

The Box Model

Website Development 1

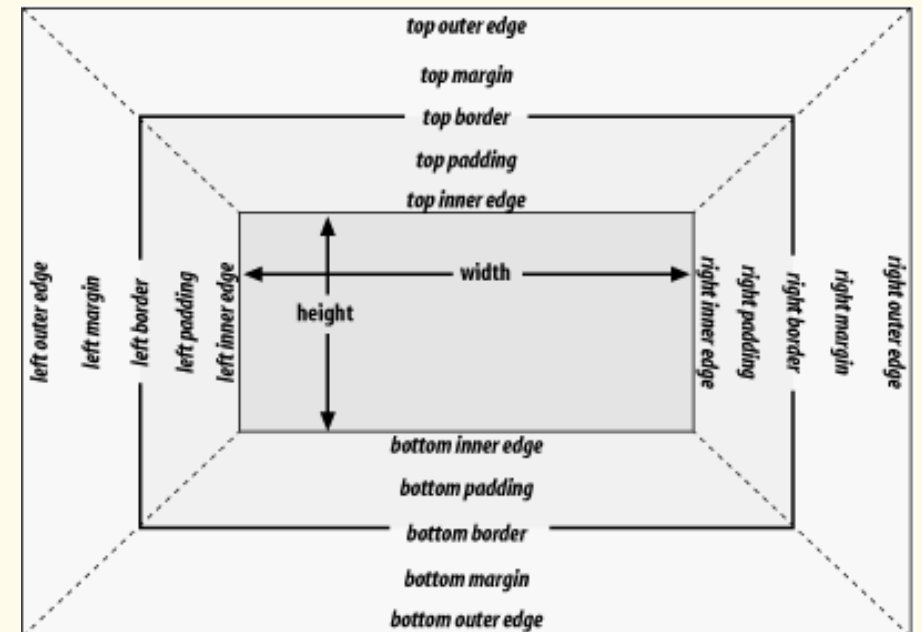
Agenda

- CSS Box Model
- Block vs. inline display
- Applying box model properties
- Box-Sizing

