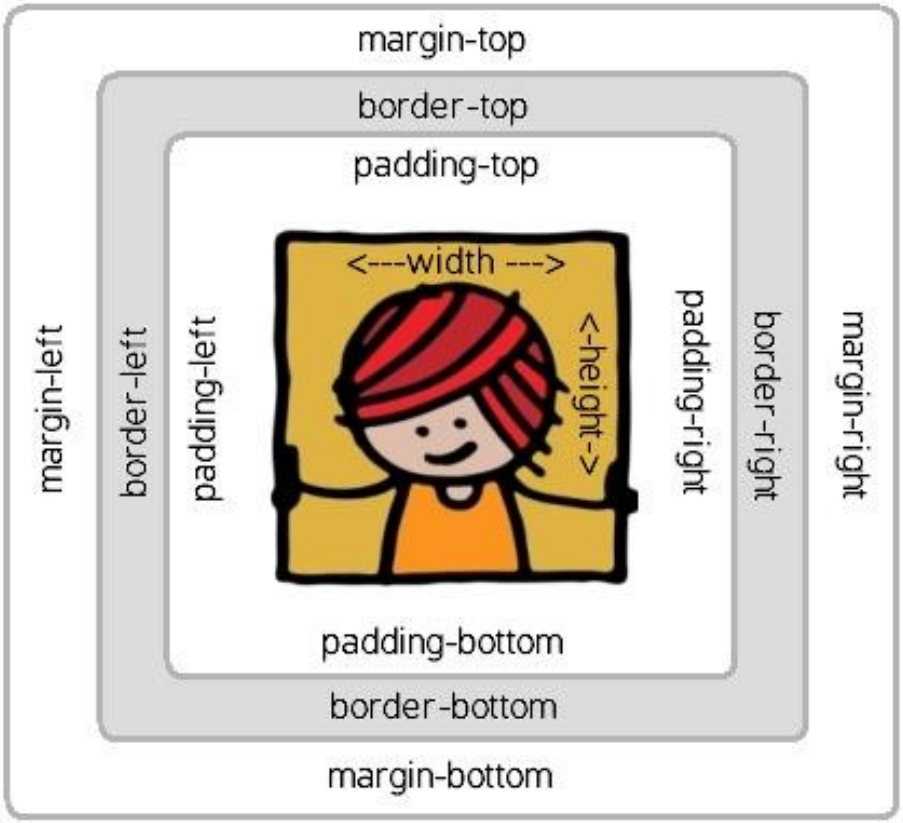
NTNU – Trondheim
Norwegian University of
Science and Technology

Cascading Style Sheets: Selectors and Properties

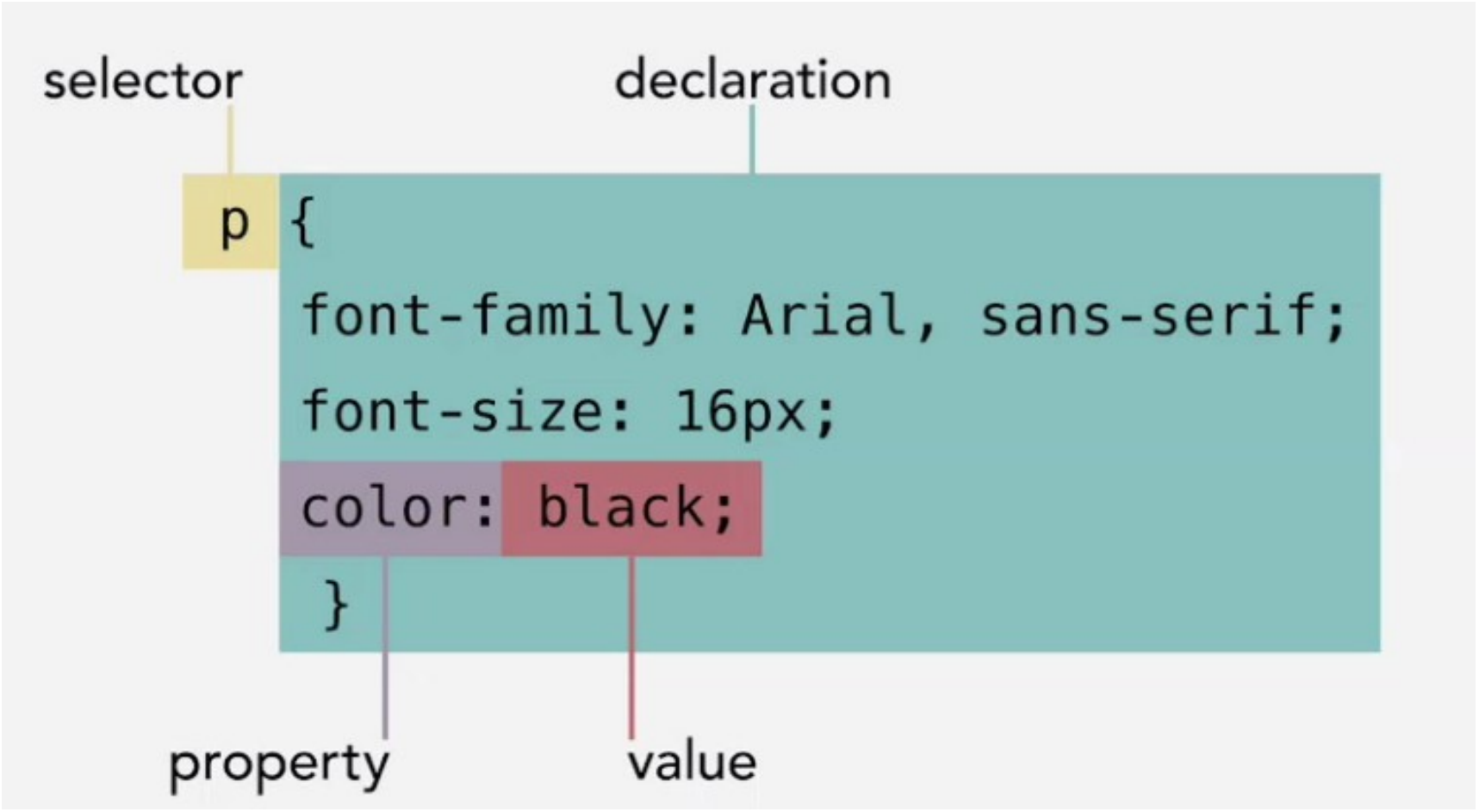
Kshitij Sharma
Norwegian University of Science & Technology (NTNU)
Department of Computer and Information Science

How CSS sees an HTML document?

- CSS treats HTML element as it appears inside its own box and uses rules to indicate how that element should look



CSS Syntax


selector
declaration
p {
font-family: Arial, sans-serif;
font-size: 16px;
color: black;
}
property
value

Using the selectors you can select every element you want!!

```
#titanic{
        float: none;
}
```

```
#tower-of-pisa{
        font-style: italic;
}
```

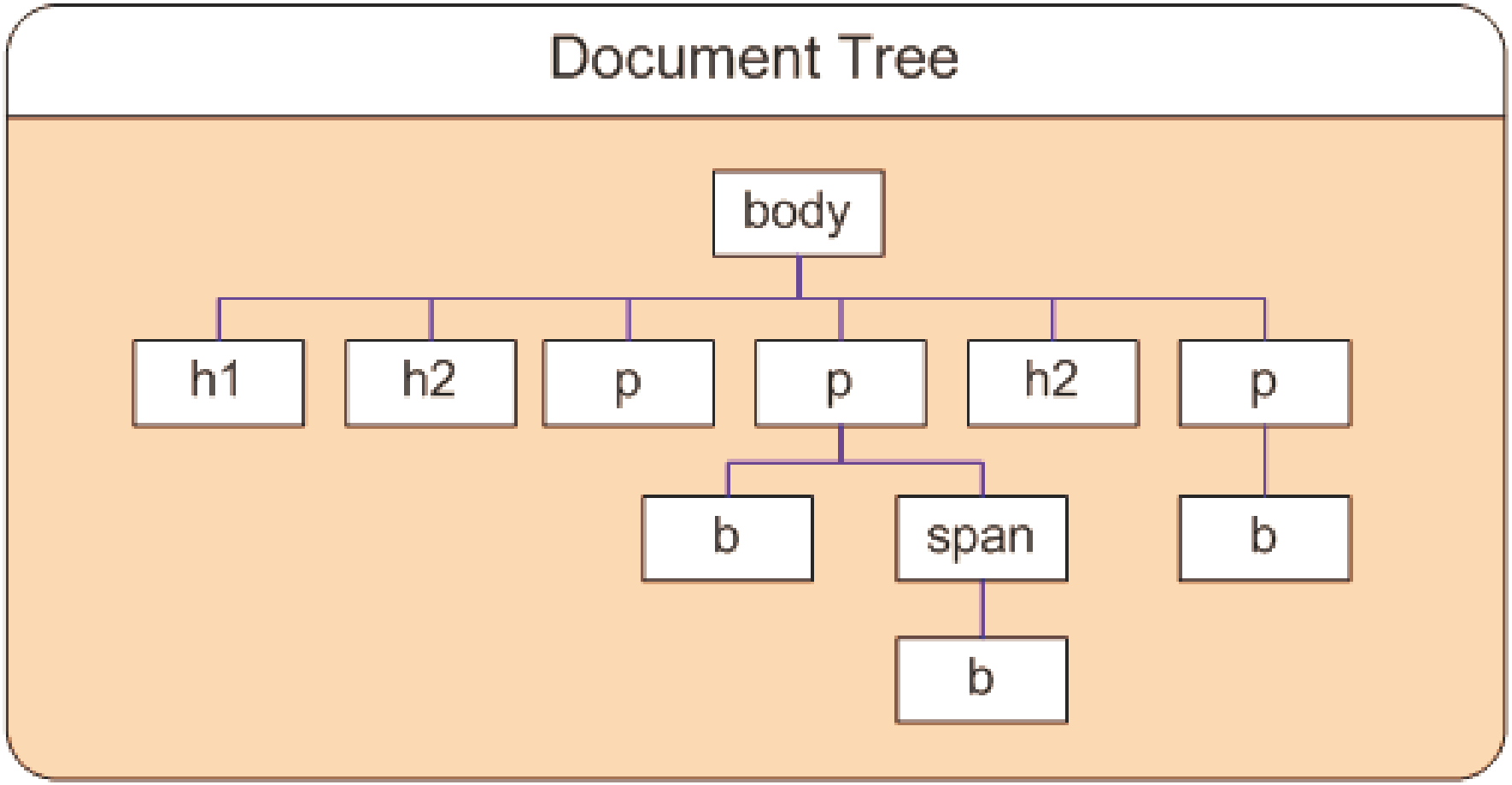
```
#government{
        transition: all  4yr
ease-out;
}
```

```
.illuminati{
        position: absolute;
        visibility: hidden;
}
```

Build the Document/HTML tree

```
<html>
    <head>
        <style type="text/css">  </style>
     </head>
      <body>
          <h1> Heading 1</h1>
          <h2> Heading 2.a</h2>
           <p> First paragraph. </p>
           <p> Second paragraph has a <b>bold</b>
                    and a <span>span with another <b>bold</b></span>. </p>
            <h2> Heading 2.b</h2>
             <p> Third paragraph has a <b>bold</b> also. </p>
     </body
</html>
```

the Document/HTML tree



Multiple declarations

- Multiple declarations for the same selector may be organized into semicolon separated groups. For example,

```
h1 {font-weight: bold}
h1 {font-size: 12pt}
h1 {font-family: Helvetica}
h1 {font-style: normal}
```

is equivalent to:

```
h1 {
    font-weight: bold;
    font-size:12pt;
   font-family: Helvetica;
    font-style: normal
 }
```

Grouping

- When several selectors share the same declarations, they may be grouped as a comma-separated list.

h1 {font-family: sans-serif}

h2 {font-family: sans-serif}

h3 {font-family: sans-serif}

is equivalent to:

h1,h2,h3 {font-family: sans-serif}

Selecting Elements

- There are numerous ways of specifying to which elements style rules apply. Here are examples of some of the more commonly used selectors:

p {color:red}	Every p element
h1,h2,h3 {…}	Group selector
strong em {…}	Contextual selector
div[secret="yes"] {…}	Attribute selector
span.important {…}	Class selector
p#1234 {…}	ID selector

Universal Selector

This selector consists of the asterisk character, like this:

```
* {
   background-color: red;
}
```

When used on its own like above, this selects every element on the page.

