THE UNIVERSITY of ADELAIDE

Web and Database Computing

adelaide.edu.au

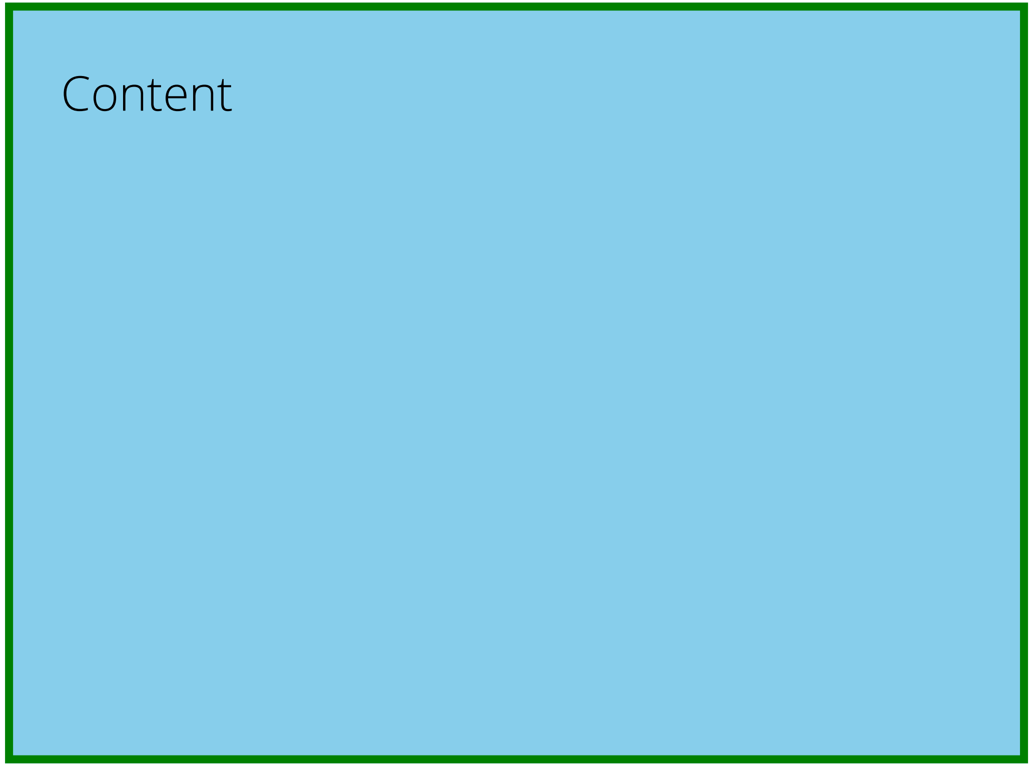
Style & CSS: CSS Layout

Element Positioning and Page Layout

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

Box Model

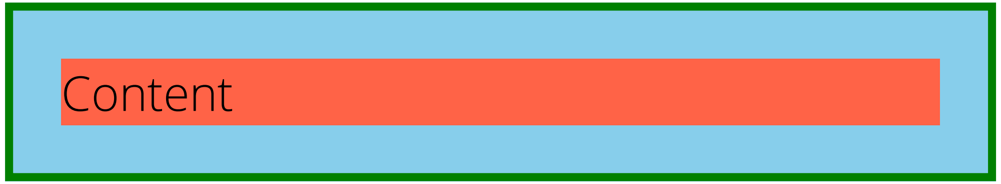
- Content
- Padding
- Borders
- Margins



Content

Content

- The area where content actually appears.

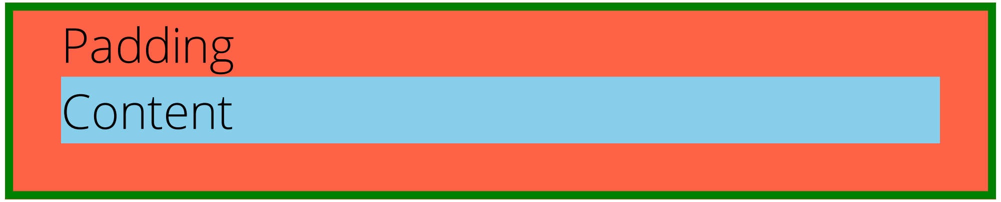


- By default width & height are measured to the content area only.
- Specify size using width & height properties:

