

EECS1012

Net-centric Introduction to Computing

Lecture 4: More CSS

Acknowledgements

Contents are adapted from web lectures for “Web Programming Step by Step”, by M. Stepp, J. Miller, and V. Kirst. Slides have been ported to PPT by Dr. Xenia Mountroudou.

These slides have been edited for EECS1012, York University.

The contents of these slides may be modified and redistributed, please give appropriate credit.

(Creative Commons) Michael S. Brown, 2018.

2

More CSS

HTML id attribute

3

```
<p>Coding Horror! Coding Horror!</p>  
<p id="mission">Our mission is to combine programming and  
<q>human</q> factors with geekiness!</p>
```

HTML

- A unique ID for an element on a page
- In the above example, the ID is now associate with the 2nd paragraph
- **Each ID must be unique; can only be used once in the page**



CSS ID selectors

4

```
#mission {  
  font-style: italic;  
  font-family: "Garamond", "Century Gothic", serif;  
}
```

CSS

Coding Horror! Coding Horror!

Our mission is to combine programming and “human” factors with geekiness!
output

- In this case, the CSS is using #mission to refer to the element with the id “mission” on the previous slide
- This CSS applies style only to the paragraph that has the ID of mission

ID selectors vs. anchors

5



```
<p>Visit <a href=
"http://www.textpad.com/download/index.html#downloads">
textpad.com</a> to get the TextPad editor.</p>
<p><a href="#mission">View our Mission Statement</a></p>
```

HTML

Visit [textpad.com](http://www.textpad.com) to get the TextPad editor.

output

- ❑ **Don't confuse ID selectors with anchors**
- ❑ Link target can include an ID at the end, preceded by a #
- ❑ Browser will load that page and scroll to element with given ID
- ❑ ID's are not anchors!

HTML class attribute

6

```
<p class="shout">Coding Horror! Coding Horror!</p>  
<p class="special">See our special deal on Droids!</p>  
  
<p class="special">Today only!</p>
```

HTML

Coding Horror! Coding Horror!

See our special deal on Droids!

Today only!

output

- A way to **group several elements** and give a style to only that group
- **Unlike an id**, a class can be reused as much as you like on the page

CSS class selectors

7

```
.special { /* we denote class selector using a period */  
background-color: yellow;  
font-weight: bold;  
}  
p.shout { /* this means all paragraph elements with */  
color: red; /* class shout */  
font-family: sans-serif;  
}
```

CSS

Coding Horror! Coding Horror!

See our special deal on Droids!

Today only!

output

You can apply more than one style

8

```
<p class="shout">Coding Horror! Coding Horror!</p>
<p class="special">See our special deal on Droids!</p>

<p class="special shout">Today only!</p>
```

HTML

This `<p>` element is associated with two class styles – special and shout (defined on previous slide). A space is used between the class names.

Coding Horror! Coding Horror!

See our special deal on Droids!

Today only!

output

Recap: selector notation

9

```
#selectorA { /* apply style to an element with */
...         /* id="selectorA", there should be */
}           /* only ONE element in the HTML with this ID */

.selectorB { /* apply style to all elements that have */
...         /* class="selectorB" */
}

selectorC.selectorD { /* apply style to all elements of */
..           /* class="selectorD" that are */
}           /* of type selectorC */

selectorA, selectorB, selectorC { /* you can apply a style to */
...                               /* multiple selectors */
}                               /* with a , between selectors*/
```

CSS

Quick example

10

```
<h1 id="top" > This is the top h1. </h1>
<h1 class="bigRed" > This is class big red. </h1>
<h3 class="bigRed" > This is class big red. </h3>
<p class="bigRed" > This is class big red. </p>
<p class="larger" > This is class larger </p>
```

HTML

```
#top      { /* apply style to an element with */
...        /* with id=top, there is only 1 above */
}
.bigRed   { /* apply style to all elements of class */
...        /* .bigRed */
}          /* multiple elements use this class */

p.larger  { /* apply style to all elements of */
...        /* class "larger" only if it is an */
}          /* element paragraph */
```

CSS

CSS pseudo-class

11

```
a:link { color: #FF0000; }      /* unvisited link */  
a:visited { color: #00FF00; }  /* visited link */  
a:hover { color: #FF00FF; }    /* mouse over link */
```

CSS

Buy Early Buy Often!

output

- Pseudo-class is used to define a special state of an element
- Great example is for hyper links `<a>` elements that can have multiple states (e.g., before visiting the link, after visiting the link, when the mouse is hovering over the link).

