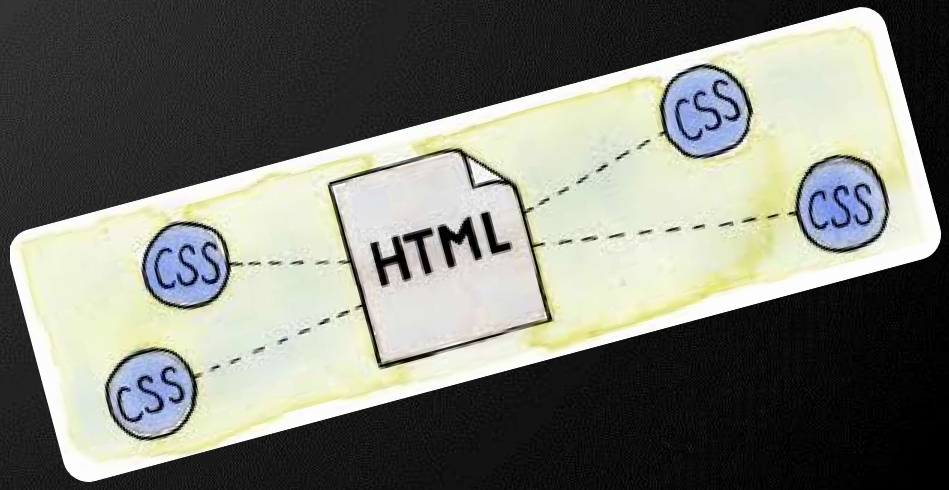
Common Selectors

Demo

Telerik Academy

Importing CSS Into HTML

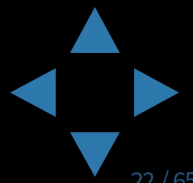
How to Use CSS with HTML?



Telerik Academy

Importing CSS Into HTML

- CSS (presentation) can be imported in HTML (content) in three ways:
  - Inline: the CSS rules in the style attribute
    - No selectors are needed
  - Embedded: in the <head> in a <style> tag
  - External: CSS rules in separate file (best)
    - Usually a file with .css extension
    - Linked via <link rel="stylesheet" href="…"> tag
    - Via @import directive in embedded CSS block



Telerik Academy

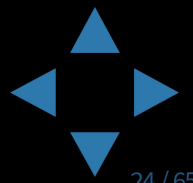
Linking HTML and CSS

- Using external CSS files is highly recommended
  - Simplifies the HTML document
  - Improves page load speed (CSS file is cached)

Telerik Academy

Inline Styles: *Example*

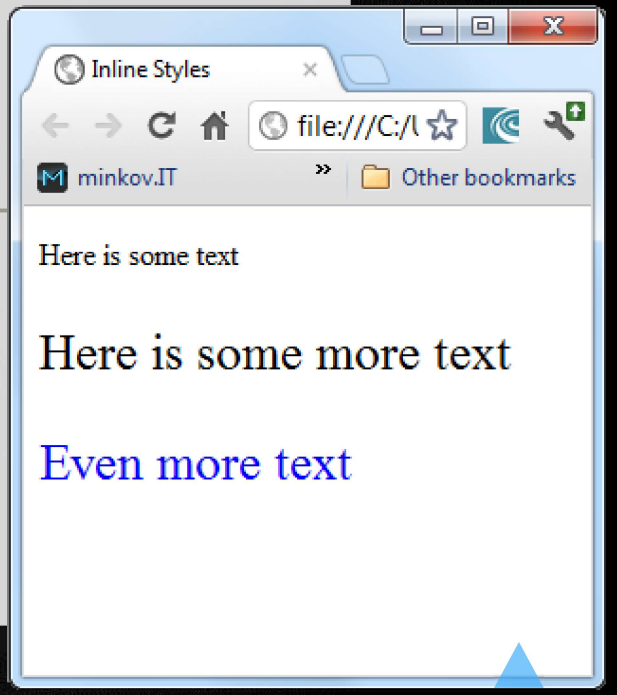
```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Inline Styles</title>
</head>
<body>
  <p>Here is some text</p>

<!-- Separate multiple styles with a semicolon -->

  <p style="font-size: 20pt">Here is some
    more text</p>
  <p style="font-size: 20pt;color:
    #0000FF" >Even more text</p>
</body>
</html>
```



