POWERING
POTENTIAL

The Box Model

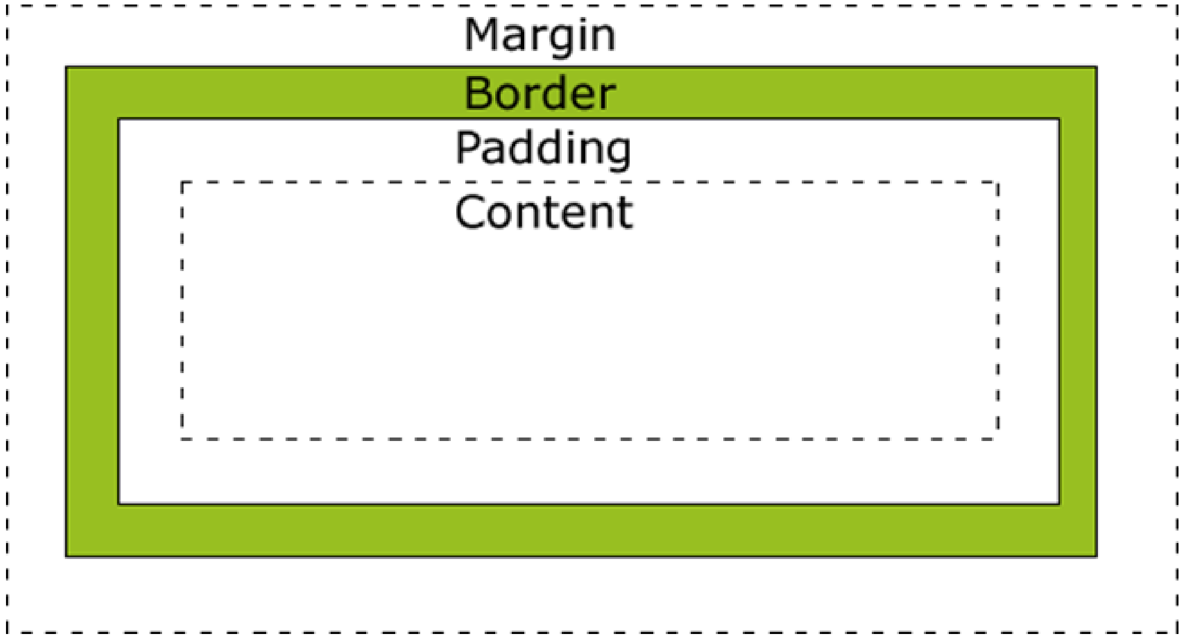
CSS Fundamentals

The Box Model

- All HTML elements can be considered as boxes
- Box Model is used when talking about design and layout
- Essentially a box that wraps around HTML elements
- Consists of margins, borders, padding, and the actual content

Box model allows:

- Placing a border around elements
- Space elements in relation to other elements



The Box Model

Margin:

- Clears an area around the border
- Is transparent (no background colour)

Border:

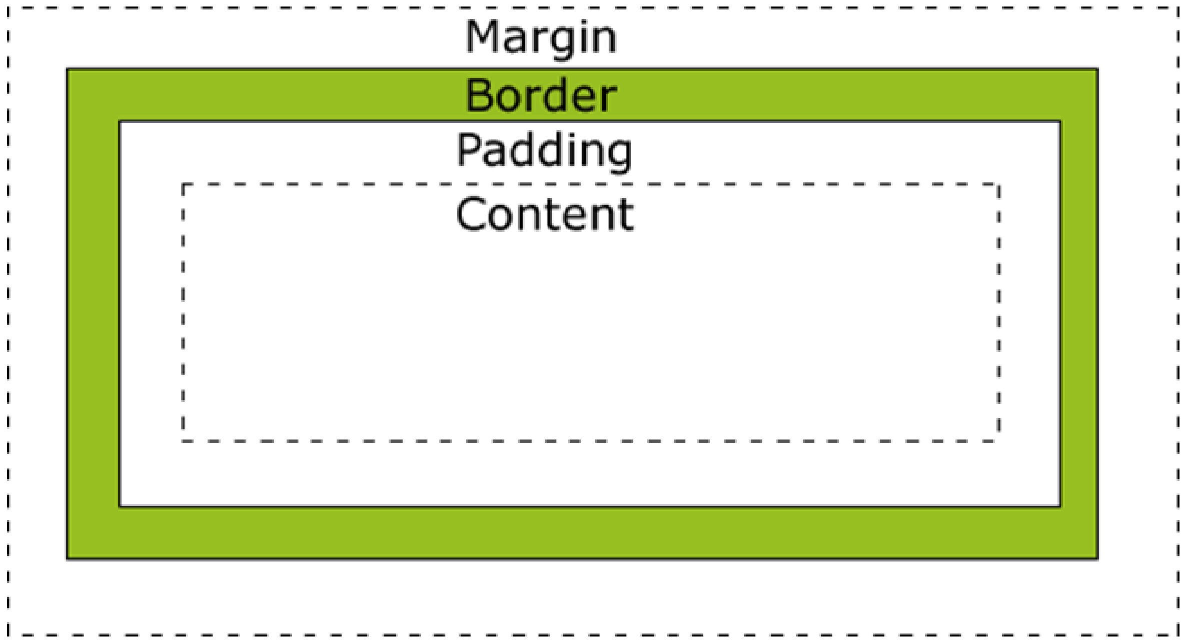
- Goes around the padding and the content
- Affected by the background colour of the box