More CSS pseudo classes

12

class	description
:active	an activated or selected element
:focus	an element that has the keyboard focus
:hover	an element that has the mouse over it
:link	a link that has not been visited
:visited	a link that has already been visited
:first-letter	the first letter of text inside an element
:first-line	the first line of text inside an element
:first-child	an element that is the first one to appear inside another

Styling tables

13

```
table { border: 2px solid black; caption-side: bottom; }  
tr { font-style: italic; }  
td { background-color: yellow; text-align: center; width: 30%;  
}
```

CSS

<i>Column 1</i>	<i>Column 2</i>
<i>1,1</i>	<i>1,2 okay</i>
<i>2,1 real wide</i>	<i>2,2</i>

My important data

output

- all standard CSS styles can be applied to a table, row, or cell
- table specific CSS properties:
border-collapse, border-spacing, caption-side, empty-cells, table-layout
- See: https://www.w3schools.com/css/css_table.asp

The rowspan and colspan attributes

14

```
<table>
  <tr><th>Column 1</th><th>Column 2</th><th>Column 3</th></tr>
  <tr><td colspan="2">1,1-1,2</td>
    <td rowspan="3">1,3-3,3</td></tr>
  <tr><td>2,1</td><td>2,2</td></tr>
  <tr><td>3,1</td><td>3,2</td></tr>
</table>
```

HTML

Column 1	Column 2	Column 3
1,1-1,2		1,3-3,3
2,1	2,2	
3,1	3,2	

output

- ❑ colspan makes a cell occupy multiple columns; rowspan multiple rows
- ❑ This is not part of the CSS, but requires modifications to the HTML ..

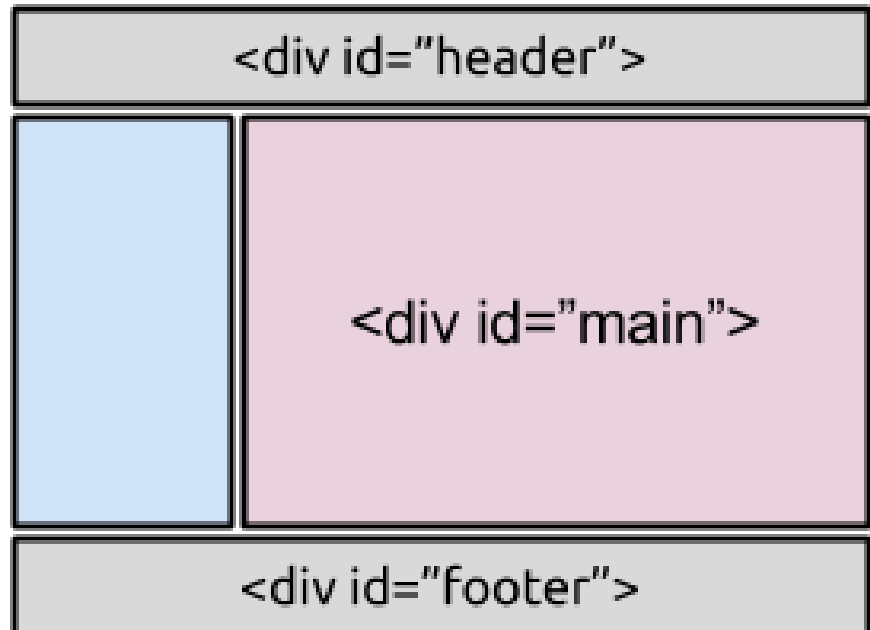
15

Styling Page Sections

Why do we need page sections?

16

- Style individual elements, groups of elements, sections of text or of the page
- Create complex page layouts



Sections of a page `<div>`

17

```
<div class="shout">  
<h2>Coding Horror! Coding Horror!</h2>  
<p class="special">See our special deal on Droids!</p> <p>We'll  
beat any advertised price!</p>  
</div>
```

HTML

Coding Horror! Coding Horror!

See our special deal on Droids!

We'll beat any advertised price!

- Tag used to indicate a logical section or area of a page
- In this example, we have “wrapped” the elements in a div.
- We can then apply a style to the whole div. In the example above, Div gets style of class “shout”.

```
shout { color: red;  
font-family: sans-serif; }
```

Inline sections ``

18

```
<h2>Coding Horror! Coding Horror!</h2>
<p>See our <span class="special">spectacular</span> deal on
Droids!</p>
<p>We'll beat <span class="shout"> any advertised
price</span>!</p>
```

HTML

Coding Horror! Coding Horror!