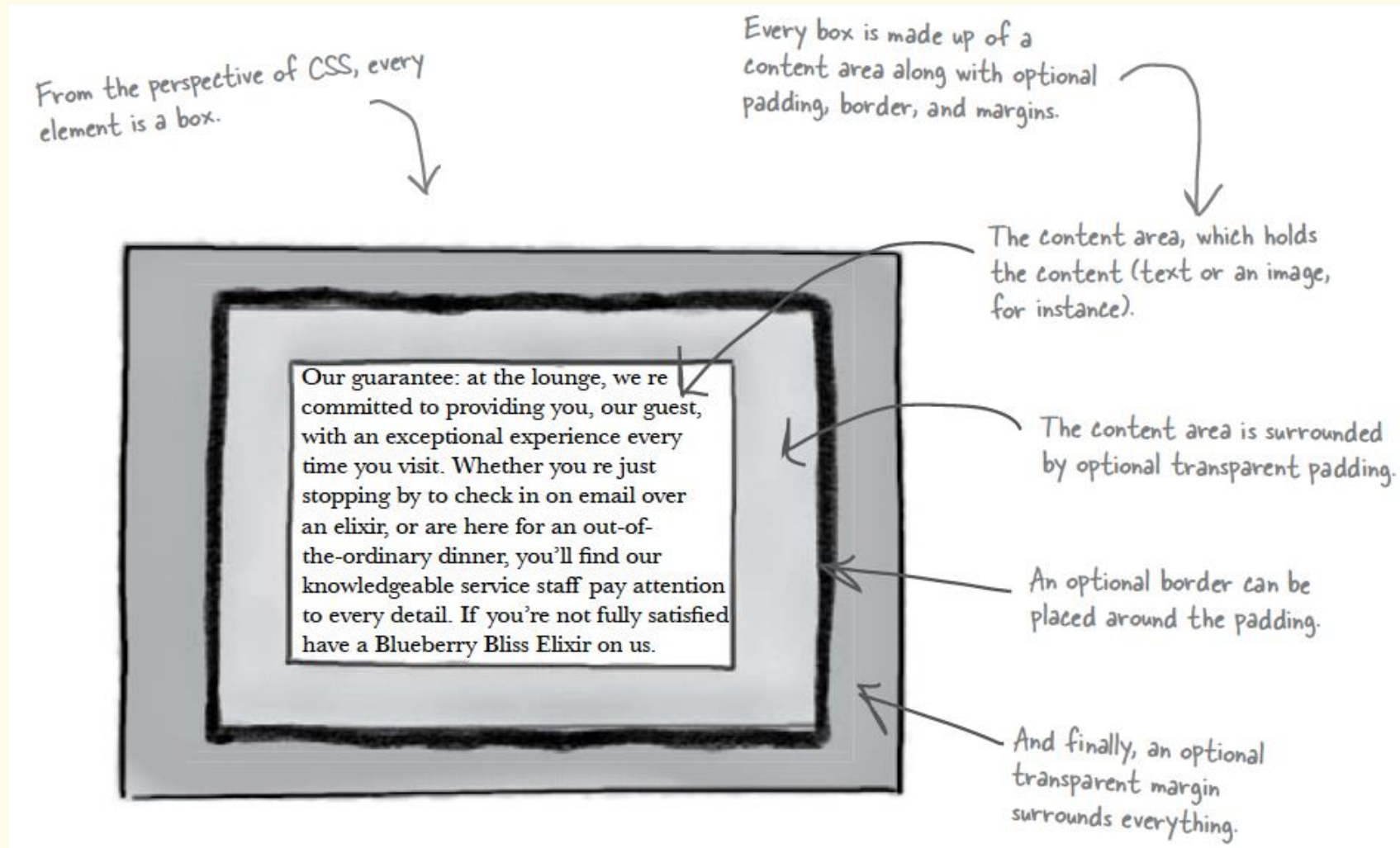
CSS Box Model

- So far, your CSS has focused on changing simple properties of elements, like size, colour, and decorations. For effective layout control, you have got to move on to the box model.
- The box model is how CSS sees elements. CSS treats every single element as if it were represented by a box.

- Boxes consist of:
 - Content
 - Padding
 - Border
 - Margin



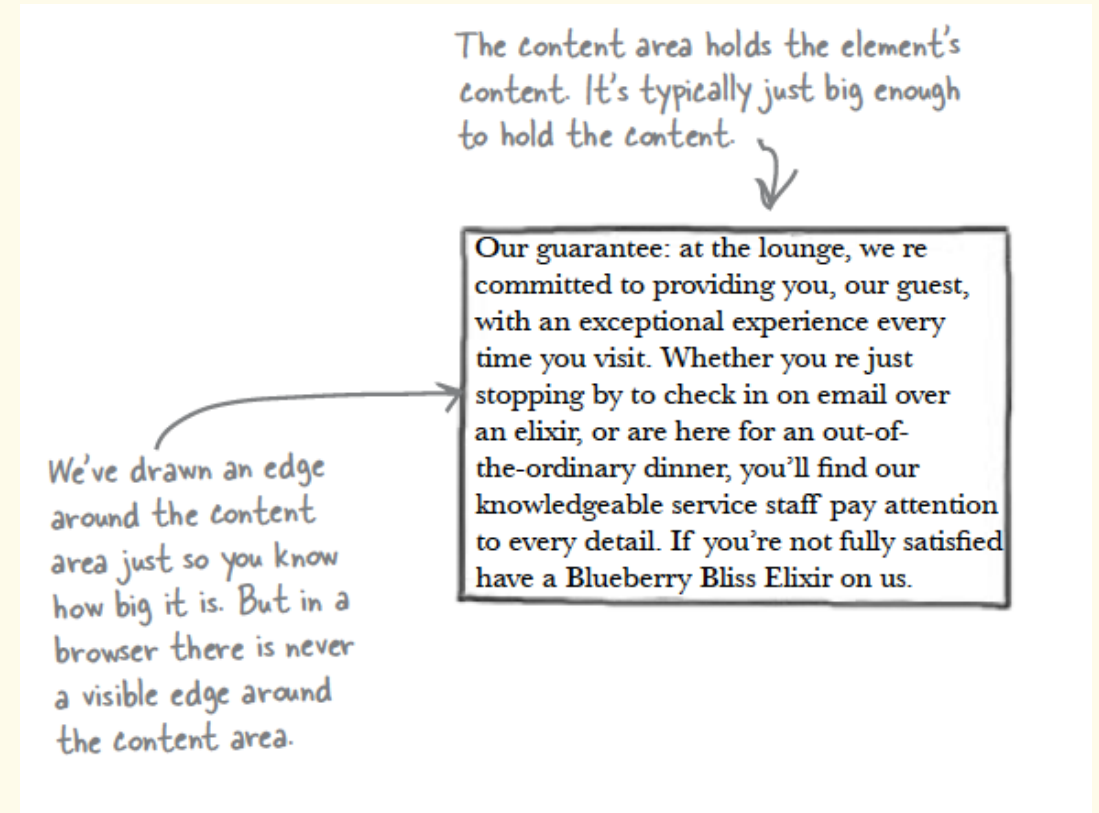
CSS Box Model



CSS Box Model

Content Area

- Every element starts with some content, like text or an image, and this content is placed inside a box that is just big enough to contain it.
- The content area has no whitespace between the content and the edge of the box.

