



School of Computer Science

Web and Database Computing 2019

Lecture 5: CSS Layout

adelaide.edu.au

seek LIGHT

More CSS

You can follow along in the lecture slides,
but also following the guide at <https://www.w3schools.com/css/>

CSS Precedence

What to do when multiple selectors apply?

- Selectors are chosen based on specificity e.g.

```
h1.main div { ... }
```

is more specific than

```
div { ... }
```

HTML CSS Result

[Edit in JSFiddle](#)

```
<div>  
<div style="padding: 1em;">A div</div>  
</div>
```

CSS Precedence

- Where selectors have the same specificity, the newest one is chosen e.g.

```
div { color: black; }  
div { color: blue; }
```

The second rule (blue) will apply.

- CSS applied inline (using a style attribute on a specific element) has the highest specificity.

HTML CSS Result

[Edit in JSFiddle](#)

```
<div style="padding: 1em;">A div</div>
```

Size Units

CSS supports a number of different units that can either specify an absolute or relative size.

Absolute size units include

- **cm** centimeters
- **mm** millimeters
- **in** inches (1in = 96px = 2.54cm)
- **px** pixels
- **pt** points (1pt = 1/72 of 1in)

Relative units include

- **em** Relative to the font-size of the element (2em means 2 times the size of the current font)
- **%** Relative to the parent element
- Other relative units that are less commonly used and supported include
ex ch rem vw vh vmin vmax

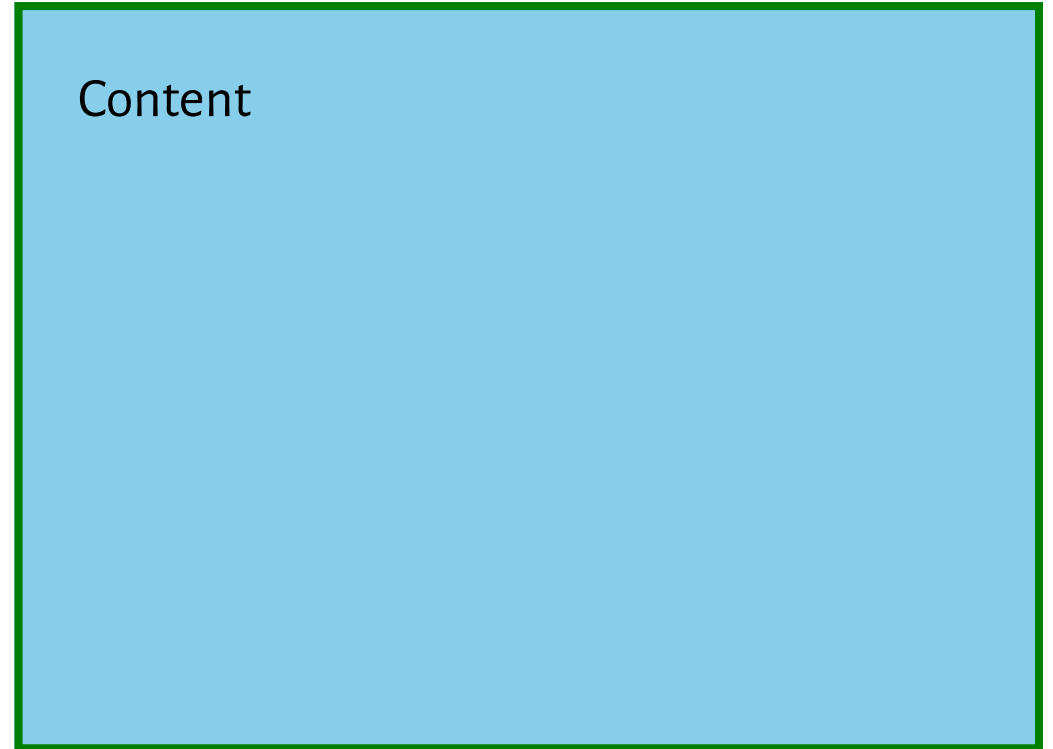
Inspecting and debugging CSS using your browser

Demo

Element Positioning and Page Layout

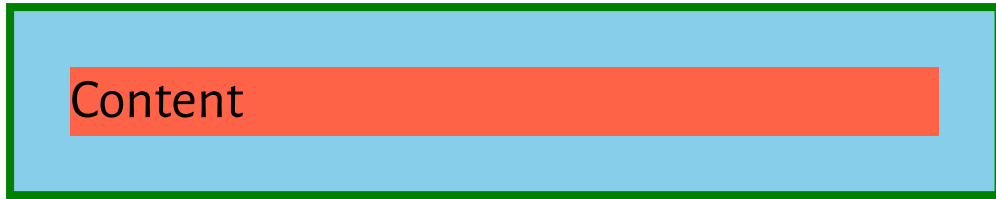
Box Model

- Content
- Padding
- Borders
- Margins



Content

- The area where content actually appears.