See our **spectacular** deal on Droids!

We'll beat **any advertised price!**

output

- Span is like div, but intended for inline elements. In this case, we have placed a span about one word “spectacular”.

CSS context selectors

19

```
selector1 selector2 {  
    properties  
}
```

CSS

- applies the given properties to selector2 **only if** it is inside a selector1 on the page

```
selector1 > selector2 {  
    properties  
}
```

CSS

- applies the given properties to selector2 only if it is *directly* inside a selector1 on the page (*that is selector2 is not defined within another tag after selector 1 – yes, it is confusing, see next slide for an example*)

Example *context selector* example

20

```
<p>Eat at <strong>Greasy's Burger</strong>...</p>
<ul>
<li>The <strong>greasiest</strong> burgers in town!</li>
<li>Yummy and greasy at the same time!</li>
</ul>
```

HTML

```
li strong { text-decoration: underline; }
```

CSS

Eat at **Greasy's Burger...**

- The **greasiest** burgers in town!
- Yummy and greasy at the same time!

output

Another example

21

```
<p> This is a <strong> test </strong> of the context selector. </p>  
<p> This is a <em> <strong> test </strong> </em> of the context  
selector. </p>
```

HTML

```
strong{  
    color: red;  
}  
p > strong {          /* only immediate "child" of <p> */  
    color: blue;  
    background-color: yellow;  
}
```

CSS

This is a **test** of the context select.

This is another **test** of the context selector.

/* this second test is inside an so it
is not directly inside the <p> </p> tag, so the
context selector does not apply. */

output

Another example

22

```
<p> This is a <strong> test </strong> of the context selector. </p>  
<p> This is a <em> <strong> test </strong> </em> of the context  
selector. </p>
```

HTML

```
strong{  
    color: red;  
}  
p strong { /* only immediate "child" of <p> */  
    color: blue;  
    background-color: yellow;  
}
```

CSS

This is a **test** of the context select.

This is another **test** of the context selector.

/* This selector means apply the style rule to
any element
appearing within a <p> */

output

Yet another (more complex) example

23

```
<div id="ad">
<p>Eat at <strong>Greasy's Burger</strong>...</p>
<ul>
<li class="important">The <strong>greasiest</strong>
burgers in town!</li>
<li>Yummy and <strong>greasy at the same time
</strong>!</li>
</ul>
</div>
```

HTML

```
#ad li.important strong { text-decoration: underline; }
```

CSS

Eat at **Greasy's Burger...**

- The **greasiest** burgers in town!
- Yummy and **greasy at the same time!**

output

More complex example (con't)

24

```
#ad li.important strong { text-decoration: underline; }
```

CSS

This selector means (reading it backwards):

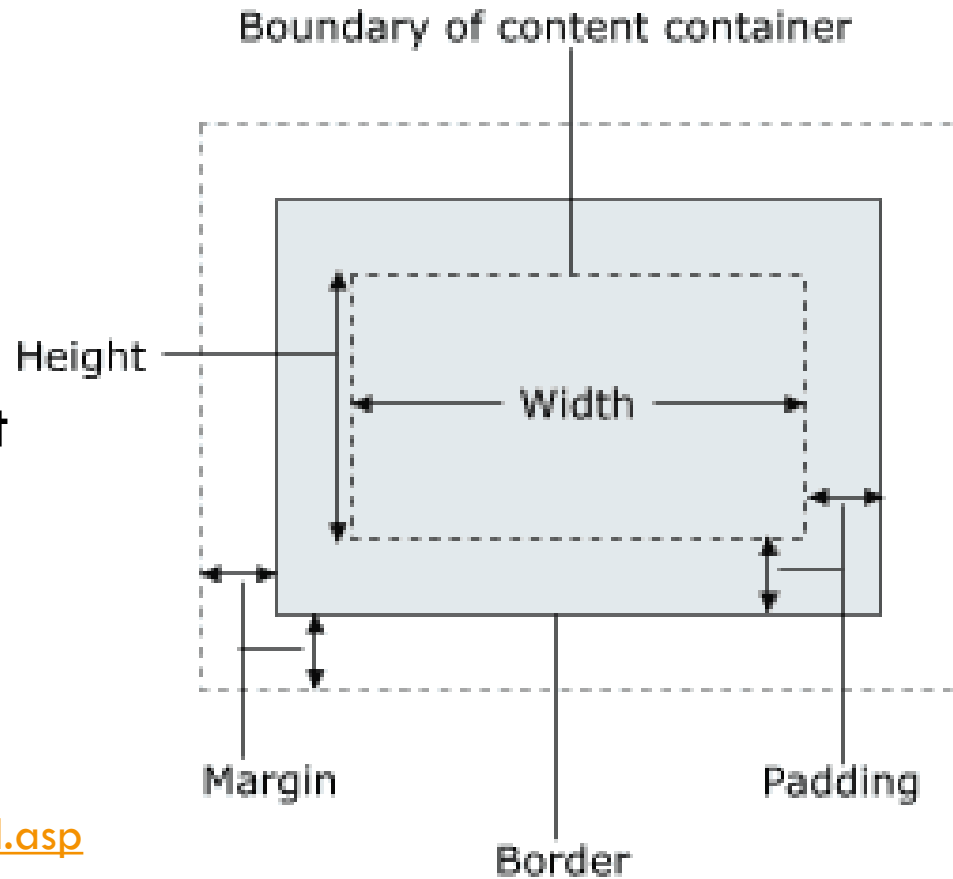
Apply to an element of type strong,
that is inside a element that is class “important”,
that is inside an element that has id=“ad”.