Element Type Selector

Also called just the “type selector”, this selector matches HTML elements by tag name. Two examples:

```
h2 {
    background-color: red;
}

div {
    background-color: red;
}
```

Class Selector

Selects an element that matches a class name defined in a class attribute in the HTML.

```
.element {
    background-color: red;
}
```

This is easily my favorite selector, and all good CSS developers should use it abundantly. You can put multiple classes separated by spaces on a single class attribute, which makes this selector quite powerful.

ID Selector

This selects an element that matches an id defined in an id attribute in the HTML.

```
#elementID{
    background-color: red;
}
```

```
#ninja{
        color: black;
        visibility: hidden;
        animation-duration: 0.00001s
}
```

Descendant Selector

This selector is defined with a space character separating two selectors, and represents a child element, but not just immediate children, further nested ones as well.

```
p a {
    background-color: red;
}
```

This example targets **any <a> elements that sit inside a <p>** element, even if there are no elements nested between them

Attribute Selector

This selector **targets an element based on an HTML attribute and/or attribute value**. Both examples below are valid attribute selectors:

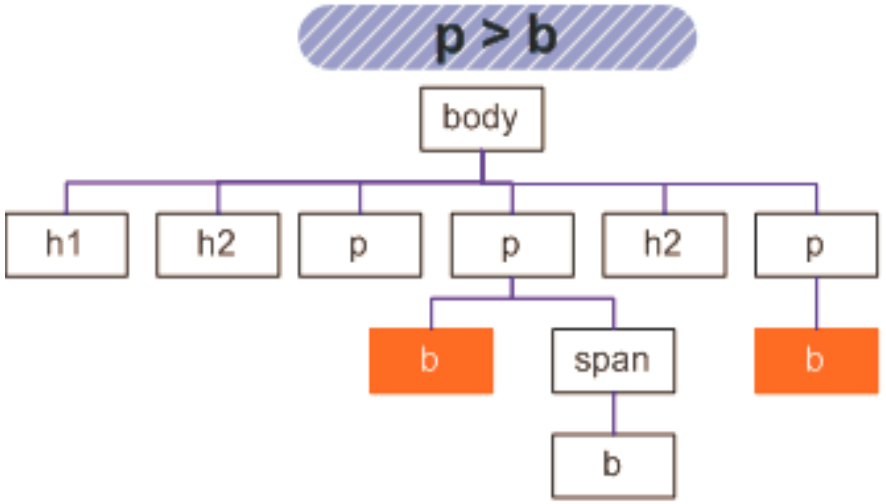
```
div[style] {
    background-color: red;
}

input[type="text"] {
    background-color: red;
}
```

The first example **targets any element that has a “style” attribute**. The second example **targets any <input> element that has a “type” attribute with a value of “text”**.

Child Selector

This selects **an element based on it being an immediate child of another element**:

```
one > two {
    background-color: red;
}
```

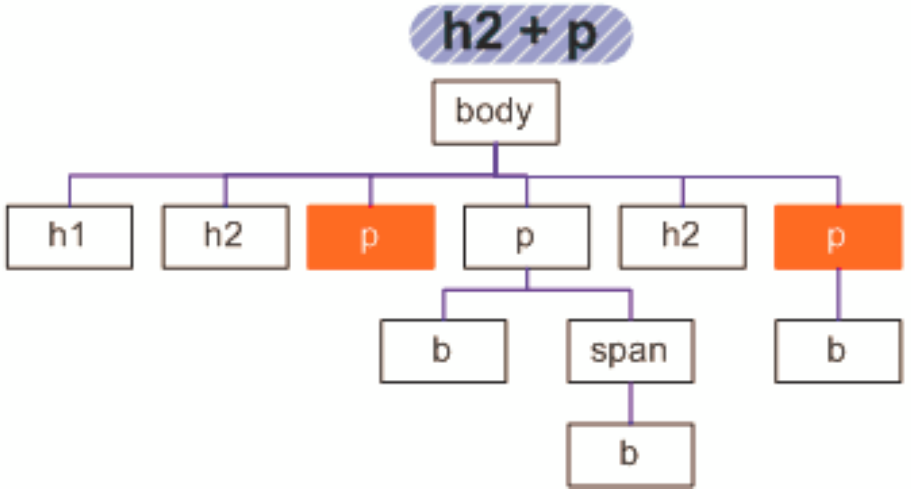


So this **will not style a "two" element unless it is an immediate child of a "one" element**. It can't be nested inside another element, it has to be an immediate child element. And only the child is styled, not the parent.

Adjacent Sibling Selector

This selector, which uses the plus sign, **targets elements that are "adjacent" to each other**, or immediate siblings, and they must have the same parent element.

```
h2+p {
    background-color: red;
}
```

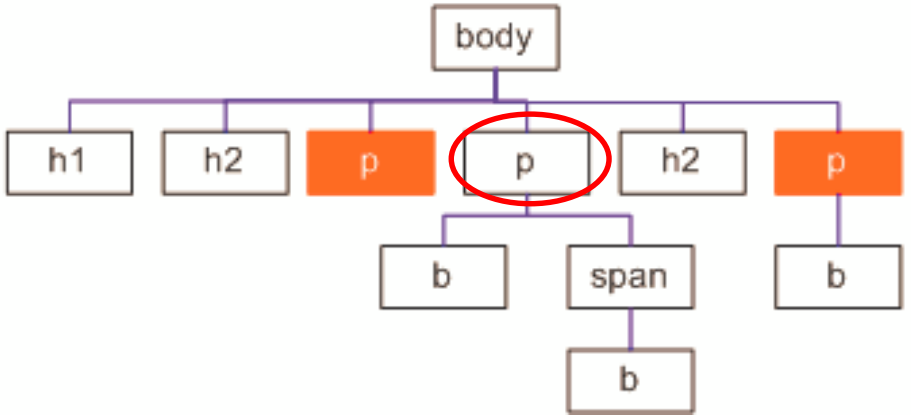


Targets **the first <p> element after any <h2> element**, but not other <p> elements

General Sibling Selector

This uses the **tilde character and is exactly the same as the adjacent sibling selector except the elements don't have to be immediate siblings**.

```
h2~p {
    background-color: red;
}
```



If you had two <p> elements that are sibling of an <h2> element, this rule would apply to both.

Pseudo-class

While technically falling under the category of “selectors”, these are not normally referred to as “selectors”, but just pseudo-classes. Pseudo-class **selects an element based on a state the element is in**. Here are a few examples:

```
a:visited {
    background-color: red;
}

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

Pseudo-element

CSS pseudo-elements are used to add special effects to some selectors.

Again, not normally referred to as a selector, these actually represent elements in the HTML page that are not really part of the rendered HTML:

```
p::first-letter {
    background-color: red;
}

p::before {
    content: "Read this -";
    background-color: yellow;
}
```

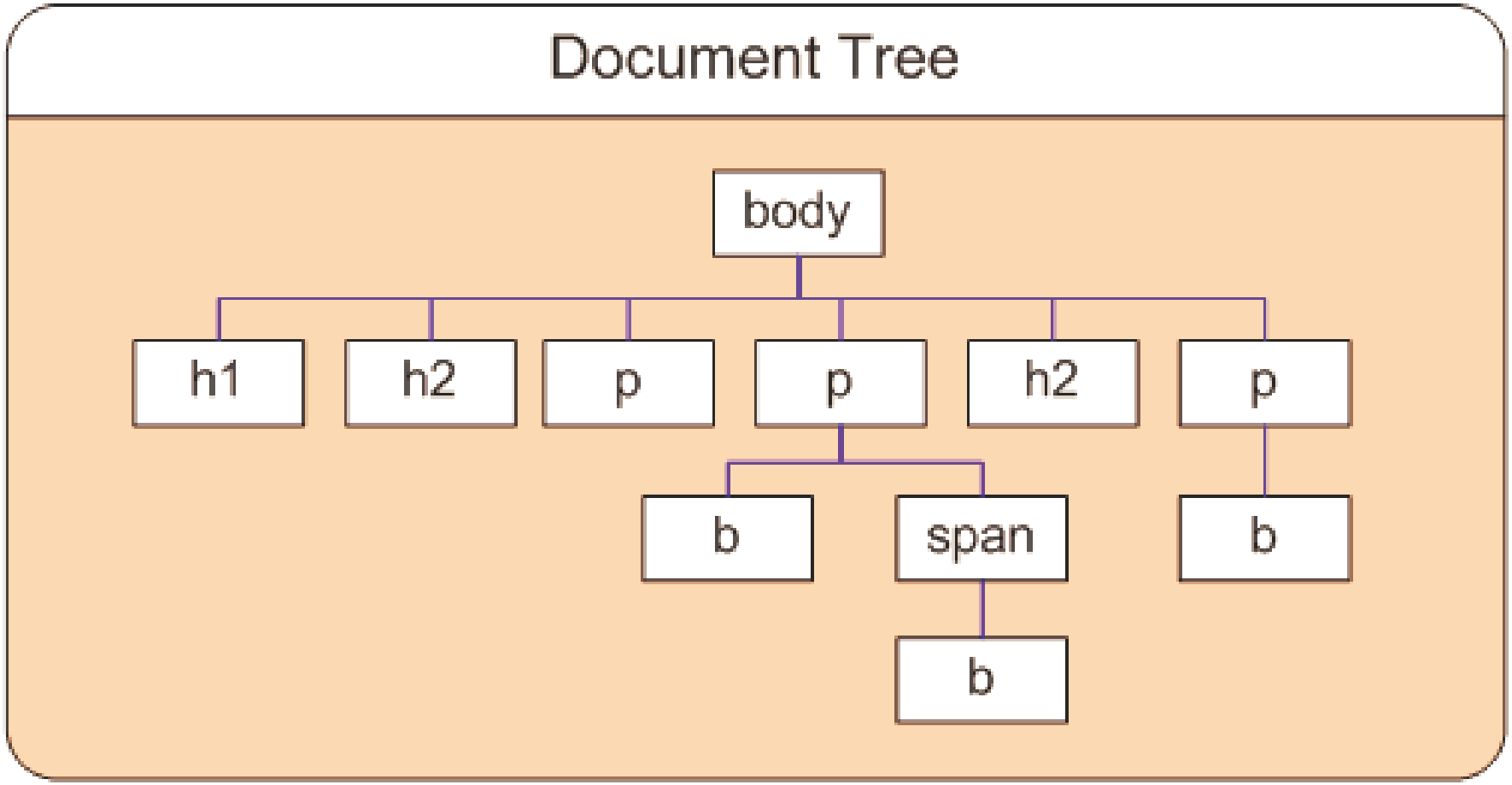
Why sometimes double colons (::)?

- Sometimes you will see double colons (::) instead of just one (:).
- This is part of CSS3 and an attempt to distinguish between pseudo-classes and pseudo-elements. Most browsers support both values.
- Pseudo-classes act as simple selectors in a sequence of selectors and thereby classify elements on non-presentational characteristics, **pseudo-elements create new virtual elements**.

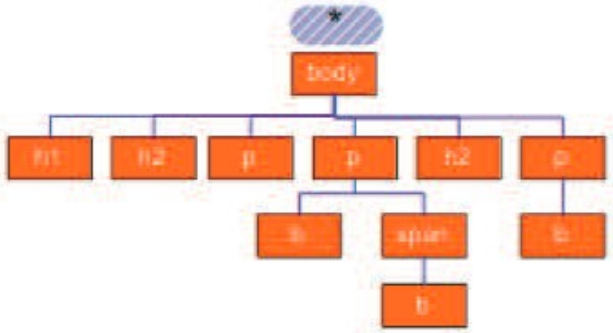
All CSS Pseudo Classes/Elements

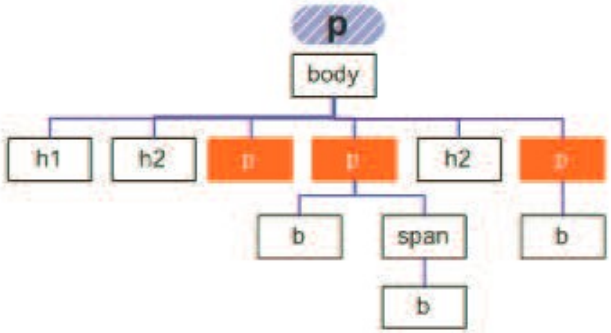
Selector	Example	Example description
:link	a:link	Selects all unvisited links
:visited	a:visited	Selects all visited links
:active	a:active	Selects the active link
:hover	a:hover	Selects links on mouse over
:focus	input:focus	Selects the input element which has focus
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
::before	p::before	Insert content before every <p> element
::after	p::after	Insert content after every <p> element
:lang(*language*)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"

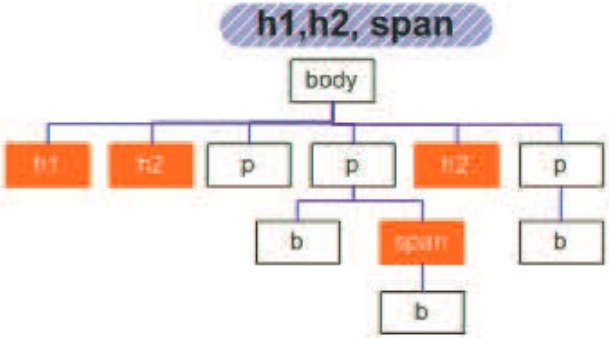
the Document/HTML tree

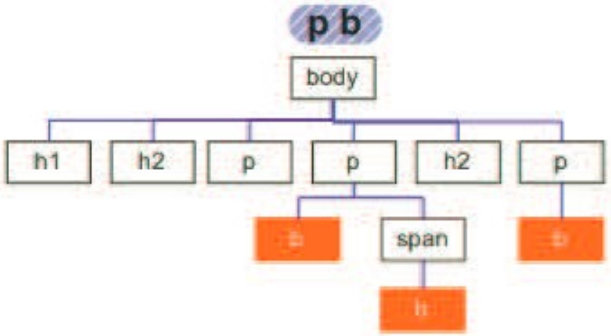


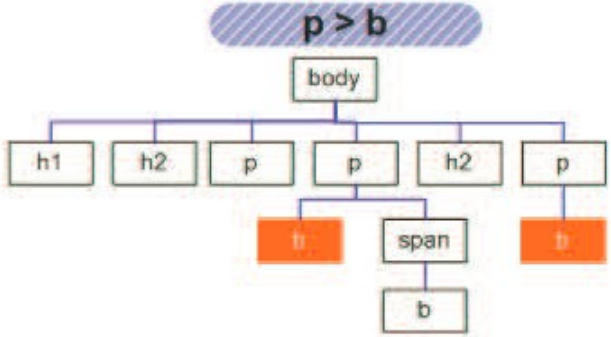
Examples Summary

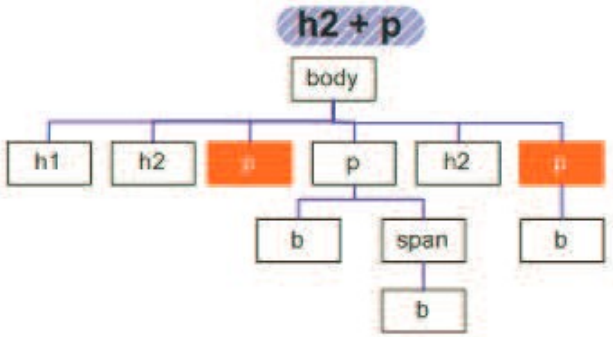












Code Summary

- Any selector example * {color:red;}
- Type Selector p example p {color:red;}
- Element Selector h1,h2,span ex. h1,h2,span {color:red;}
- Descendent Selector p b example p b {color:red;}
- Child Selector p>b example p>b {color:red;}
- Adjacent Sibling Selector h2+p example h2+p {color:red;}
- General Sibling Selector h2~p example h2~p {color:red;}

Element CLASS selector

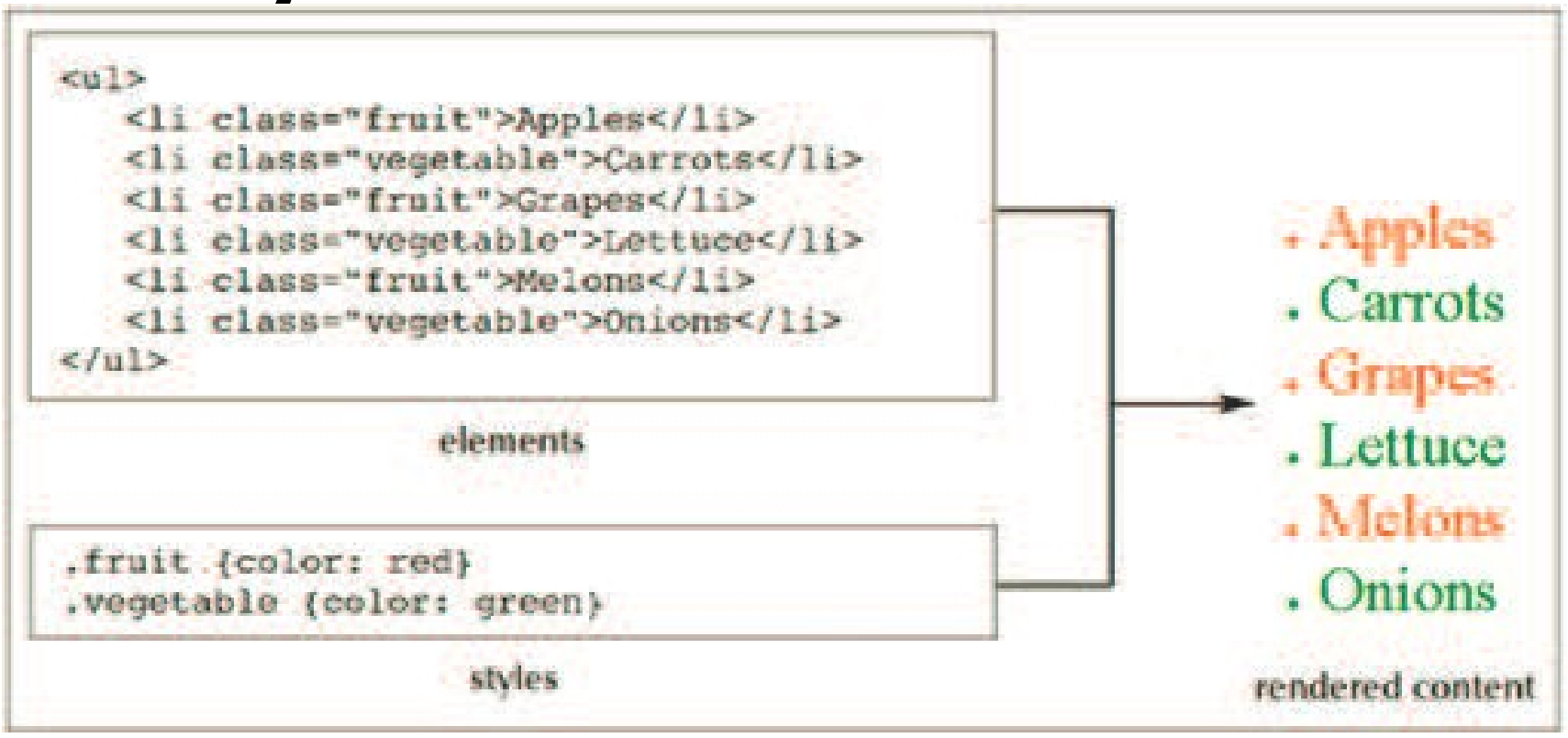
- Elements can be selected on the basis of their **class**:

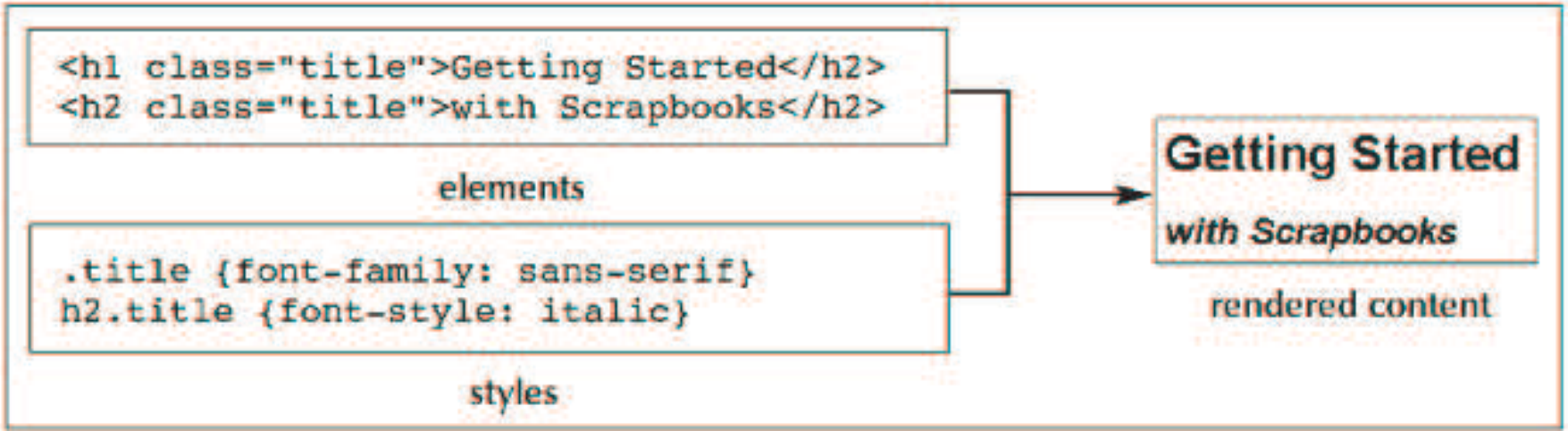
```
<html><head>
<title>CLASS selector example</title>
<style type="text/css">
        .important {font-size:larger}
        .trivial {font-size:smaller}
</style></head>
<body><h2>Warning</h2>
        <p class="important">Important text</p>
        <p class="trivial">Less important text</p>
</body>
</html>
```

What this CSS will do?

