

Boxes

Gerard Luskin

Boxes

Everything is a box

- As mentioned and seen previously, every element in the body of the page is actually a rectangular *box*
 - ▶ *block* element boxes always expand their width to fill all available space
 - ▶ *inline* element boxes do not. They merely surround the *content* of the element.
- By default, *block* element boxes *stack* vertically

Pink = inline, all others are block

Balloons Ltd.

What We Do

We make bespoke balloons of all shapes and sizes, for all occasions and purposes.

From *helium-filled party balloons* to *hot-air balloons*, we have it covered. So, whether you are celebrating a birthday or flying across the channel, give us a call!

Who We Are

We are a small team of dedicated professionals that are mad about balloons.

Our engineers are all **BBEI** accredited and are highly experienced in all things related to the balloon industry.

Where To Find Us

Our workshop and showroom are located in Northampton Square, London. **Visits are by appointment only**, so please give us a call or send us an email before you come.

CSS and Boxes

- In CSS, *block* element boxes are described by the *box-model*
 - ▶ *Inline* elements are too, but to a lesser extent
- The *box model* is concerned with the space occupied by an element
- Understanding the *box model* is the key to unlocking many of the secrets of CSS!

The Box model

- Every *box* consists of:
 - ▶ A *content* area
 - ▶ A *padding* area
 - ▶ A *border* area
 - ▶ A *margin* area

The Box model



The *content* area

- The **content** area is the space which the element's content will occupy
- By default:
 - ▶ A *block* element's content area expands in width to fill all available space
 - ▶ An *inline* element's content area expands in width just enough to enclose the content
- The *height* of the content area is determined by the amount of content within the element

The *content* area and CSS

- With CSS, we can control both the *width* and *height* of the content area.
- However... it is complicated!
 - ▶ We will revisit this aspect of the *box* later in the course.

The *border area*

- The *border area* represents the visible boundary of the element.
- By default, most elements have a zero-width border
 - ▶ However, the border area is still present!
- As we will see, we can control the border colour, size and style with CSS

The *padding* area

- The *padding area* is the space between the *content area* and the *border*
- With CSS, we can control the amount of padding applied to each side of the *box*
- By default, most elements have no padding
 - ▶ i.e. the *content area* touches the *border area* on all sides

The *padding* area defaults

- Some of the elements we have already used *do* have default padding
 - ▶ Can you guess which ones?

The *margin* area

- The *margin area* is the *empty* space surrounding an element's border
 - ▶ i.e. it determines the space between the element and it's neighbours
- With CSS, we can control the amount of margin applied to each side of the *box*

The *margin* area defaults

- By default, a lot of *block* elements have top and bottom margins applied to them
 - ▶ E.g.: *h1-h6, p* (notice the space above and below each of these elements in your pages)
- However, by default, they have no left or right margin

CSS box properties

- All of these *box properties* can be manipulated and controlled with CSS
- We can set these properties on each *side* of the box individually
 - ▶ Or, we can set them for all 4 sides at once
- We can also decorate the *box* with things like backgrounds, drop-shadows, etc.

Borders

Border properties

- We can customise the element's border with the *border* family of properties
 - ▶ *border-style* (what kind of line should be used)
 - ▶ *border-width* (what size should the border be)
 - ▶ *border-color* (what colour should the border be)

Property: border-style

- The *border-style* properties allows us to specify the type of line that will be used for the border
- The values we can use are:
 - ▶ **solid, dotted, dashed, double, groove, ridge, inset, outset, none, hidden**
- The default value for most elements is **none**

Property: border-style

- To set the border style for each side of the box, we can use the properties:
 - ▶ **border-top-style**
 - ▶ **border-bottom-style**
 - ▶ **border-left-style**
 - ▶ **border-right-style**

Property: border-style - example 1

- To set different styles on each side:

```
h1 {  
    border-top-style: solid;  
    border-right-style: double;  
    border-bottom-style: dashed;  
    border-left-style: dotted;  
}
```

Property: border-style - example 2

- To set the style for bottom and left sides (others will have no border)

```
h1 {  
    border-bottom-style: solid;  
    border-left-style: solid;  
}
```

- **Q:** Why will the top and right edges have no border?