Padding:

- Clears an area around the content
- Affected by the background colour of the box

Content:

- The content of the box, where text and images appear



The Box Model – Settings Properties

All HTML elements have four sides – top, bottom, left, and right

- Properties can be set for each dimension or in a compound rule:

```
margin: 10px;
margin: 10px 5px;
margin: 0px 5px 10px 15px;
margin-bottom: 2em;
```

10px in all directions → `margin: 10px;`

10px top and bottom 5px left and right → `margin: 10px 5px;`

top, left, bottom right → `margin: 0px 5px 10px 15px;`

Individual direction → `margin-bottom: 2em;`

Child elements typically have their own block properties

- Can be set independent of the parent
- The inner width of an element (content) available is a remainder of reserved space by parent elements
- Background colours and images can also be set

Element Width and Height

- When you set the width and height properties of an element with CSS you only set them for the CONTENT area
- To calculate the full size of an element you must add the padding, borders and margin to the width of the content
- What is the TOTAL width of the space the element takes here?

```
width: 250px;
padding: 10px;
border: 5px solid gray
margin: 10px
```

It is 300px

Let's do the maths

+ 250px (content width)

+ 20px (left and right padding)

+ 10px (left and right border)

+ 20px (left and right margin)

300px

The border-box model

The broken box model is a familiar tale of woe to most

- CSS3 includes an attribute called box-sizing
- Set to content-box to get the traditional W3C box model

```
article { box-sizing: content-box; }
```

- The total width of the element will be:
  - the width set on the element
  - plus the width of the borders and padding
- If border-box borders and paddings include in the width

```
article { box-sizing: border-box; }
```

Borders

Borders can have the following attributes set:

- `border-width: all [top, right, bottom left]`
- `border-style: all [top and bottom, left and right]`
- `border-color: top [left, bottom, right]`
    - Properties can be set individually for all by using shorthand `border` property

```
div { border: 2px dashed blue; }
```

Can specify `border` for each side by inserting `top`, `left`, `bottom` or `right` between `border` and *property to set* or use the shorthand:

```
div { border-top-style: double; }
div { border-left: 5px inset purple; }
```

Rounded borders

- Pre-CSS3 had to be achieved through JavaScript or images:

```
border-radius: 30px
```

- Different radius can be added to different corners

```
border-top-left-radius: 50px;
border-top-right-radius: 30px;
border-bottom-right-radius: 50px;
border-bottom-left-radius: 30px;
```

- Shorthand

```
border-radius: 50px 30px 50px 30px;
```

Outline

Renders a uniform line for viewers attention

- Rendered on top of an elements rendering box
- Does not influence a box's position or size

```
outline: 3px dashed #3a5c7a;
```

Optional outline-offset property

- Offsets an outline
- Then draws it beyond the border edge.

```
outline-offset: 10px
```

POWERING POTENTIAL

Positioning Elements

CSS Fundamentals

Positioning Elements

Position: relative | static

- The content edge of the nearest block-level ancestor

Position: absolute

The nearest positioned ancestor according to:

- The padding edge of the if the ancestor is block-level
- The content edge of the first/last box if the ancestor is inline

Position: fixed

- The window / printed page

Relative positioning

Relative positioning: offset from default position

- i.e. moved from where it would have been
- Offset not measured from containing block