- By default width & height are measured to the content area only.
- Specify size using width & height properties:

```
div {  
  width: 10em;  
  height: 4em;  
}
```

Padding

- The area between the border of an object and the content:



- No padding vs padding:



- Takes up space in addition to the width of the element.

Specifying padding

- Specify size using padding property:

```
div {  
  padding: 1em;  
}
```

- Can also specify individual paddings for each side of the element:

```
div {  
  padding-top: 1em;  
  padding-right: 2em;  
  padding-bottom: 3em;  
  padding-left: 4em;  
}
```

- Or shortened (note clockwise order):

```
div {  
  padding: 1em 2em 3em 4em;  
}
```

Borders

- A line around the outside of the object:



- Also takes up space in addition to the width of the element.
- Borders have several properties:
 - width
Measured in standard measurement units
 - style
solid vs dashed vs double
 - color
Standard color (name, rgb(a), hex, hsl(a))

Specifying Borders

- Specify size using border property and each of the sub-properties:

```
div {  
  border: 1px solid red;  
}
```

- Can also specify individual borders for each side of the element:

```
div {  
  border-top: 1px solid red;  
  border-right: 2px dashed #00ffff;  
  border-bottom: 3px double rgba(255,128,128,50%);  
  border-left: none;  
}
```

- Rounded corners with border-radius property:

```
div {  
  border-radius: 5px;  
}
```

Margins

- Space between the element and neighbouring elements:



- Not included in the width of the element.

Specifying margin

- Same as for padding:

```
div {  
  margin: 1em;  
}
```

- Again, can specify individual margins for each side of the element:

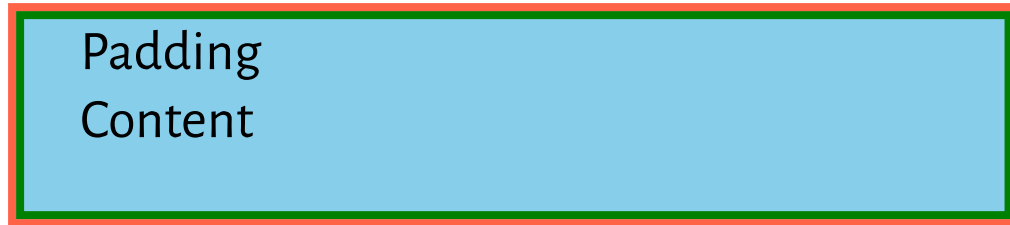
```
div {  
  margin-top: 1em;  
  margin-right: 2em;  
  margin-bottom: 3em;  
  margin-left: 4em;  
}
```

- Or shortened (again, clockwise order):

```
div {  
  margin: 1em 2em 3em 4em;  
}
```

Outlines

- A line drawn outside of the border to make an element stand out



- Specify same as border.

Box Sizing

- Width and height only specify content area, but often we want size to include padding and borders.
- We can get around this using the **box-sizing** property
 - The **content-box** value is the default. width and height are the content area only
 - The **border-box** value changes width and height to include padding and borders

HTML Result

[Edit in JSFiddle](#)

```
<div style="height: 5em; padding: 2em;">
  content-box
</div>
<br />
<div style="box-sizing: border-box; height: 5em; padding: 2em"
  border-box
</div>
```

Positioning

Elements can be moved on the page using the top, left, bottom and right properties.

The position property specifies the method by which an element is positioned.

- **static** (default) the element is positioned normally within the page's structure.
- **relative** the element's position is modified relative to its normal position
- **absolute** the element's position is modified parent elements that have been positioned.
- **fixed** the element's position is set relative to the viewport (window).

Floats

Elements can be separated from the main flow of content in a container using the **float** property.

- The **float** property can push elements to the **left** or **right** of their container, causing inline content to 'wrap' around the floating content.
- The **clear** property can force non-floating content to appear beneath the floating content instead of wrapping. Can be set to **left**, **right** or **both**;



Display

The **display** property allows us to specify how an object will appear and behave on the page.

- The default value depends on the element
- **none** The element is hidden and takes up no space on the page.
- **inline** The element takes up only as much space as needed and does not start on a new line. Width and height cannot be set.
- **block** The element starts on a new line and fills the width of the available space.
- **inline-block** The element does not start on a new line, but can have a set width and height.

HTML CSS Result

[Edit in JSFiddle](#)

```
<div>block</div>
<div>block</div>
<br />
<div style="display: inline;">inline</div>
<div style="display: inline;">inline</div>
<br />
<div style="display: inline-block;">inline-block</div>
<div style="display: inline-block;">inline-block</div>
```

What's happening

Due:

- Prac Exercise 1 should be completed by tonight.
- Prac Exercise 2 will be released tonight.
 - Websub to follow early next week.

Next week:

- Monday we will start to make our pages interactive with Javascript.
- Workshops next week.

Further learning:

- Work through the CSS tutorial at <https://www.w3schools.com/css/>
- Try looking at the layout of the pages you use, inspecting elements to see how it's done.
- Make your forum posts *pretty*!



THE UNIVERSITY
of ADELAIDE