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

Padding

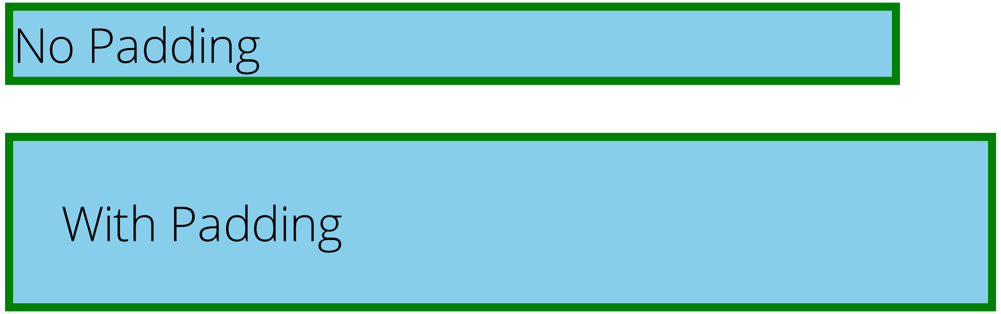
- The area between the border of an object and the content:



Padding

Content

- No padding vs padding:



No Padding

With 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:

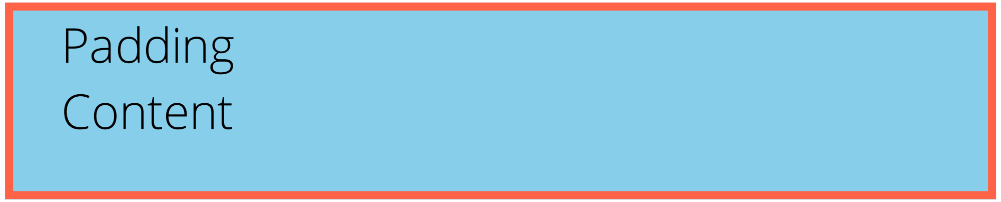
```
div {
  padding-top:    1em;
  padding-right:  2em;
  padding-bottom: 3em;
  padding-left:   4em;
}
```

- Or shortened (note clockwise order):

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

Borders

- A line around the outside of the object:



Padding
Content

- 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:

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

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

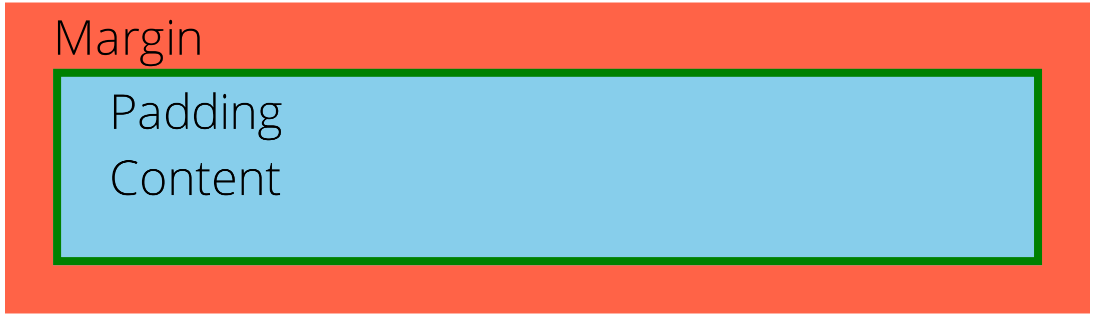
```css
div {
  border-top:    1px solid red;
  border-right:  2px dashed #00ffff;
  border-bottom: 3px double rgba(255,128,128,50%);
  border-left:   none;
}
```

- Rounded corners with border-radius property:

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

Margins

- Space between the element and neighbouring elements:



Margin

Padding

Content

- 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:

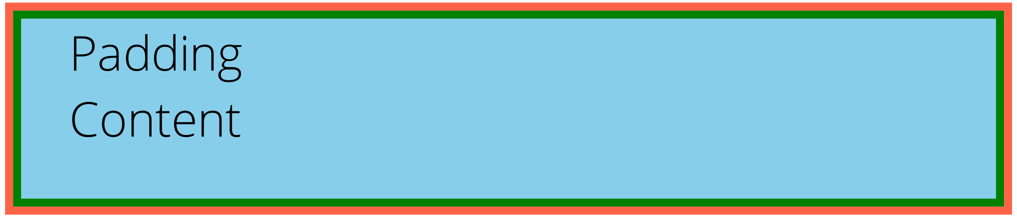
```
div {
  margin-top:    1em;
  margin-right:  2em;
  margin-bottom: 3em;
  margin-left:   4em;
}
```