```
.bruce-banner{
        color: pink;}
.hulk{
        color: green;}
```

Style for a class of elements





Element ID selector

- Styles can be applied to elements with a specific id

```
<html><head>
<title>ID selector example</title>
<style type="text/css">

        p {font-family: "Times New Roman", serif}
        #special {font-family: Courier, sans-serif}

</style></head>
<body>
        <p>Any element may have a ID attribute.</p>
        <p id="special">So long as it is unique.</p>
</body>
</html>
```

What this CSS will do?

```
.hobbit{
        height: 50%;}
.hobbit #foot{
        width: 200%;}
```

Pseudo-element selectors

- CSS has two pseudo-elements: first-letter and first-line.

```
<html>
    <head><title>The style of a news letter</title>
        <style type="text/css">
            p.special:first-line {font-variant:small-caps}
            p.special:first-letter {font-size: 300%; float:left}
        </style>
    </head>
    <body>
        <h1>Student Club News</h1>
        <p class="special">The first line of this news letter is in
            small capital letters and the first letter is 300% larger (
            a drop cap).</p> …
```

What this CSS will do?

The resulting page …

Student Club news

THE FIRST LINE OF THIS NEWS LETTER IS IN SMALL CAPITAL LETTERS AND THE First letter is 300% larger (a drop cap).

Types of selectors (extra)

- Contextual selectors
  - **h1 em {color: blue}** /*the **** within the **<h1>***/
  - **body > p {line-height: 1.3}** /***<p>** directly within the **<body>***/
  - **h1+h2 {margin-top: -5mm}** /* **<h2>** directly after an **<h1>** */
  - **div > p:first-child {text-indent: 0}**

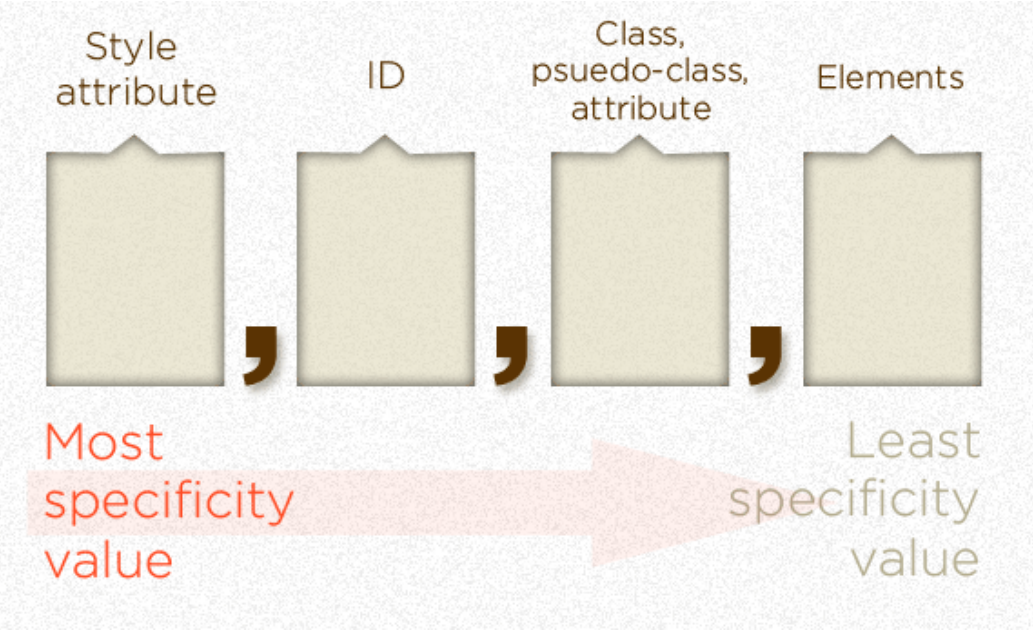
    /***<p>** that is the first child of a ** ***/
- Selection on attribute
  - **a[href] {border: solid}** /* selection by existence */
  - **span[class="example"] {color:blue}** /* selection by value */
  - **div[status~="important"] {z-order:2}**

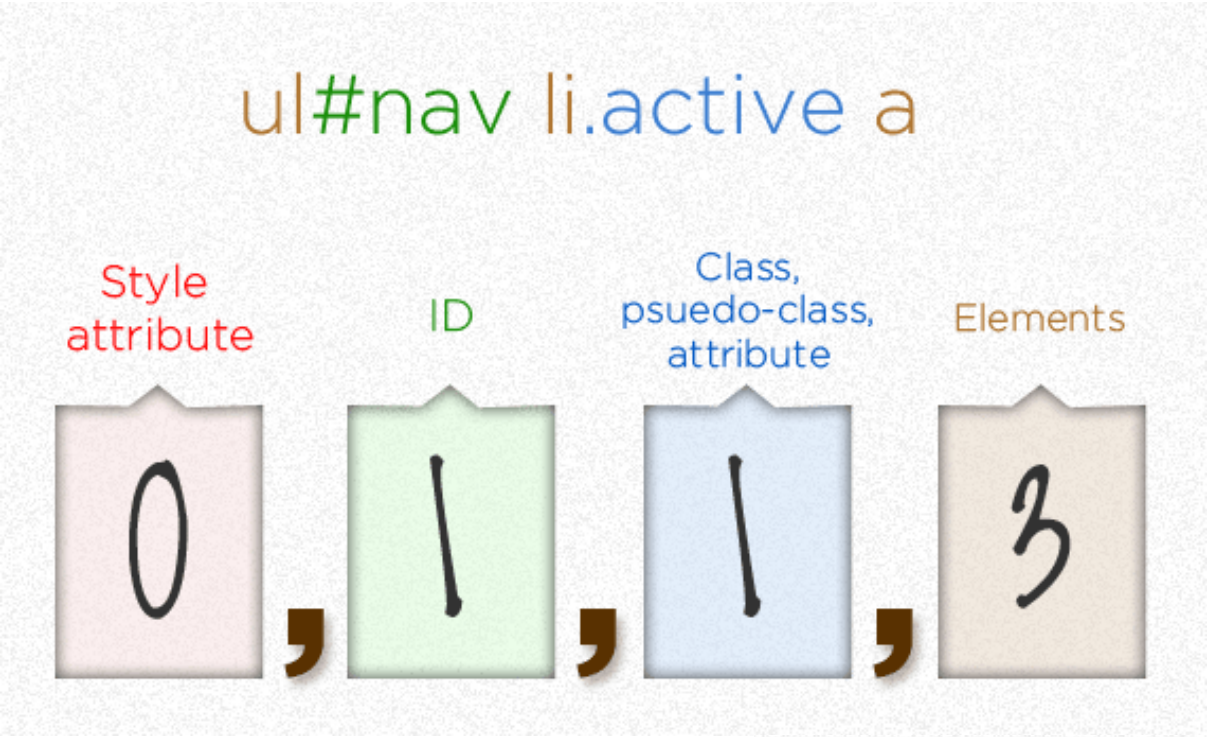
    /* selection by value: from comma-separated list */

Calculating CSS Specificity Value

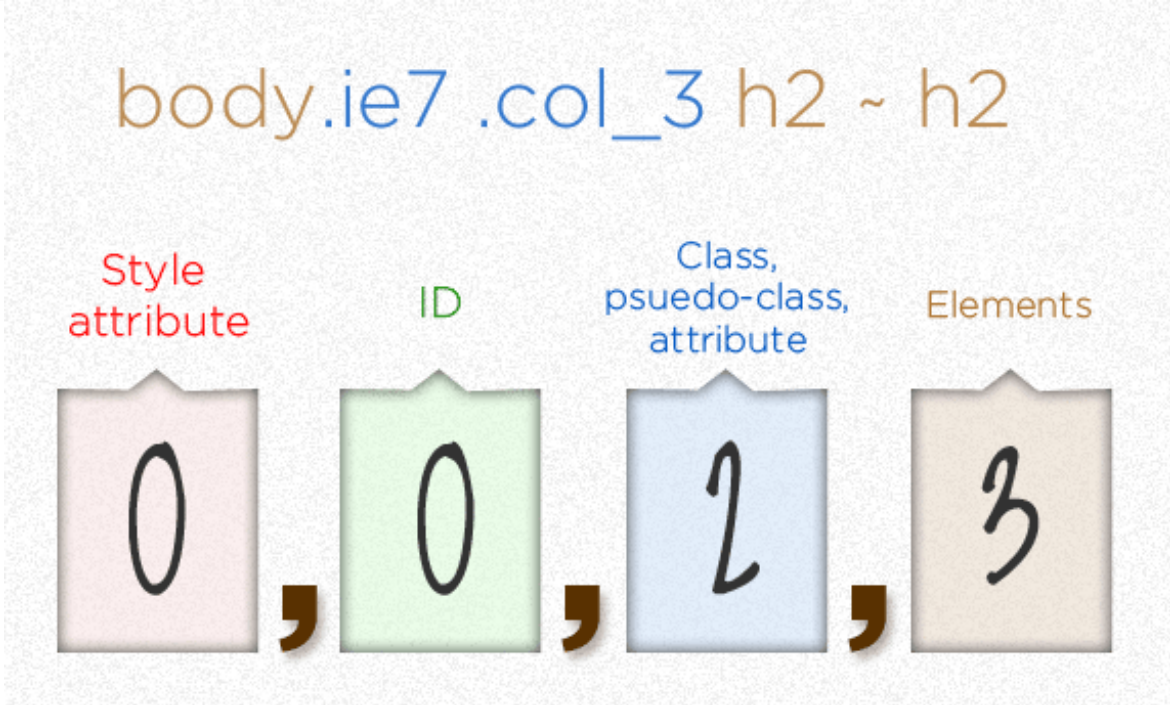
In other words:

- If the element has inline styling, that automatically wins (1,0,0,0 points)
- For each ID value, apply 0,1,0,0 points
- For each class value (or pseudo-class or attribute selector), apply 0,0,1,0 points
- For each element reference, apply 0,0,0,1 point





Sample Calculations



https://flukeout.github.io/

http://davidshariff.com/quiz/

P0 and P1

P0 Getting Started (Sept 15) and

P1 Project Requirements (Sept 20)

Next class

Sept 16

Multimedia in HTML

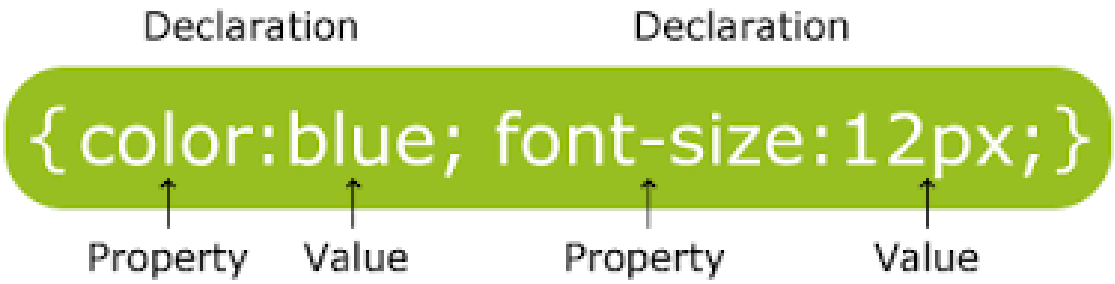
One hour only: 12:15 – 13:00

Questions?

NTNU
Norwegian University of
Science and Technology

Department of
Computer and Information Science

Lets go to Properties now ☺



Properties: Colors

- CSS properties allow authors to specify the foreground color and background of an element. Backgrounds may be colors or images.
- Background properties allow authors to position a background image, repeat it, and declare whether it should be fixed or scroll along with the document.

– color **(a colour)**

em {color:red}

– background-color **(a colour, or “transparent”)**

h1 {background-color:white}

– background-image **(a URI)**

body {background-image:URL(“stripe.gif”)}

CSS2 Properties: Color Values

CSS colors can either be a named color or follow a numerical RGB specification:

– HTML 4.0 Color names are used in CSS2 as well.

- **Aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, sliver, teal, white and yellow**.

– Colors in numerical RGB specification

em {color: rgb(255,0,0)} /* 0-255 (red) */

em {color: rgb(100%, 0%, 0%) /* 0.0% - 100.0% (red) */

– Colors in hexadecimal RGB specification

em {color: #ff0000} /* #rrggbb (red) */

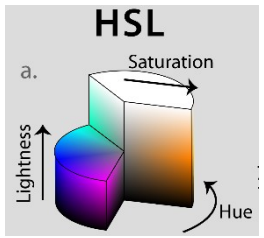
CSS3 Properties: Color Values

- CSS3 has introduced an extra value for RGB colors to indicate opacity. It is known as RGBA