Telerik Academy

Inline Styles: *Example*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Inline Styles</title>
</head>
<body>
  <p>Here is some text</p>

<!-- Separate multiple styles with a semicolon -

  <p style="font-size: 20pt">Here is some
    more text</p>
  <p style="font-size: 20pt;color:
    #0000FF" >Even more text</p>
</body>
</html>
```

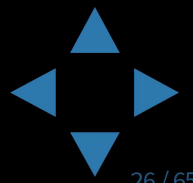


Telerik Academy

Embedded Styles

- Embedded in the HTML in the <style> tag:
  - The <style> tag is placed in the <head> section of the document
  - type attribute specifies the MIME type
    - MIME describes the format of the content
    - Other MIME types include text/html, image/gif, text/javascript …
    - Not required in HTML5
- Used for document-specific styles

```
<style type="text/css">
```



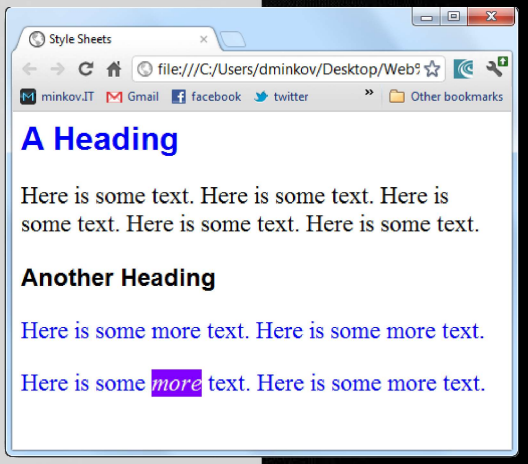
Telerik Academy

Embedded Styles: *Example*

```
<!DOCTYPE html>
<html>
<head>
  <title>Style Sheets</title>
  <style type="text/css">
    em {background-color:#8000FF; color:white}
    h1 {font-family:Arial, sans-serif}
    p  {font-size:18pt}
    .blue {color:blue}
  </style>
<head>
```

Telerik Academy Embedded Styles: *Example*

```
<body>
  <header>
      <h1 class="blue">A Heading</h1>
  </header>
  <article>
      <p>Here is some text. Here is some text.
        Here is some text. Here is some text. Here
        is some text.
      </p>
      <h1>Another Heading</h1>
      <p class="blue">Here is some more text.
         Here is some more text.</p>
      <p class="blue">Here is some <em>more</em>
         text. Here is some more text.
     </p>
  </article>
</body>
</html>
```



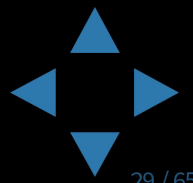
Telerik Academy

External CSS Styles

- External linking
  - Separate pages can all use a shared style sheet
  - Only modify a single file to change the styles across your entire Web site (see http://www.csszengarden.com)
- link tag (with a rel attribute)
  - Specifies a relationship between current document and another document

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

  - link elements should be in the <head>



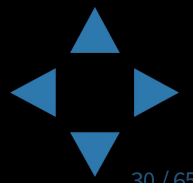
Telerik Academy

External CSS Styles

- @import
  - Another way to link external CSS files
  - *Example*:

```
<style type="text/css">
@import url("styles.css");  /* same as */
@import "styles.css";
</style>
```

  - Ancient browsers do not recognize @import
  - Use @import in an external CSS file to workaround the IE CSS file limit of 31 files



Telerik Academy

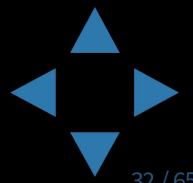
External Styles: *Example*