- 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



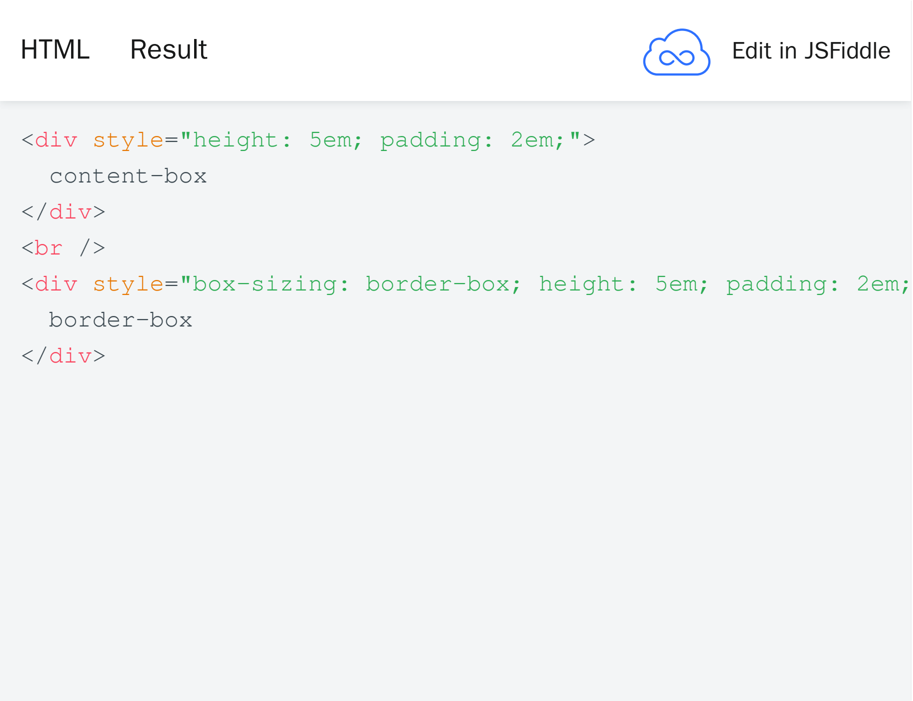
Padding

Content

- Specify same as border.
- Does not affect size of the element.

Box Sizing

- By default, 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
<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.

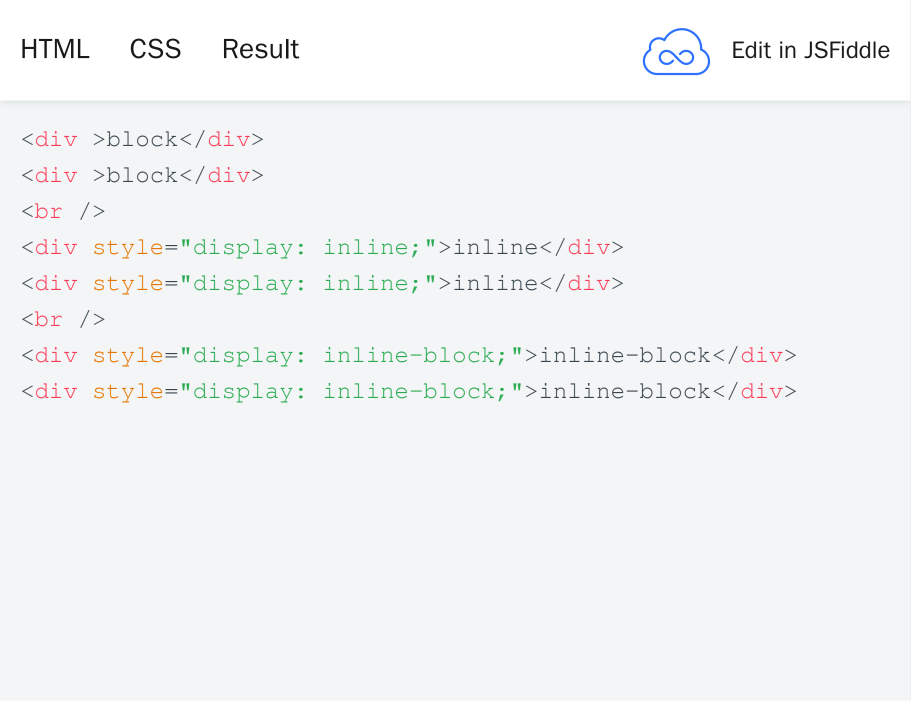
- The `float` property can push elements to the `left` or `right` of their container, causing inline content to 'wrap' around the floating content.

- 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
<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>
```

Flexbox & Grid

Newer developments in display/positioning have been the introduction of CSS Grid & CSS Flexbox.

- This allows us to layout our webpages in a more responsive way, where parts can be aligned to a grid and stretch in different ways when the page is resized.
- It consists of a parent element/flex container that holds the items that need positioning, and child elements/flex items that are positioned within.

Flex container

- Use `display: flex;`
- You can define whether it's organised by row/column, item alignment, wrapping, stretching and more.

Flex items

- You can define ordering, relative size, ability to stretch and more.

See https://css-tricks.com/snippets/css/a-guide-to-flexbox/ for a complete guide.

SUB CRUCE LUMEN

THE UNIVERSITY
of ADELAIDE

CRICOS PROVIDER NUMBER 00123M