```
background-color: rgba(0, 255, 255, 0.5);
```

- CSS3 also allows you to specify colors as HSL values, with an optional opacity value. It is known as HSLA

```
background-color: hsla(0, 100%, 100%, 0.5);
```



CSS Properties: Fonts

- font family (can specify order of preference)

  body {font-family: “Book Antiqua”, “Times New Roman”, serif}
- font style **(normal, italic or oblique)**

  h1,h2,h3 {font-style:italic}
- font variant **(normal, small-caps)**

  h3 {font-variant: small-caps}
- font weight **(normal, bold, bolder, lighter)**

  strong {font-weight: normal}

CSS Properties: Font Size

- Absolute font sizes used to fix sizes to specific values.
- Five standard units (mm, cm, in, pt, pc):

```
p {font-size: 0.5in}
p {font-size: 1cm}
p {font-size: 5mm}
p {font-size: 12pt}
p {font-size: 3pc}
```

- 1 inch (in) = 72 points (pt) = 6 picas (pc)

CSS Properties: Font Size

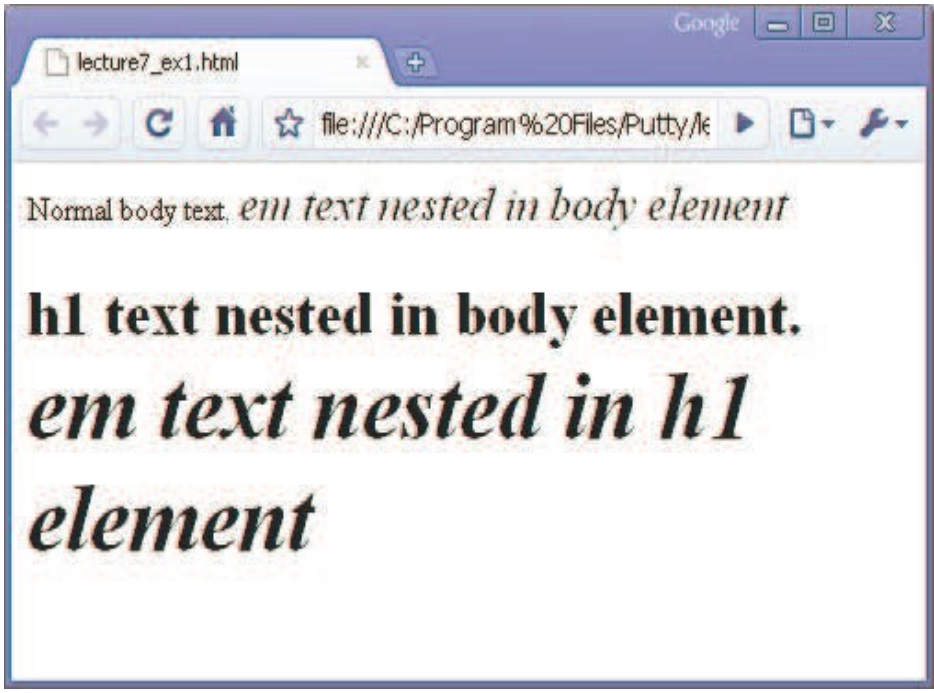
- Relative font size can make web page ***scalable-*** adapts automatically to font that reader uses.
- Examples of relative units: ***percentage*** and ***em unit***

  p {font-size: 150%}

  em {font-size: 1.5em}
- Don’t confuse **em selector** from **em unit**.
- 100% or 1em is equal to font size of the parent element.

CSS Properties: Font Size

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

                   h1 {font-size: 2em}
                   em {font-size: 1.5em}

                </style>
        </head>
        <body>
                Normal body text.
                <em> em text nested in body element</em>
                <h1> h1 text nested in body element.
                     <em> em text nested in h1 element</em>
                </h1>
        </body>
</html>
```



Normal body text = 100%
First em text = 150%
h1 text = 200%
Second em text = 300%

CSS Properties: Font Size in Pixels

Using Pixels:

- 1 pixel = 1 dot on output device
- Different devices have different resolutions
- 600 dpi printer has more pixels per inch than PC monitor

```
h1 { font-size:20px }
```

Using keywords:

- xx-small, x-small, small, medium, large, x-large, xx-large
- smaller, larger (relative: 1 size smaller or larger)

```
h1 { font-size:xx-small }
h1 { font-size:larger }
```

CSS Properties: Text

The presentation of text can be adjusted by:

- text-indent **(the amount of indentation using absolute length or percentage)**

  p {text-indent:3em}

- text-align **(left, center, right, justify)**

  div.center {text-align:center}

- text-decoration **(none, underline, overline, line-through)**

  a[href] {text-decoration: underline}

- Other properties are also available.

Properties: Boxes

- CSS treats each HTML element as if it lives in its own box.
- We can set several properties that affect appearance of these boxes
  - Control the dimensions of the boxes
  - Create borders around boxes
  - Set margins and padding for boxes
  - Show and hide boxes

Working with the Box Model

- The CSS Box Model is essentially a box that wraps around HTML elements. The **box model** is an element composed of four parts:
  - Margin
  - Border
  - Padding
  - Content

Explanation of the different parts:

- **Margin** - Clears an area around the border. The margin does not have a background color, it is completely transparent
- **Border** - A border that goes around the padding and content. The border is inherited from the color property of the box
- **Padding** - Clears an area around the content. The padding is affected by the background color of the box
- **Content** - The content of the box, where text and images appear

Properties: Boxes

- Every displayable element considered to fall inside a rectangular box.
- Each box has an external margin, a border, internal padding and content (e.g. text or images, etc.)
  - margin **(a length or a percentage)**

    body {margin:2em}
  - padding **(a length or a percentage)**

    td {padding: 5pt}
  - border-width **(thin, medium, thick or a length)**

    colgroup {border-width: 5pt}
  - border-color **(a colour)**

    p {border-color:red}
  - border-style **(solid, double, dashed, …)**

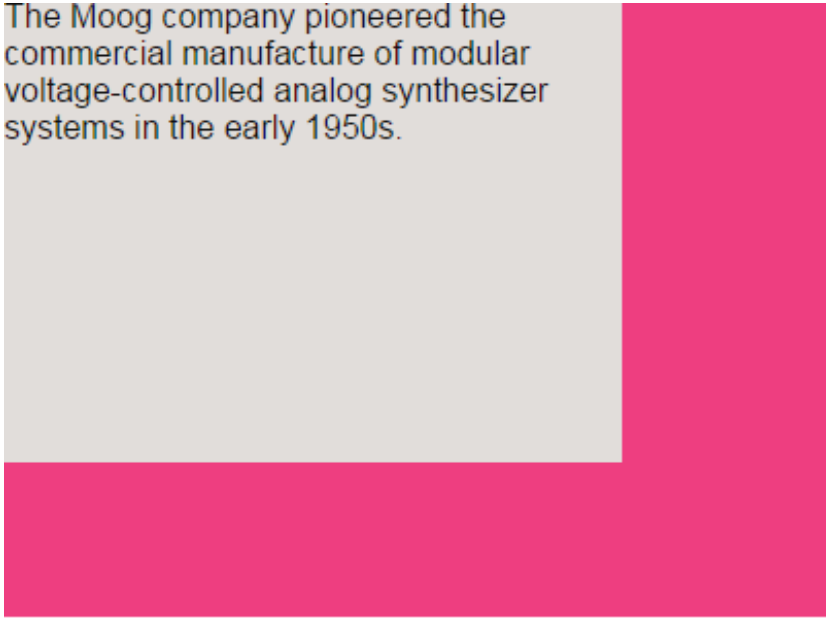
    h2 {border-style:double}
  - Individual box sides can also be targeted:

    div {border-style-left:dashed}

Box Dimensions

 The Moog company pioneered the commercial manufacture of modular voltage-controlled analog synthesizer systems in the early 1950s.

```
div {    width: 400px;
         height: 300px;
         background-color: #ee3e80;}

p {      height: 75%;
         width: 75%;
         background-color: #e1ddda;}
```



min/max-width property

<td></td> <td class="description">The Rhodes piano is an electro-mechanical piano, invented by Harold Rhodes during the fifties and later manufactured in a number of models, first in collaboration with Fender and after 1965 by CBS. It employs a piano-like keyboard with hammers that hit small metal tines, amplified by electromagnetic pickups.</td><td>$1400</td></tr>

```
td.description {  min-width: 450px;
                  max-width: 650px;
                  text-align: left;
                  padding: 5px;
                  margin: 0px;}
```

Photo	Description	Price
	The Rhodes piano is an electro-mechanical piano, invented by Harold Rhodes during the fifties and later manufactured in a number of models, first in collaboration with Fender and after 1965 by CBS. It employs a piano-like keyboard with hammers that hit small metal tines, amplified by electromagnetic pickups.	$1400
	The Wurlitzer electric piano is an electro-mechanical piano, created by the Rudolph Wurlitzer Company of Mississippi. The Wurlitzer company itself never called the instrument an "electric piano", instead inventing the phrase "Electronic Piano" and using this as a trademark throughout the production of the instrument. It employs a piano-like keyboard with hammers that hit small metal tines, amplified by electromagnetic pickups.	$1600
	A Clavinet is an electronically amplified clavichord manufactured by the Hohner company. Each key uses a rubber tip to perform a hammer on a string. Its distinctive bright staccato sound is often compared to that of an electric guitar. Various models were produced over the years, including the models I, II, L, C, D6, and E7.	$1200

Overflowing Content

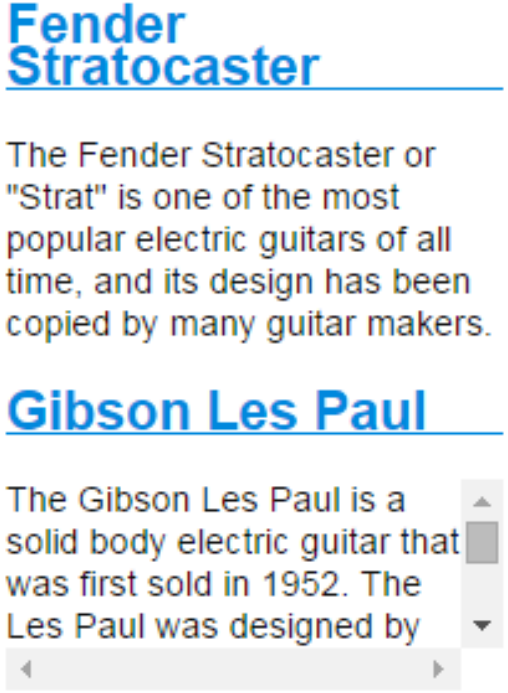
- The overflow property tells the browser what to do if the content is larger than the box itself

```
p.one {overflow: hidden;}
p.two {overflow: scroll;}
```

hidden simply hides any extra text

scroll, adds a scrollbar to the box

```
<h2>Fender Stratocaster</h2>
<p class="one">The Fender Stratocaster or "Strat" is one of the most popular electric guitars of …..</p>
<h2>Gibson Les Paul</h2>
<p class="two">The Gibson Les Paul is a solid body electric guitar that was first sold in 1952….</p>
```