```
/* CSS Document */
a {
    text-decoration: none
}

a:hover {
    text-decoration: underline;
    color: red;
    background-color: #CCFFCC
}

li em {
    color: red;
    font-weight: bold
}

ul {
    margin-left: 2cm
}

ul ul {
    text-decoration: underline;
    margin-left: .5cm
}
```

Telerik Academy

External Styles: *Example*

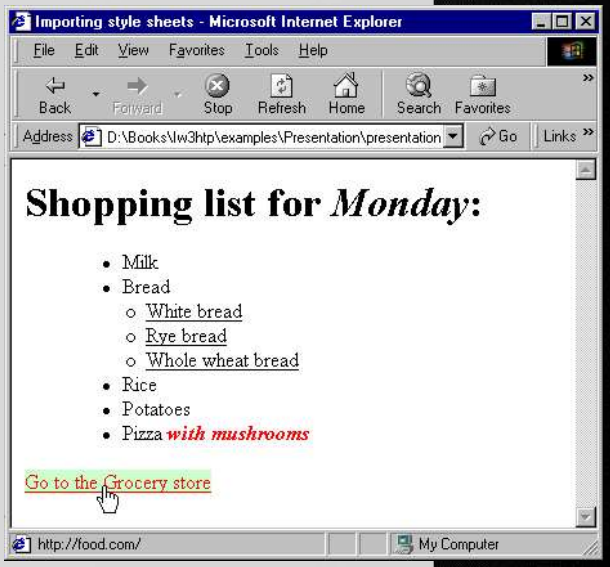
```
  <li>Bread
    <ul>
      <li>White bread</li>
      <li>Rye bread</li>
      <li>Whole wheat bread</li>
    </ul>
  </li>
  <li>Rice</li>
  <li>Potatoes</li>
  <li>Pizza <em>with mushrooms</em></li>
</ul>
<a href="http://food.com" title="grocery
  store">Go to the Grocery store</a>
</body>
</html>
```

Telerik Academy

External Styles: *Example*

```
  <li>Bread
    <ul>
      <li>White bread</li>
      <li>Rye bread</li>
      <li>Whole wheat bread</li>
    </ul>
  </li>
  <li>Rice</li>
  <li>Potatoes</li>
  <li>Pizza <em>with mushrooms</em></li>
</ul>
<a href="http://food.com" title="grocery
  store">Go to the Grocery store</a>
</body>
</html>
```



Telerik Academy

Attribute Selectors

Picking Elements with Certain Attributes







Telerik Academy

Attribute Selectors

- [ ] selects elements based on attributes
  - Element with a given attributeSelects <a> elements with title

```
a[title] {color:black}
```

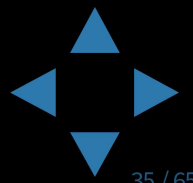
  - Elements with a concrete attribute value
  - Selects <input> elements with type=text

```
input[type=text] { font-family:Consolas}
```

  - Elements whose attribute values contain a word

```
a[title*=logo] {border: none}
```

  - Selects <a> elements whose title attribute value contains logo



Telerik Academy

Attribute Selectors

Demo

Telerik Academy

Pseudo Selectors

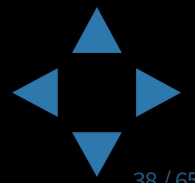
Relative to Element Content or State

Telerik Academy

Common Pseudo Selectors

- Pseudo-classes define state
  - `:hover`, `:visited`, `:active` , `:lang`
- Pseudo-elements define element "parts" or are used to generate content
  - `:first-line` , `:before`, `:after`

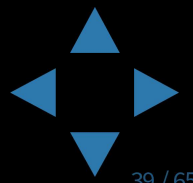
```
a:hover {
    color: red;
}