- Yes, this is crazy – **but**, this example shows you just how specific the CSS can be about presentation style of an HTML page.

The CSS Box Model

25

- Every element is composed of:
 - ▣ content
 - ▣ a **border** around the element
 - ▣ **padding** between the content and the border
 - ▣ a **margin** between the border and other content



https://www.w3schools.com/css/css_boxmodel.asp

The CSS Box model (cont.)

26

- **width** = content width + L/R padding + L/R border + L/R margin
- **height** = content height + T/B padding + T/B border + T/B margin
- And of course, IE6 doesn't do this right



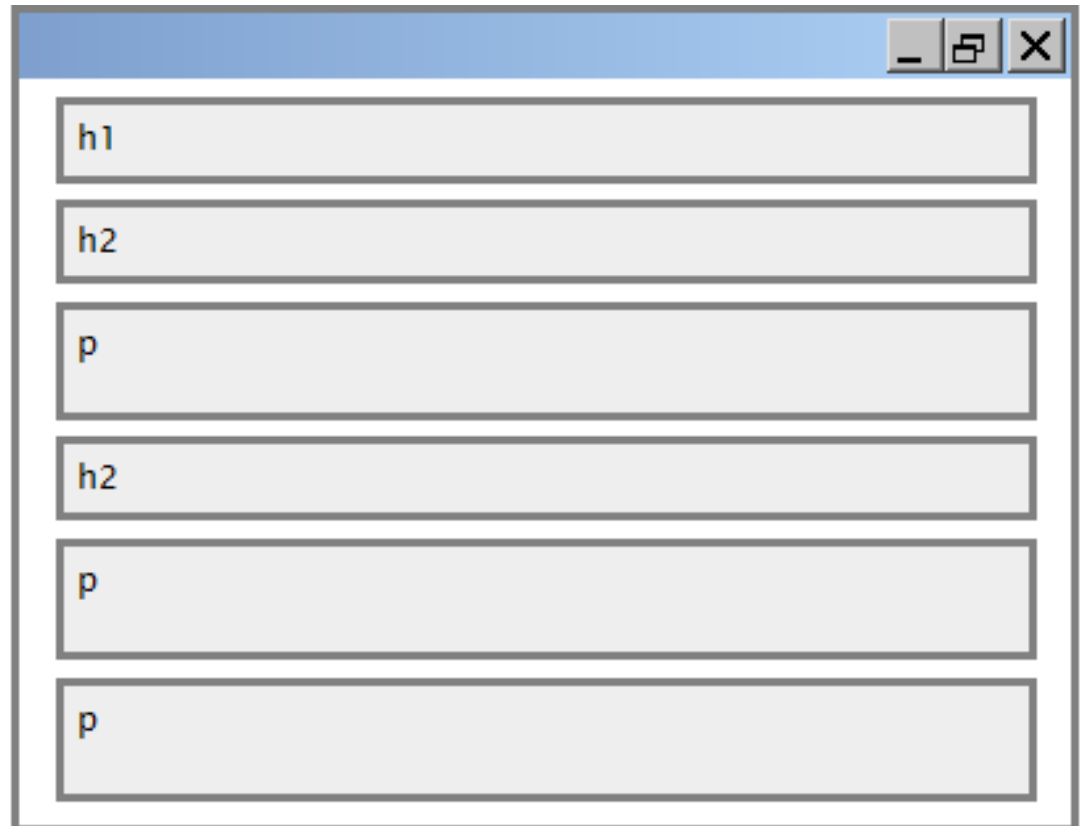
*L/R means left and right padding
(not divide L by R)

Document Flow – block elements

27

We can think of each element as a “block” that flows on the page.