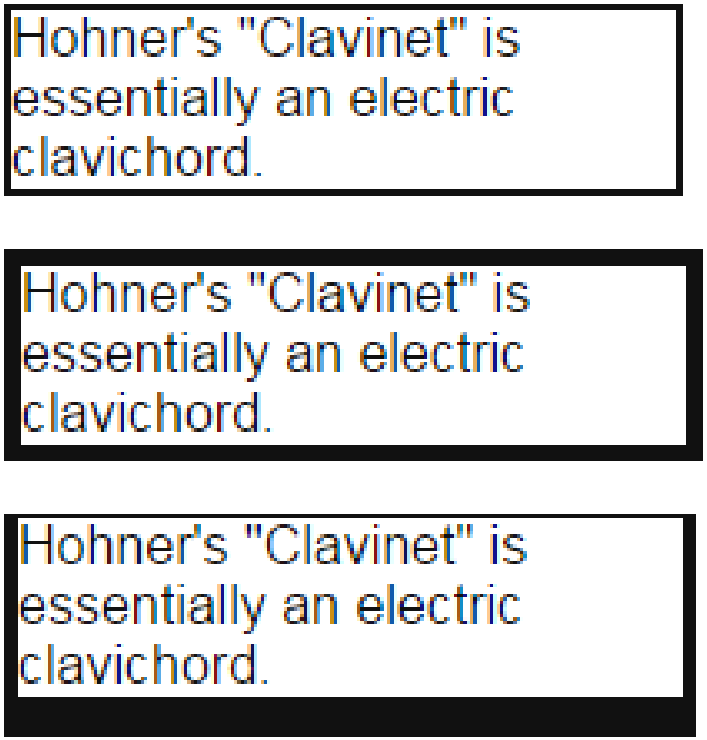


Border Width border-width

The border-width property is used to control the width of borders in pixels on using the thin, medium or thick value.

```
<p class="one">Hohner's "Clavinet" is essentially an electric clavichord.</p>
<p class="two">Hohner's "Clavinet" is essentially an electric clavichord.
</p> <p class="three">Hohner's "Clavinet" is essentially an electric clavichord.</p>

p.one { border-width: 2px;}
p.two {border-width: thick;}
p.three {border-width: 1px 4px 12px 4px;}
```

clockwise starting from the top



Border Style border-style

```
<p class="one">Wurlitzer Electric Piano</p>
<p class="two">Wurlitzer Electric Piano</p>
<p class="three">Wurlitzer Electric Piano</p>
<p class="four">Wurlitzer Electric Piano</p>
<p class="five">Wurlitzer Electric Piano</p>
<p class="six">Wurlitzer Electric Piano</p>
<p class="seven">Wurlitzer Electric Piano</p>
<p class="eight">Wurlitzer Electric Piano</p>
```

```
p.one {border-style: solid;}
p.two {border-style: dotted;}
p.three {border-style: dashed;}
p.four {border-style: double;}
p.five {border-style: groove;}
p.six {border-style: ridge;}
p.seven {border-style: inset;}
p.eight {border-style: outset;}
```

Wurlitzer Electric Piano

Wurlitzer Electric Piano

Wurlitzer Electric Piano

Wurlitzer Electric Piano

Wurlitzer Electric Piano

Wurlitzer Electric Piano

Wurlitzer Electric Piano

Wurlitzer Electric Piano

Border Style Types

none: Defines no border

dotted: Defines a dotted border

dashed: Defines a dashed border

solid: Defines a solid border

double: Defines two borders. The width of the two borders are the same as the border-width value

groove: Defines a 3D grooved border. The effect depends on the border-color value

ridge: Defines a 3D ridged border. The effect depends on the border-color value

inset: Defines a 3D inset border. The effect depends on the border-color value

outset: Defines a 3D outset border. The effect depends on the border-color value

The border-style property can have from one to four values

- **border-style: dotted;**
  - all four borders are dotted

different border styles.

- **border-style: dotted solid;**
  - top and bottom borders are dotted
  - right and left borders are solid

different border styles.

- **border-style: dotted solid double;**
  - top border is dotted
  - right and left borders are solid
  - bottom border is double

different border styles.

- **border-style: dotted solid double dashed;**
  - top border is dotted
  - right border is solid
  - bottom border is double
  - left border is dashed

different border styles.

Border Color border-color Shorthand

```
<p class="one">The ARP Odyssey was introduced in 1972.</p>
<p class="two">The ARP Odyssey was introduced in 1972.</p>
```

The ARP Odyssey was introduced in 1972.

The ARP Odyssey was introduced in 1972.

```
p.one {
        border-color: #0088dd;}
p.two {
        border-color: #bbbbaa #111111 #ee3e80 #0088dd;}
```

```
<p>Here is a simple chord sequence played on a Hammond organ through a Leslie speaker.</p>
```

Here is a simple chord sequence played on a Hammond organ through a Leslie speaker.

```
p {
        width: 250px;
        border: 3px dotted #0088dd;}
```

CSS Border Style

Property	Description
border	Sets all the border properties in one declaration
border-bottom	Sets all the bottom border properties in one declaration
border-bottom-color	Sets the color of the bottom border
border-bottom-style	Sets the style of the bottom border
border-bottom-width	Sets the width of the bottom border
border-color	Sets the color of the four borders
border-left	Sets all the left border properties in one declaration
border-left-color	Sets the color of the left border
border-left-style	Sets the style of the left border
border-left-width	Sets the width of the left border
border-right	Sets all the right border properties in one declaration
border-right-color	Sets the color of the right border
border-right-style	Sets the style of the right border
border-right-width	Sets the width of the right border
border-style	Sets the style of the four borders
border-top	Sets all the top border properties in one declaration
border-top-color	Sets the color of the top border
border-top-style	Sets the style of the top border
border-top-width	Sets the width of the top border
border-width	Sets the width of the four borders

Padding Example

- Padding property allows you to specify how much space should appear between the content of an element and its border

```
<p>Analog synths produce a wave sound,
whereas the sounds stored on a digital
synth have been …</p>
<p class="example">Analog synths produce
a wave sound, whereas the sounds stored
on a digital synth have been …</p>

p {     width: 275px;
        border: 2px solid #0088dd;}

p.example { padding: 10px;}
```

Analog synths produce a wave sound, whereas the sounds stored on a digital synth have been sampled and then turned into numbers.

Analog synths produce a wave sound, whereas the sounds stored on a digital synth have been sampled and then turned into numbers.

Margin Example

- The margin property controls the gap between boxes.

```
<p>Analog synthesizers are often said to have a "warmer" sound than their digital counterparts.</p>
<p class="example">Analog synthesizers are often said to have a "warmer" sound than their digital counterparts.</p>
```

```
p {
        width: 200px;
        border: 2px solid #0088dd;
        padding: 10px;}
p.example {
        margin: 20px;}
```

Analog synthesizers are often said to have a "warmer" sound than their digital counterparts.

Analog synthesizers are often said to have a "warmer" sound than their digital counterparts.

Change inline/block

- The display property allows you to turn an inline element into a block-level or vice versa.

```
<ul>
        <li>Home</li>
        <li>Products</li>
        <li class="coming-soon">Services</li>
        <li>About</li>
        <li>Contact</li>
</ul>
```

Home Products About Contact

```
li {display: inline;
    margin-right: 10px;}
li.coming-soon {display: none;}
```

Hiding

- Visibility property allows you to hide boxes from users but it leaves a space where the element would have been.

```
<ul>
        <li>Home</li>
        <li>Products</li>
        <li class="coming-soon">Services</li>
        <li>About</li>
        <li>Contact</li>
</ul>
```

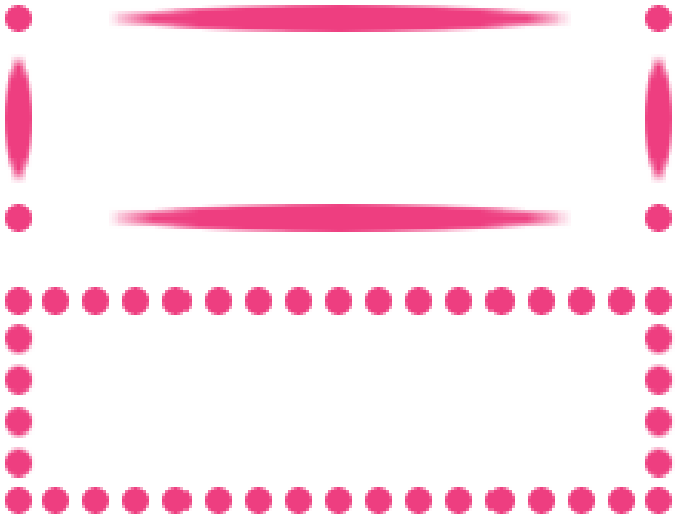
Home Products About Contact

```
li {
        display: inline;
        margin-right: 10px;}
li.coming-soon {
        visibility: hidden;}
```

Border Images

The border-image property applies an image to the border of any box.
The property requires three pieces of information:

1. The url of the image
2. Where to slice the image
3. What to do with the straight edges
    1. Stretch, stretches the image
    2. Round, repeats the image



```
<p class="one"></p>
<p class="two"></p>
```

```
p.one {
border-image: url("images/dots.gif") 11 11 11 11 stretch;}
p.two {
border-image: url("images/dots.gif") 11 11 11 11 round;}
```

Box Shadows

The box-shadow property allows you to add a drop shadow around a box. You must use at least the first two of the next values:

- Horizontal offset
- Vertical offset
- Blur distance
- Spread of shadow

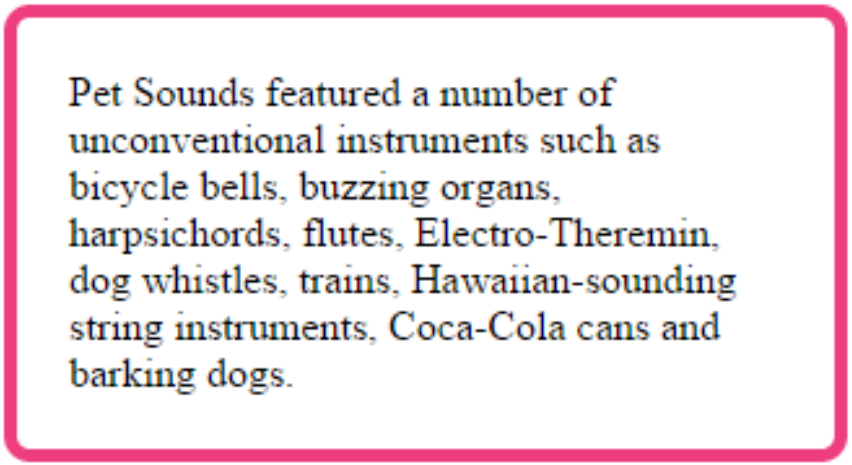
```
p.one { box-shadow: -5px -5px #777777;}
p.two {box-shadow: 5px 5px 5px #777777;}
p.three {box-shadow: 5px 5px 5px 5px #777777;}
p.four {box-shadow: 0 0 10px #777777;}
p.five {box-shadow: inset 0 0 10px #777777;}
```

Rounded Corners border-radius

CSS3 introduces the ability to create rounded corners on any box, using a property called border-radius.