p:first-line {
    text-transform: uppercase;
}

.title:before {
    content: "»";
}

.title:after {
    content: "«";
}
```

Telerik Academy

Common Pseudo Selectors

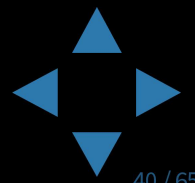
Demo



Telerik Academy

Structural Pseudo-classes

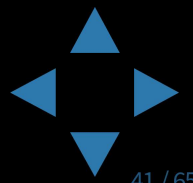
- `:root`
  - The root of the document
- `E:nth-child(n)`
  - An E element, the n-th child of its parent
- `E:nth-last-child(n)`
  - An E element, the n-th child of its parent, counting from the last on
- `E:nth-of-type(n)`
  - An E element, the n-th sibling of its type



Telerik Academy

Structural Pseudo-classes

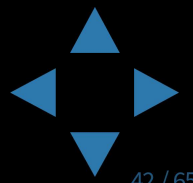
- `E:nth-last-of-type(n)`
  - An E element, the n-th sibling of its type, counting from the last one
- `E:last-child`
  - An E element, last child of its parent
- `E:first-of-type`
  - An E element, first sibling of its type
- `E:last-of-type`
  - An E element, last sibling of its type



Telerik Academy

Structural Pseudo-classes

- `E:only-child`
  - An `E` element, only child of its parent
- `E:only-of-type`
  - An `E` element, only sibling of its type
- `E:empty`
  - An `E` element that has no children (including text nodes)
- More detailed descriptions:
- http://www.w3.org/TR/css3-selectors/#structural-pseudos



Telerik Academy

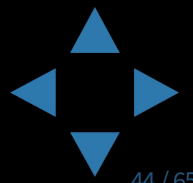
Structural Selectors

Demo

Telerik Academy

The UI Element States Pseudo-Classes

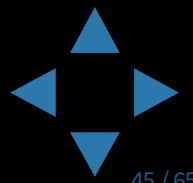
- `E:enabled`
  - A user interface element E which is enabled
- `E:disabled`
  - A user interface element E which is disabled
- `E:checked`
  - A user interface element E which is checked (for instance a radio-button or checkbox)
  - Currently supported only in Opera and IE10 !



Telerik Academy

UI Selectors

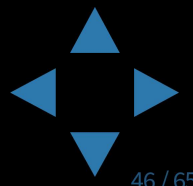
Demo



Telerik Academy

Other CSS 3 Selectors

- `E:target`
  - An E element being the target of the referring URI
- `E:not(s)`
  - An E element that does not match simple selector
- `E~F`
  - An F element preceded by an E element



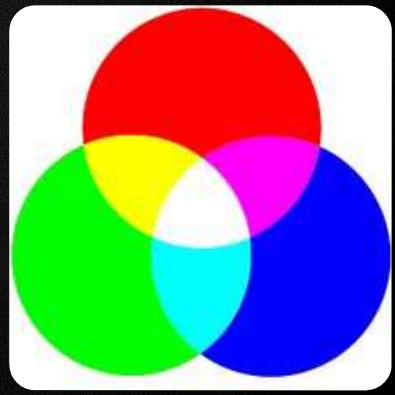
Telerik Academy

Other CSS 3 Selectors

Demo

Telerik Academy

CSS Values



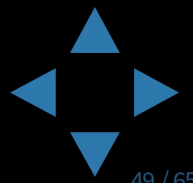
Telerik Academy

CSS Values

- All values in CSS are strings
  - They can represent values that are not strings
  - I.e. 14px means size 14 pixels
- Colors are set in a red-green-blue format (RGB)
  - Both in hex and decimal