Shorthand for border style

- CSS provides a *shorthand* property that can make life a bit easier for us
 - ▶ The **border-style** property
- As the *value*, we can specify the desired styles for each side, separated by a space
- The order of the the values **must** be:
 - ▶ **T**op, **R**ight, **B**ottom, **L**eft
- Or, we can specify a single value to apply to all sides

Border style shorthand example

- Set different styles on each side:

```
h1 {  
    border-style: solid double dashed dotted;  
}
```

- To set the same style on all 4 sides:

```
h1 {  
    border-style: solid;  
}
```

Border style : Try it

- Open *index.html* from your *test* website in your code editor. If you do not have a *h1* element in the *body*, add one.
- Open the style sheet for your *test* website in your code editor and add this rule:

```
h1 {  
    border-style: solid;  
}
```

- Save your files and preview the page. Can you see a border around the heading?

Property: border-width

- The *border-width* property allows us to set the thickness of the border
- The *value* can be any valid CSS unit of measurement (px, em, etc.)
 - ▶ Commonly, we use *pixels* for border dimensions
 - ▶ The default width of borders in FireFox is 3 pixels (other browsers may differ)

Property: border-width

- To set the border width for each side of the box, we use the properties:
 - ▶ **border-top-width**
 - ▶ **border-bottom-width**
 - ▶ **border-left-width**
 - ▶ **border-right-width**

Border width - useless on it's own

- When setting the *width* of the border, we still have to set the *border-style*
 - ▶ Without a *border-style* declaration, the *border-width* declarations would have no effect.
- The same is true for all other border properties
- Why?... because the default value for *border-style* is *none*
 - ▶ *none* = no border

Property: border-width - example 1

- To set different widths on each side:

```
h1 {  
    border-style: solid;  
    border-top-width: 5px;  
    border-right-width: 20px;  
    border-bottom-width: 30px;  
    border-left-width: 40px;  
}
```

Property: border-width - example 2a

- To set the width for selected sides (top & bottom)

```
h1 {  
    border-style: solid;  
    border-top-width: 5px;  
    border-bottom-width: 30px;  
}
```

- **Q:** What will happen with the left and right borders?

Property: border-width - example 2b

- In this case, if we want to remove the left and right borders, we have to set their width to **0**

```
h1 {  
    border-style: solid;  
    border-top-width: 5px;  
    border-right-width: 0;  
    border-bottom-width: 30px;  
    border-left-width: 0;  
}
```

- **Q:** Could we achieve the same result with different rules?

Shorthand for border width

- Just like *border-style*, there is also a shorthand we can use to set the border width:
 - ▶ **border-width**
- Again, we can specify 4 values in the order: **T**op, **R**ight, **B**ottom, **L**eft
- Or we can specify a single value to apply to all 4 sides

Shorthand: border-width example

- Set different widths on each side:

```
h1 {  
    border-style: solid;  
    border-width: 5px 20px 30px 40px;  
}
```

- Set the width on all 4 sides to 5 pixels:

```
h1 {  
    border-style: solid;  
    border-width: 5px;  
}
```


Border width: Try it

- Open the style sheet for your *test* website in your code editor
- Add a *border-width* declaration to your *h1* rule:

```
h1 {  
    border-style: solid;  
    border-width: 10px;  
}
```

- Save your files and preview the page. Is the border thicker than before?

Property: border-color

- By default, the colour of the borders will be the same as the colour of the element's text
- We can change this with the *border-color* property
- The *values* we can use are the same as for *color*
 - ▶ Colour keywords, Hexadecimal, RGB, Etc.