Pet Sounds featured a number of unconventional instruments such as bicycle bells, …

```
p {
        border: 5px solid #ee3e80;
        padding: 20px;
        width: 275px;
        border-radius: 10px;}
```



Elliptical Shapes

To create more complex shapes, you can specify different distances for the horizontal and the vertical parts of the rounded corners.

You can target all corners or just an individual and define the size of radius.

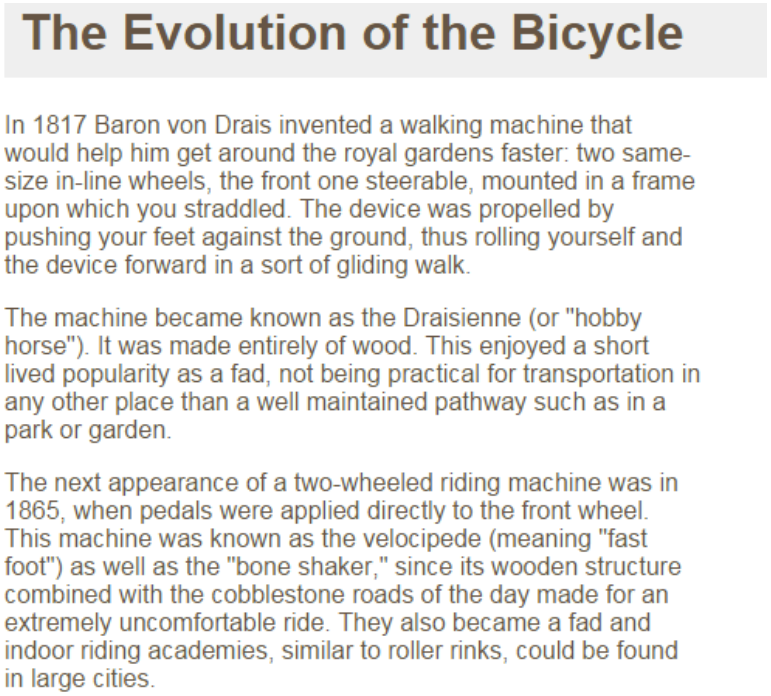
```
p.one {border-top-left-radius: 80px 50px;
p.two {border-radius: 1em 4em 1em 4em / 2em 1em 2em 1em; horiz values / vertic.
p.three {padding: 0px; border-radius: 100px;}
```

```
<p class="one"></p>
<p class="two"></p>
<p class="three"></p>
```

Position: static

- In normal flow, each block-level element sits on top of the next one. This is the default way in which browsers treat HTML, you don't need CSS for that, however this would be:

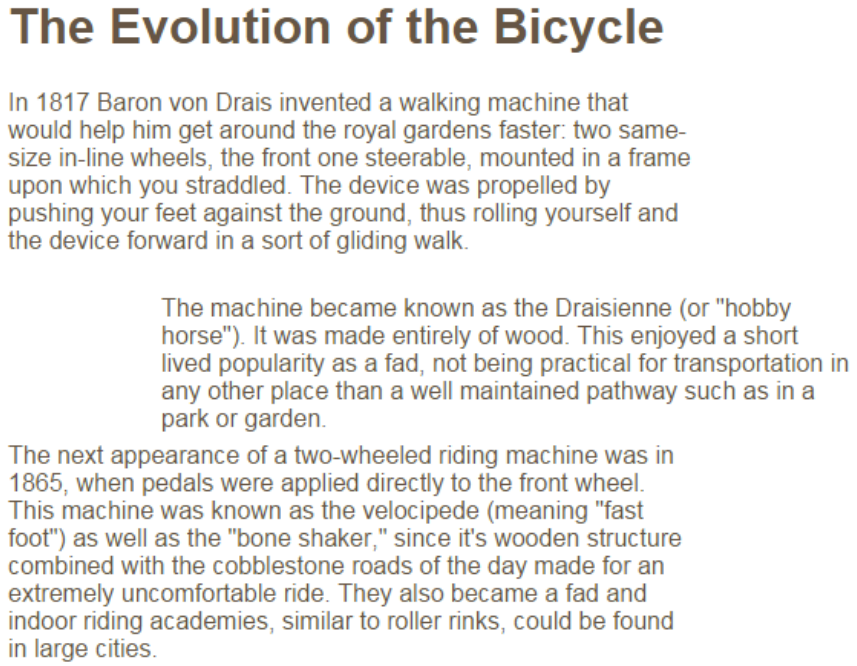
position: static



Position: relative

- Relative positioning moves an element in relation to where it would have been in normal flow.
- You can move it top or bottom and left or right using pixels or presentences.

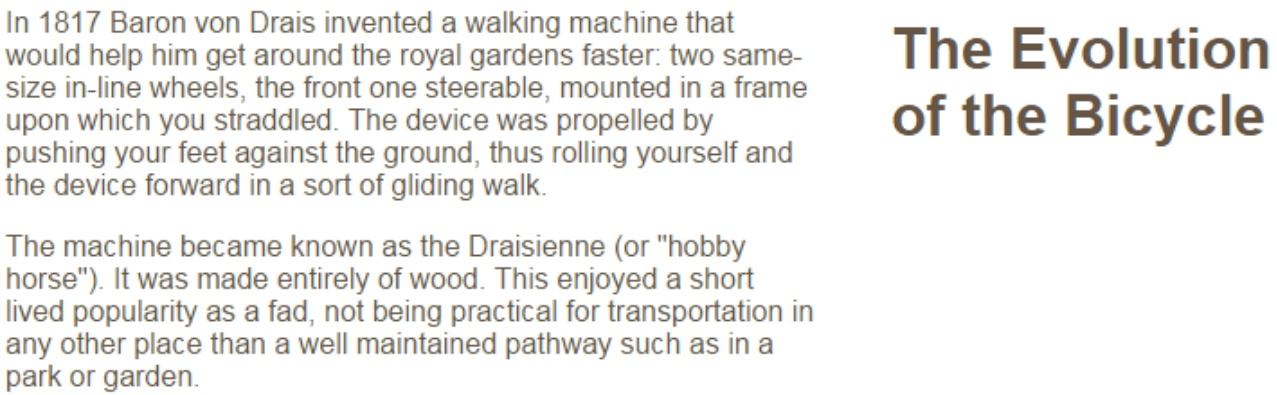
```
p.example {
        position: relative;
        top: 10px;
        left: 100px;}
```



Position: absolute

- When the position property is given a value of absolute, **the box is taken out of normal flow and no longer affects the position of the other elements**.

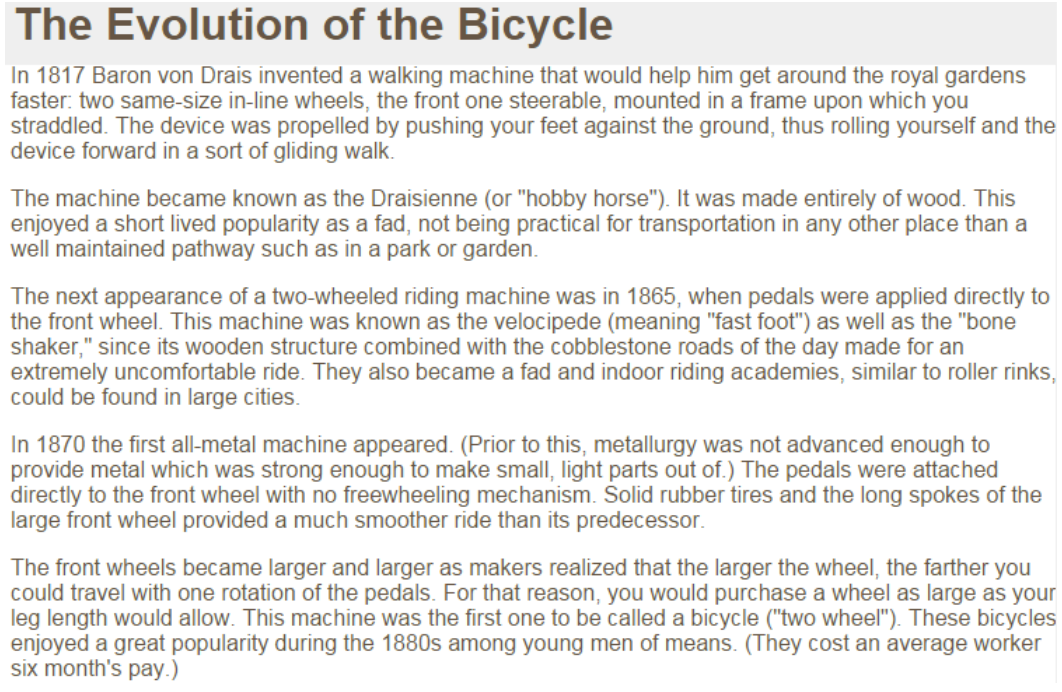
```
h1 {
        position: absolute;
        top: 0px;
        left: 500px;
        width: 250px;}
p {
        width: 450px;}
```



Position: fixed

- Fixed positioning is a type of absolute positioning that requires the position property to have a value of fixed.
- The position is specified in relation to the browser window, as such the user scrolls down, but the element stays on the same place.

```
h1 {
        position: fixed;
        top: 0px;
        left: 0px;
        padding: 10px;
        margin: 0px;
        width: 100%;
        background-color: #efefef;}
p.example {
        margin-top: 100px;}
```



The Evolution of the Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since its wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

In 1870 the first all-metal machine appeared. (Prior to this, metallurgy was not advanced enough to provide metal which was strong enough to make small, light parts out of.) The pedals were attached directly to the front wheel with no freewheeling mechanism. Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

The front wheels became larger and larger as makers realized that the larger the wheel, the farther you could travel with one rotation of the pedals. For that reason, you would purchase a wheel as large as your leg length would allow. This machine was the first one to be called a bicycle ("two wheel"). These bicycles enjoyed a great popularity during the 1880s among young men of means. (They cost an average worker six month's pay.)

Position

Property Value	Description
static	This is the default setting – no special positioning.
absolute	Move element relative to upper left corner of page or a containing element.
relative	Move element relative to its default position.
fixed	Move element relative to browser window – ie doesn't change position if scrolling content.