```
li.nav-item {
  color: #44f1e1
}
```

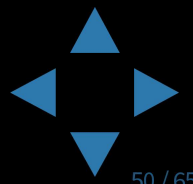
```
li.nav-item {
  color: rgb(68, 241, 255)
}
```



Telerik Academy

Size Values

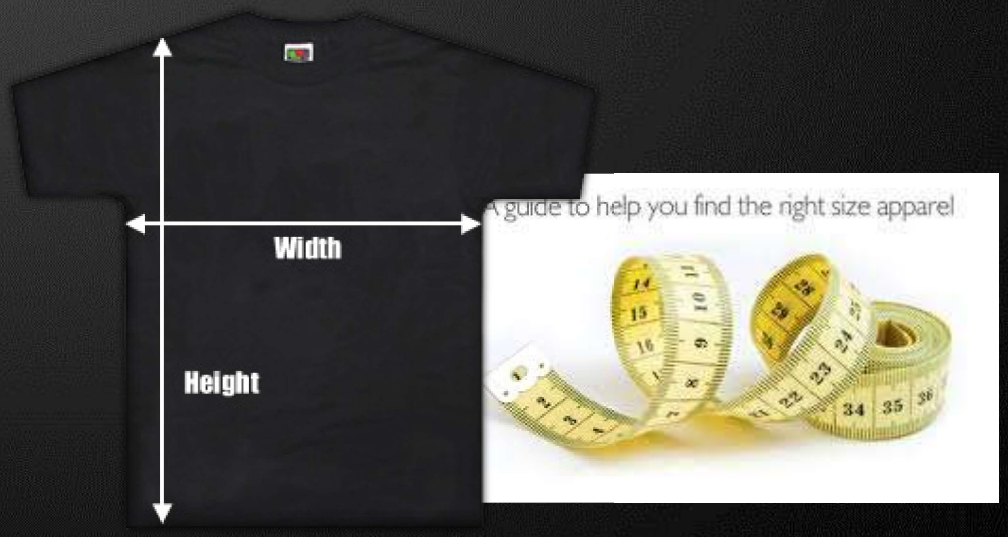
- When setting a size (width, height, font-size…) the values are given as numbers
  - Multiple formats / metrics may be used
  - Pixels, ems, e.g. `12px` , `1.4em`
  - Points, inches, centimeters, millimeters
    - E.g. `10pt` , `1in`, `1cm`, `1mm`
  - Percentages, e.g. `50%`
    - Of the size of the container/font size
  - Zero can be used with no unit: `border: 0;`



Telerik Academy

Size Values

Demo



Telerik Academy

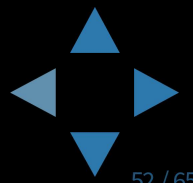
Color Values

- Colors in CSS can be represented in few ways
  - Using red-green-blue
    - Or red-green-blue-alpha

```
color: #f1a2ff
color: rgb(241, 162, 255)
color: rgba(241, 162, 255, 0.1)
```

    - The opacity values are from 0.0 to 1.0
  - Using hue-saturation-light
    - Or hue-saturation-light-alpha

```
color: hsl(291, 85%, 89%);
color: hsl(291, 85%, 89%, 0.1);
```

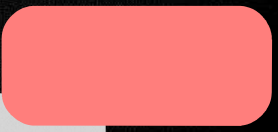



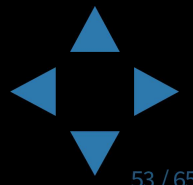
Telerik Academy

RGB Colors

- RGB colors are defined with values for red, green and blue intensity
- Syntax:
  - `#44fa36` – values are in hex
  - `rgb(<red>, <green>, <blue>)` – decimal values
- The range for red, green and blue is between integers `0` and `255`