Property: border-color

- To set the border colour for each side of the box, we use:
 - ▶ **border-top-color**
 - ▶ **border-bottom-color**
 - ▶ **border-left-color**
 - ▶ **border-right-color**

Property: border-color - example 1

- To set different border colours for each side:

```
h1 {  
    border-style: solid;  
    border-width: 5px;  
    border-top-color: #ff0000;  
    border-right-color: #0000ff;  
    border-bottom-color: #ff6600;  
    border-left-color: #00ff00;  
}
```

Property: border-color - example 2

- To set the colour of the top and bottom borders:

```
h1 {  
    border-style: solid;  
    border-width: 5px;  
    border-top-color: #ff0000;  
    border-bottom-color: #ff6600;  
}
```

- **Q:** What colour will the left and right borders be?

Shorthand for border colours

- There is also a shorthand property we can use to set the border colour:
 - ▶ **border-color**
- Again, we can specify 4 values in the order: **T**op, **R**ight, **B**ottom, **L**eft
- Or we can specify a single value to apply to all sides

Shorthand border-color example 1

- To set the colour for all 4 sides to orange:

```
h1 {  
    border-style: solid;  
    border-width: 5px;  
    border-color: #ff6600;  
}
```

Shorthand border-color example 2

- To set different colours for each side:

```
h1 {  
  border-style: solid;  
  border-width: 5px;  
  border-color: #ff0000 #0000ff #ff6600 #00ff00;  
}
```


Border color : Try it

- Open the style sheet for your *test* website in your code editor
- Add a *border-color* declaration to the *h1* rule:

```
h1 {  
    border-style: solid;  
    border-width: 10px;  
    border-color: #ff6600;  
}
```

- Save your files and preview the page.

Shorter Shorthand for borders

- If you are setting the same *style*, *width* and *color* on all four sides, there is another shorthand property you can use:
 - ▶ **border**
- For the value, we use the values for:
 - ▶ [border-width] [border-style] [border-color]
- The *values* are separated by spaces

Shorthand: border example

- This code:

```
h1 {  
    border-style: solid;  
    border-width: 5px;  
    border-color: #ff6600;  
}
```

- Can be rewritten as this:

```
h1 {  
    border: 5px solid #ff6600;  
}
```

Shorter shorthand - specific sides

- There is also a shorthand that allows us to set all of the border properties for a specific side in one line:
 - ▶ **border-top**
 - ▶ **border-right**
 - ▶ **border-bottom**
 - ▶ **border-left**

Shorter shorthand - example

- To set all of the properties for the *top* border at once:

```
h1 {  
    border-top: 5px solid #ff6600;  
}
```

- Will there be any border on the other 3 sides?

Border shorthand: Try it

- Open the style sheet for your *test* website in your code editor
- Change your *h1* rule to look like this:

```
h1 {  
    border: 10px solid #ff6600;  
}
```

- Save your files and preview the page (there should be no visible change).

Border shorthand: Try it again

- In your *test* website's *index.html*, make sure you have at least one paragraph in the *body*
- In your style sheet, add the following rule:

```
p {  
    border-bottom: 3px dotted #0099ff;  
}
```

- Save your files and preview the page. Does the paragraph have a bottom border?

Backgrounds

Background properties

- We can also use CSS to apply a *background* to an element's box
 - ▶ We can specify a solid colour
 - ▶ We can specify background images (More on this later in the course)
- By default, most elements have a *transparent* background

Background coverage

- When we set the background for an element, it will cover:
 - ▶ The *content* area
 - ▶ The *padding* area
 - ▶ The *border* area (but underneath the border)
- The *margin* area will **not** have a background

Property: background-color

- To set a solid colour as the background, we use the property:
 - ▶ **background-color**
- For the *value*, we can use the same values we used for *color* and *border-color*
 - ▶ Colour keywords, Hexadecimal, RGB, Etc.

Property: background-color - example

- To give an element an orange background:

```
h1 {  
    background-color: #ff6600;  
}
```

Background color: Try it

- Open the style sheet for your *test* website in your code editor
- Add a *background-color* declaration to your *h1* rule:

```
h1 {  
    border: 10px solid #ff6600;  
    background-color: #ffb380;  
}
```

- Save your files and preview the page. Can you see the background colour?