Next element flows as if the box hasn't been moved

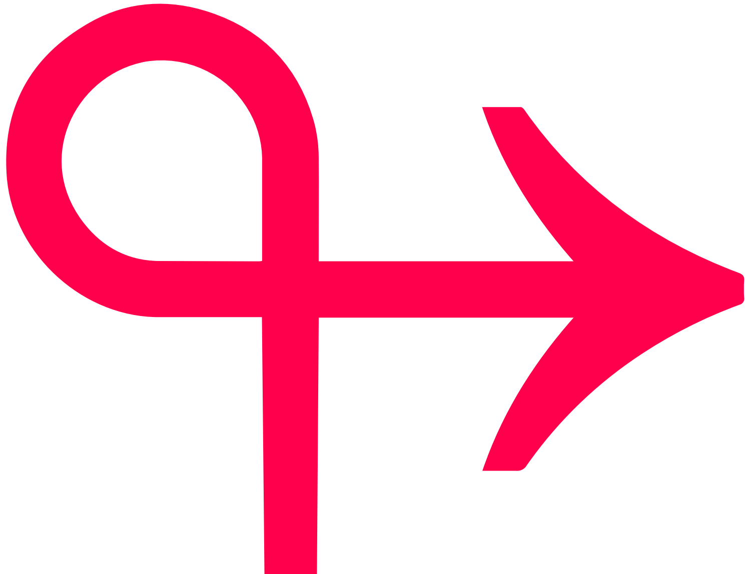
- Relative boxes take up space where they would have been

Moved element has same size as if it hadn't been moved

- Hence specify only one of left/right and top/bottom
  - e.g., if you specify left and right this could change the width of the element, which is not allowed, hence one of left/right will be ignored

See

- http://www.w3.org/TR/CSS21/visuren.html#relative-positioning



ABSOLUTE POSITIONING

Absolute positioning: offset from container's position

- i.e., relative to container not page

Offset measured from

- Block-level ancestor: the top, left of ancestor's padding box
  - i.e., outside of padding, inside of border
- Inline ancestor: the top, left of the ancestor's content box
  - i.e., outside of content

See

- http://www.w3.org/TR/CSS21/visuren.html#position-props
- http://www.w3.org/TR/CSS21/visuren.html#absolutely-positioned

Margin - Positive and Negative Values

Giving CSS positive values for padding or margin puts space between element and its reference

```
margin-left: 20px;
```

- Puts 20 pixels between the left margin of the element and its reference - effectively moves the element 20 pixels to the right

Giving CSS negative values for padding or margin moves the element towards its reference

```
margin-left: -20px;
```

- Effectively moves the element 20 pixels to the left

Float and Clear

Float will move an element and flow text around it

- Treats the element as a block element and moves it left/right
- Rest of the page flows around the floated element
  - The available box is shrunk by the amount the floats take up

Clear will move an element to after the float

- Adds clearance to the top margin to move it clear of the float
  - Moves top border edge below the bottom outer edge of the float
  - Unless the cleared element is also a float (line up outer edges)

See

- http://www.w3.org/TR/CSS21/visuren.html#propdef-float
- http://www.w3.org/TR/CSS21/visuren.html#propdef-clear

OVERFLOW, MIN, & MAX DIMENSIONS

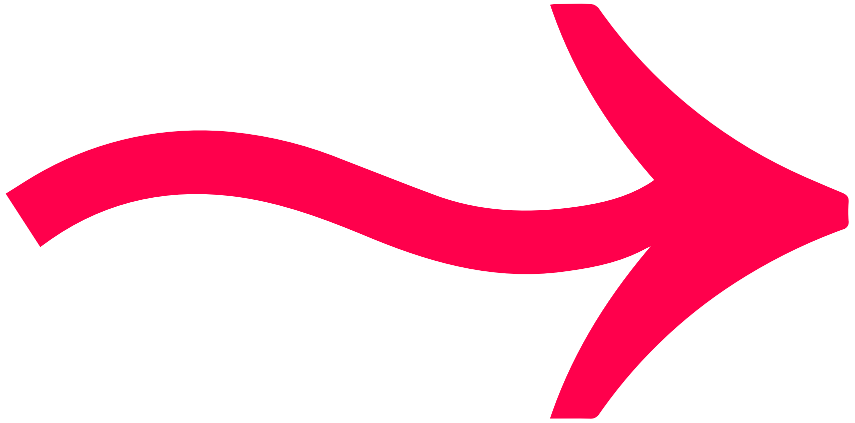
The width and height of an object can be constrained

- With **min-height/min-width** and **max-height/max-width**
- Once set, an element will never grow/shrink beyond these values

The element is now smaller than the content it displays

- What happens to this content can be controlled with the **overflow**
- Can be set to:
  - auto
  - visible
  - hidden
- CSS3 allows overflow control on a specific axis **overflow-x/y**
- In CSS3, we also have the **hidden** property

Controlling how an element displays



Elements are primarily set to be block or inline as their display type

- This behaviour can be changed in CSS
- By modifying the display attribute
- By setting an element property **`display:none`** it is hidden
    - The element is then removed from the flow
    - Can be accomplished with a **`hidden`** attribute in HTML5
    - Alternatively there is the **`visible`** property
        - Does not remove the element from the document flow

Elements can also be switched between inline and block display

- Useful for advanced layout

Quick Lab Chapter 9 – Positioning Elements

Use positioning and styling techniques to lay out a page to a given design