The default behavior is for blocks is not to overlap.

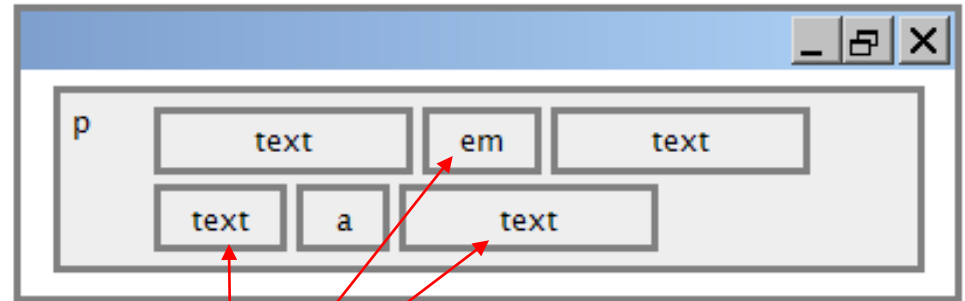


Document flow - inline elements

28

We can think of inline elements as being contained within a block element. They flow as if they are inline text.

The default is that inline elements will not overlap within the block they are contained in.

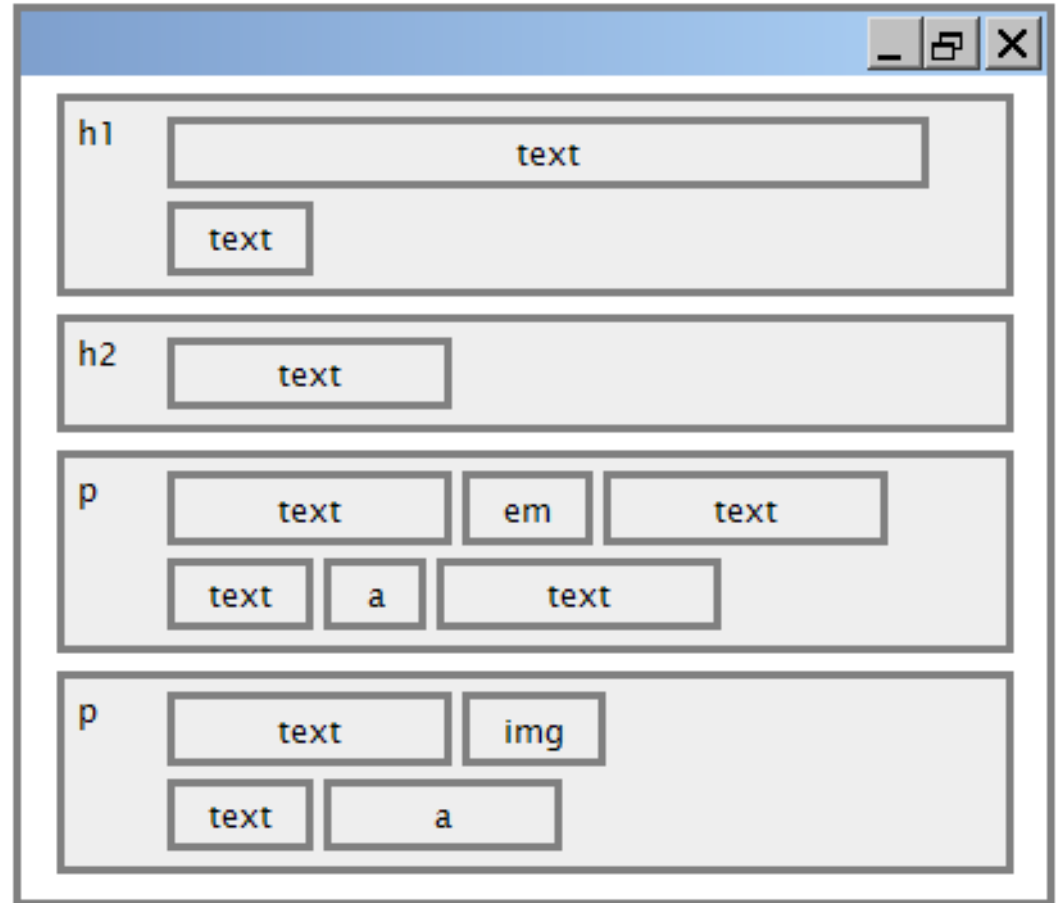


Inline elements within a `<p>` block element.