```
color: #07f2b3
  <!- or -->
color: rgb (7, 242, 179)
```

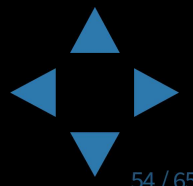




Telerik Academy

RGBA Colors

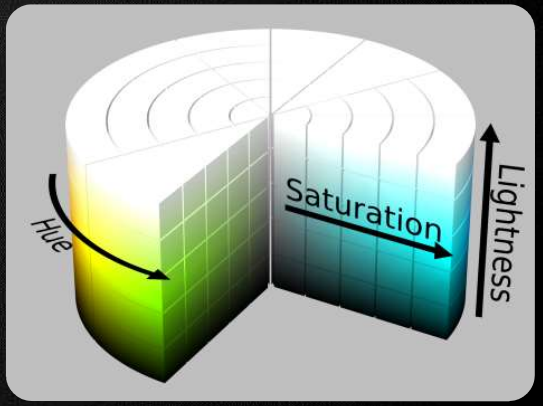
- Standard RGB colors with an opacity value for the color (alpha channel)
- Syntax: rgba(<red>, <green>, <blue>, <alpha>)
- The range for red, green and blue is between integers `0` and 255
- The range for the alpha channel is between `0.0` and `1.0`
- *Example*: rgba(255, 0, 0, 0.5)

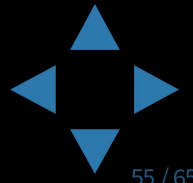


Telerik Academy

HSL Colors

- Hue is a degree on the color wheel
  - `0` (or `360`) is red, `120` is green, `240` is blue
- Saturation is a percentage value
  - `100%` is the full color
- Lightness is also a percentage
  - `0%` is dark (black)
  - `100%` is light (white)
  - `50%` is the average

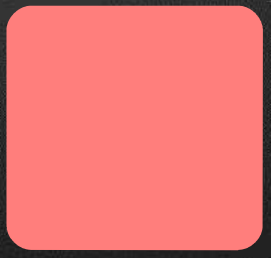


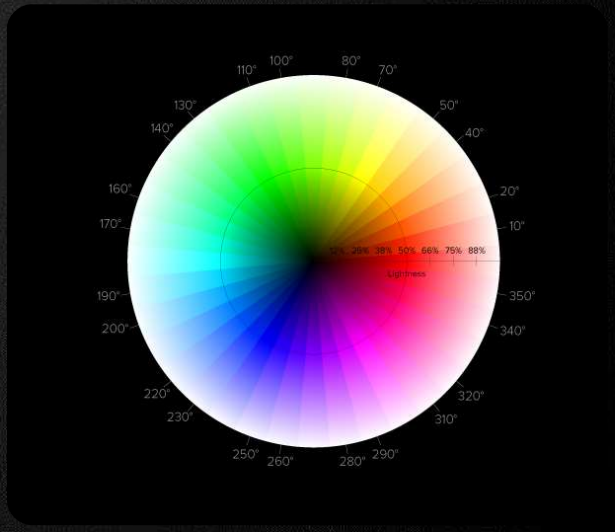


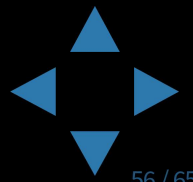
Telerik Academy

HSLA Colors

- HSLA allows a fourth value, which sets the Opacity (via the Alpha channel) of the element
- As RGBA is to RGB, HSLA is to HSL
- Supported in IE9+, Firefox 3+, Chrome, Safari, and in Opera 10+
- *Example*:
  - hsla(0, 100%, 50%, 0.5)
  - Result:







Telerik Academy

Color Values

Demo

Telerik Academy

Default Browser Styles

Why Things Look Different on Different Browsers?

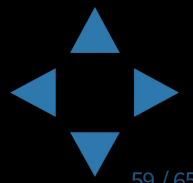
Telerik Academy

Default Browser Styles

- Browsers have predefined CSS styles
  - Used when there is no CSS information or any other style information in the document
- Caution: default styles differ in browsers
  - E.g. margins, paddings and font sizes differ most often
  - Usually developers reset them

```
* {
    margin: 0;
    padding: 0;
}
```

```
body, h1, p, ul, li {
    margin: 0;
    padding: 0;
}
```

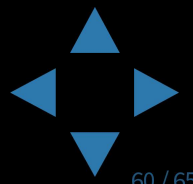



Telerik Academy

CSS Cascade (Precedence)

- There are browser, user and author stylesheets with "normal" and "important" declarations
  - Browser styles (least priority)
  - Normal user styles
  - Normal author styles (external, in head, inline)
  - Important author styles
  - Important user styles (max priority)

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

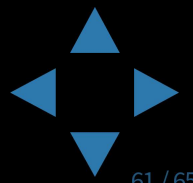
Telerik Academy

CSS Specificity

- CSS specificity is used to determine the precedence (priority) of the CSS style declarations with the same origin
  - Simple calculation:

```
#id = 100, .class = 10, :pseudo = 10, [attr] = 10, tag = 1, * = 0
```

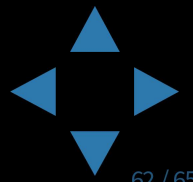
  - Same number of points? Order matters!
  - See also:
    - CSS Specificity: Things You Should Know
    - CSS Selectors



Telerik Academy

CSS Rules Precedence

Demo



Telerik Academy

CSS References

- The CSS documentation at WebPlatform.org:
  - http://docs.webplatform.org/wiki/css
- CSS documentation at MDN
  - https://developer.mozilla.org/en-US/docs/CSS
- CSS3 tutorial
  - http://www.w3schools.com/css3/
- A list of all CSS 2.1 properties is available at http://www.w3.org/TR/CSS2/propidx.html