Margin and Padding

About Margin and Padding

- Both of these properties are very similar
 - ▶ They both define a region of *space*
 - ▶ They both accept the same kind of *values*
- The *values* we can use with them are any valid CSS unit of measurement:
 - ▶ *em, pixels, %, Etc.*

Margin and Padding - the difference

- The difference between the two properties is:
 - ▶ **margin** creates space *outside* the *visible* box
 - ▶ **padding** creates space *inside* the *visible* box
- Note, if an element has no border or background, these properties will appear to do the same thing!

Margin and Padding values

- When setting the values for these properties, we can set them individually:
 - ▶ **padding-top, padding-right, padding-bottom, padding-left**
 - ▶ **margin-top, margin-right, margin-bottom, margin-left**
- Or, we can use the shorthand for each:
 - ▶ **padding**
 - ▶ **margin**

Margin and Padding units of measure

- It is common to use *em* units for margin and padding values
 - ▶ Where: $1em = 1 \times \text{Element's font-size}$
- Why?... if we increase the *font-size* of an element, or the user increases the *default font-size* in their browser:
 - ▶ the margin and padding will increase proportionally (a good thing!).

Setting margin/padding individually

- Add *1em* of space *inside the box* on the left side only:

```
h1 {  
    padding-left: 1em;  
}
```

- Add *2em* of space *outside the box* on the bottom side only:

```
h1 {  
    margin-bottom: 2em;  
}
```

Margin and Padding shorthand

- When using the shorthand syntax:
 - ▶ We can specify 4 values in the order: **T**op, **R**ight, **B**ottom, **L**eft
 - ▶ Or we can specify a single value to apply to all 4 sides
- Just like we did with the border shorthands

Margin and Padding shorthand example 1

- Add *1em* of space *inside the box* on the left side only:

```
h1 {  
    padding: 0 0 0 1em;  
}
```

- Add *1em* of space *outside the box* on the bottom side only:

```
h1 {  
    margin: 0 0 1em 0;  
}
```

Margin and Padding shorthand example 2

- Add *1em* of space *inside the box* on all sides:

```
h1 {  
    padding: 1em;  
}
```

- Add *2em* of space *outside the box* on all sides:

```
h1 {  
    margin: 2em;  
}
```

Padding: Try it

- Open the style sheet for your *test* website in your code editor
- Add a *padding* declaration to your *h1* rule:

```
h1 {  
    border: 10px solid #ff6600;  
    background-color: #ffb380;  
    padding: 1em;  
}
```

- Save your files and preview the page.

Margin: Try it

- Open *index.html* from your hello website in your code editor and make sure there is a paragraph below the *h1* element. If there is not, add one with a few words in it.
- Preview your page and note the amount of space between the heading box and the paragraph
- Continued on next slide...

Margin: Try it

- In your style sheet, add a *margin-bottom* declaration to the *h1* rule:

```
h1 {  
    border: 10px solid #ff6600;  
    background-color: #ffb380;  
    padding: 1em;  
    margin-bottom: 5em;  
}
```

- Save your files and preview your page. Notice what has changed?

Exercise

- Now do the *Boxes exercise*

Good to know...

Inheritance and the box model

- Unlike the text properties we used previously, box-model properties are **not** inherited by an element's children.
- They only apply to the element on which they are used.

```
body {  
    font-family: Arial; /* Inherited by children */  
    border: 1px solid #000; /* Not inherited */  
}
```

Regarding shorthands

- For all of the shorthand properties that accept either *one* or *four* values, you may also see them specified with *two* or *three* values:

```
h1 {  
  padding: 1em 2em;  
  margin: 1em 2em 3em;  
}
```

- Q:** Can you guess what these do?

Shorthand values

- **Two values:** the first value is applied to the *top* and the *bottom*, the second is applied to the *left* and *right*
- **Three values:** the first value applies to the *top*, the second applies to the *left* and *right*, the third applies to the *bottom*

Regarding Inline boxes

- All the examples utilise *block* elements (h1, p, etc.)
- When used with *inline* elements, they may appear to be broken
 - ▶ In particular, top and bottom margins
- Due to the way *inline* elements are used, this is not a problem