Document flow - a larger example

29

A more accurate view is to visualize our page as blocks containing inline elements.



CSS properties for borders

30

```
h2 { border: 5px solid red; }
```

CSS

This is a heading.

output

property	description
border	thickness/style/size of border on all 4 sides

- **Thickness:** px, pt, em, or thin, medium, thick
- **Style:** none, hidden, dotted, dashed, double, groove, inset, outset, ridge, solid
- **color** (specific the same as other elements)

Border styles examples

31

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

More border properties

32

property	description
border-color, border-width, border-style	specific properties of border on all 4 sides
border-bottom, border-left, border-right, border-top	all properties of border on a particular side
border-bottom-color, border-bottom-style, border-bottom-width, border-left-color, border-left-style, border-left-width, border-right-color, border-right-style, border-right-width, border-top-color, border-top-style, border-top-width	properties of border on a particular side
Complete list of border properties https://www.w3schools.com/css/css_border.asp	

Another border example

33

```
h2 {  
  border-left: thick dotted #CC0088;  
  border-bottom-color: rgb(0, 128, 128);  
  border-bottom-style: double;  
}
```

CSS

This is a heading.

output

- each side's border properties can be set individually
- if you omit some properties, they receive the default

CSS properties for padding

34

property	description
padding	padding on all 4 sides
padding-bottom	padding on bottom side only
padding-left	padding on left side only
padding-right	padding on right side only
padding-top	padding on top side only
Complete list of padding properties https://www.w3schools.com/css/css_padding.asp	

Padding example 1

35

```
<!DOCTYPE html>
<html>
<head>
<style>
    p { padding: 20px; border: 3px solid black; }
    h2 { padding: 0px; background-color: yellow; }
</style>
</head>
<body>


<h2> Test header style </h2>
<p> Test paragraph style </p>

</body>
</html>
```

Padding example 1 output

36

Test header style



Test paragraph style

The diagram consists of a yellow rectangular header at the top. Below it is a white rectangular container with a black border. Inside this container, the text 'Test paragraph style' is positioned. A blue arrow points from the bottom-left corner of the container to the text, indicating the padding between the container's border and the text.

Content has a 20px padding.

The content above has no padding (or padding of 0px).

CSS properties for margins

37

property	description
margin	margin on all 4 sides
margin-bottom	margin on bottom side only
margin-left	margin on left side only
margin-right	margin on right side only
margin-top	margin on top side only
Complete list of margin properties https://www.w3schools.com/cssref/pr_margin.asp	

Margin example 1

38

```
p {  
    margin: 50px;  
    background-color: fuchsia;  
}
```

CSS

This is a first paragraph

This is a second paragraph

output

- notice that margins are always transparent (they don't contain the elements background properties).

Margin example 2

39

```
p {  
  margin-left: 8em;  
  background-color: fuchsia;  
}
```

CSS

This is a first paragraph

This is a second paragraph

output

- each side's margin can be set individually
- above only the left margin has been set (to 8x the current font size)

CSS properties for dimensions

40

```
p { width: 350px; background-color: yellow; }  
h2 { width: 50%; background-color: aqua; }
```

CSS

This paragraph uses the first style above

An h2 heading

output

property	description
width, height	how wide or tall to make this element (block elements only)
max-width, max-height, min-width, min-height	max/min size of this element in given dimension

Centering a block element: auto margins

41

```
p {  
  margin: auto;  
  width: 300px;  
}
```

CSS

This is a first paragraph

output

- This `<p>` element is size 300px and margins are auto. This gives equal margins on both side. Note that the text content is not centered.
- to center inline elements within a block element, use `text-align: center;`

Recap:

42

- You now know enough about CSS to create nice webpages and understand the complexity of the “context selector”.
- This will be useful if you download a complex CSS and try to modify it (or . . . create your own)
- You also know the CSS box model regarding padding, margin, border, and content.