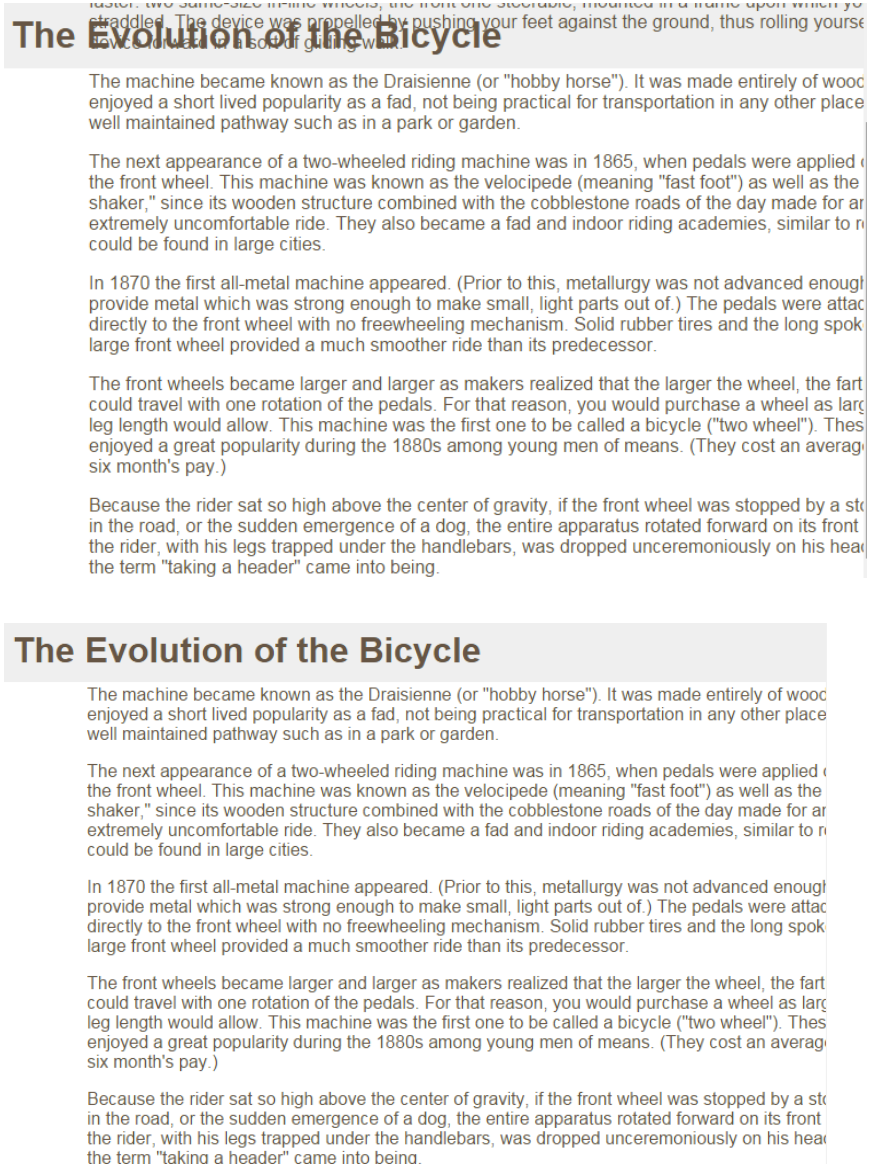
These used in conjunction with the **top**, **right**, **bottom**, **left** offset properties.

Overlapping Elements

- If you use relative, fixed, or absolute positioning, boxes can overlap. If boxes do overlap, the elements that appear later sit one on the top of the other.
- If you want to control which element sits on the top you use the z-index property. The higher value z-index property has the closer that element is to the front.

Overlapping Elements

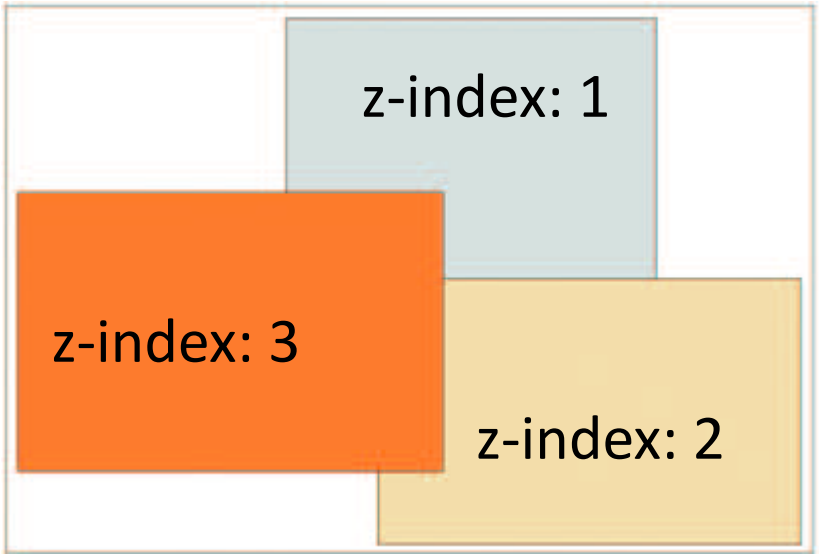
```
h1 {
        position: fixed;
        top: 0px;
        left: 0px;
        margin: 0px;
        padding: 10px;
        width: 100%;
        background-color: #efefef;
        z-index: 10;}
p {
        position: relative;
        top: 70px;
        left: 70px;}
```



Stacking Elements

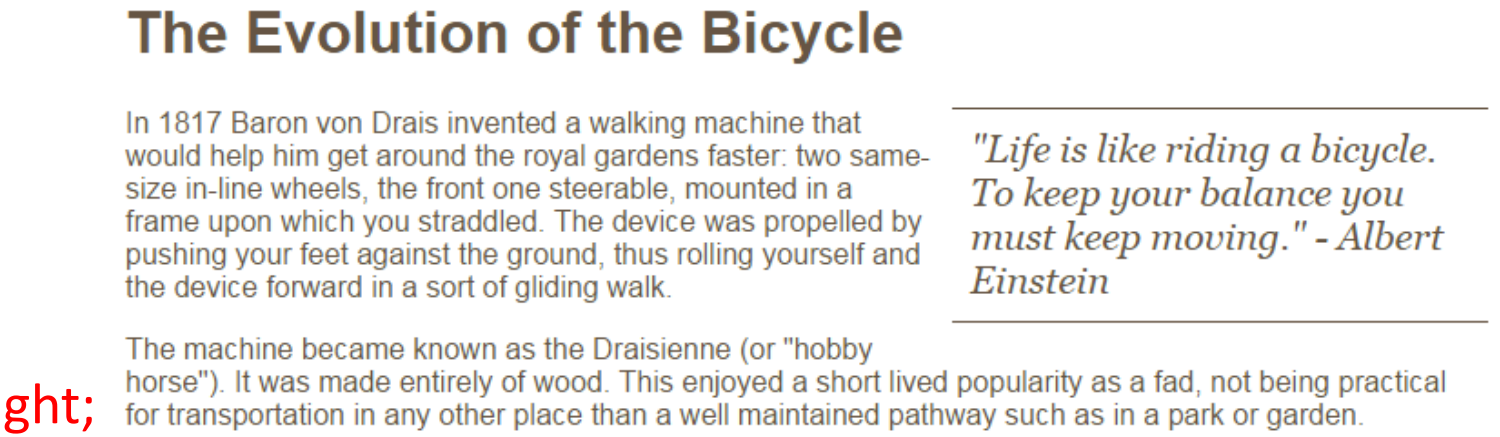
- Specify stacking order with:

– **z-index: value**



Floating Elements

- The float property allows you to take an element in normal flow and place it as far to the left right.




```
blockquote {
        float: right;
        width: 275px;
        font-size: 130%;
        font-style: italic;
        font-family: Georgia, Times, serif;
        margin: 0px 0px 10px 10px;
        padding: 10px;
        border-top: 1px solid #665544;
        border-bottom: 1px solid #665544;}
```

You can use float to place elements side-by-side

Table Properties

- We have already talked several properties that can be used with tables. Here we will put together some of the most commonly used ones, in a single example.
- **Width,** set width
- **Padding,** set the space between the border of each cell
- **Letter-spacing, font-size,** set additional styles
- **Border-top/bottom,** set borders above and below the headers
- **:hover,** to highlight a table row when a user's mouse goes over it

Table Properties

```
<table><tr>
        <th>Author</th>
        <th>Title</th>
        <th class="money">Reserve Price</th>
        <th class="money">Current Bid</th>
</tr><tr>
        <td>E.E. Cummings</td>
        <td>Tulips & Chimneys</td>
        <td class="money">$2,000.00</td>
        <td class="money">$2,642.50</td>
</table>
```

```
body {
        font-family: Arial, Verdana, sans-serif;
        color: #111111;}
table {  width: 600px;}
th, td {  padding: 7px 10px 10px 10px;}
th {
        text-transform: uppercase;
        letter-spacing: 0.1em;
        font-size: 90%;
        border-bottom: 2px solid #111111;
        border-top: 1px solid #999;
        text-align: left;}
tr.even {  background-color: #efefef;}
r:hover {  background-color: #c3e6e5;}
.money {  text-align: right;}
```

First Edition Auctions

AUTHOR	TITLE	RESERVE PRICE	CURRENT BID
E.E. Cummings	Tulips & Chimneys	$2,000.00	$2,642.50
Charles d'Orleans	Poemes		$5,866.00
T.S. Eliot	Poems 1909 - 1925	$1,250.00	$8,499.35
Sylvia Plath	The Colossus		$1031.72

Tips for tables

- Give cells padding
- Distinguish headings
- Shade alternate rows
- Align numerals

Controlling Size of Images using CSS

- Control the size of an image using width and height properties.

```
<img src="images/magnolia-large.jpg" class="large" alt="Magnolia" />
<img src="images/magnolia-medium.jpg" class="medium" alt="Magnolia" />
<img src="images/magnolia-small.jpg" class="small" alt="Magnolia" />
```

```
img.large {
        width: 500px;
        height: 500px;}
img.medium {
        width: 250px;
        height: 250px;}
img.small {
        width: 100px;
        height: 100px;}
```

Align Images with CSS

- Use float to align images

```
<p><img src="images/magnolia-medium.jpg" alt="Magnolia" class="align-left medium" /><b><i>Magnolia is …</p>

<p><img src="images/magnolia-medium.jpg" alt="Magnolia" class="align-right medium" />Some magnolias…</p>
```

```
img.align-left      {
                    float: left;
                    margin-right: 10px;}
img.align-right     {
                    float: right;
                    margin-left: 10px;}
img.medium {

                    width: 250px;
                    height: 250px;}
```

Magnolia is a large genus that contains over 200 flowering plant species. It is named after French botanist Pierre Magnol and, having evolved before bees appeared, the flowers were developed to encourage pollination by beetle.

Some magnolias, such as *Magnolia stellata* and *Magnolia soulangeana*, flower quite early in the spring before the leaves open. Others flower in late spring or early summer, such as *Magnolia grandiflora*.

Display style

selector {display:**Value**}

Value	Description
None	The element will generate no box at all.
block	The element will generate a block box (a line break before and after the element).
Inline	The element will generate an inline box (no line break before or after the element). This is default.
inline-block	The element will generate a block box, laid out as an inline box.
inline-table	The element will generate an inline box (like <table>, with no line break before or after).
inherit	Specifies that the value of the display property should be inherited from the parent element.

Display style cont.

Value	Description
list-item	The element will generate a block box, and an inline box for the list marker
run-in	The element will generate a block or inline box, depending on context
table	The element will behave like a table (like <table>, with a line break before and after)
table-caption	The element will behave like a table caption (like <caption>)
table-cell	The element will behave like a table cell
table-column	The element will behave like a table column
table-column-group	The element will behave like a table column group (like <colgroup>)
table-footer-group	The element will behave like a table footer row group
table-header-group	The element will behave like a table header row group
table-row	The element will behave like a table row
table-row-group	The element will behave like a table row group

Summary

- How to create style sheets to control the style and layout of web sites.
- How to use CSS to add backgrounds, format text, add and format borders, and specify padding and margins of elements.
- How to position an element, control the visibility and size of an element, set the shape of an element, place an element behind another, and to add special effects to some selectors, like links.
- How many of the new features in CSS3: rounded borders, box and text shadows, gradient backgrounds and more.

P0 and P1

P0 Getting Started (Sept 15) and

P1 Project Requirements (Sept 20)

Questions?

NTNU
Norwegian University of
Science and Technology

Department of
Computer and Information Science