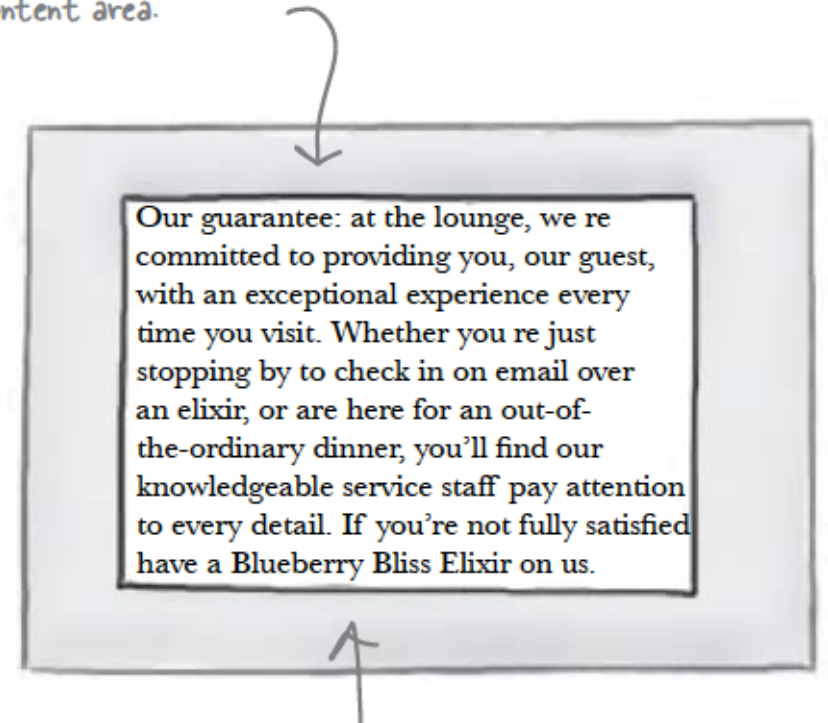


CSS Box Model

Padding

- Any box can have a layer of padding around the content area.
- Padding is optional, so you don't have to have it, but you can use padding to create visual whitespace between the content and the border of the box.
- The padding is transparent and has no colour or decoration of its own.

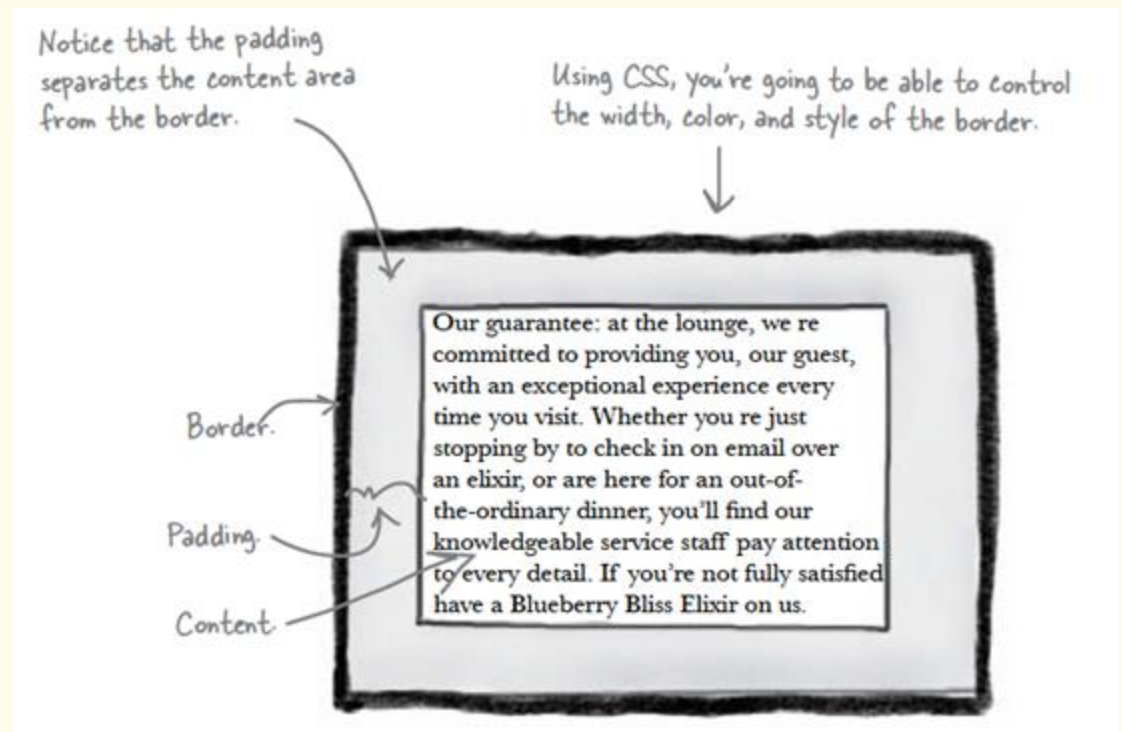
The browser adds optional padding around the content area.



CSS Box Model

Border

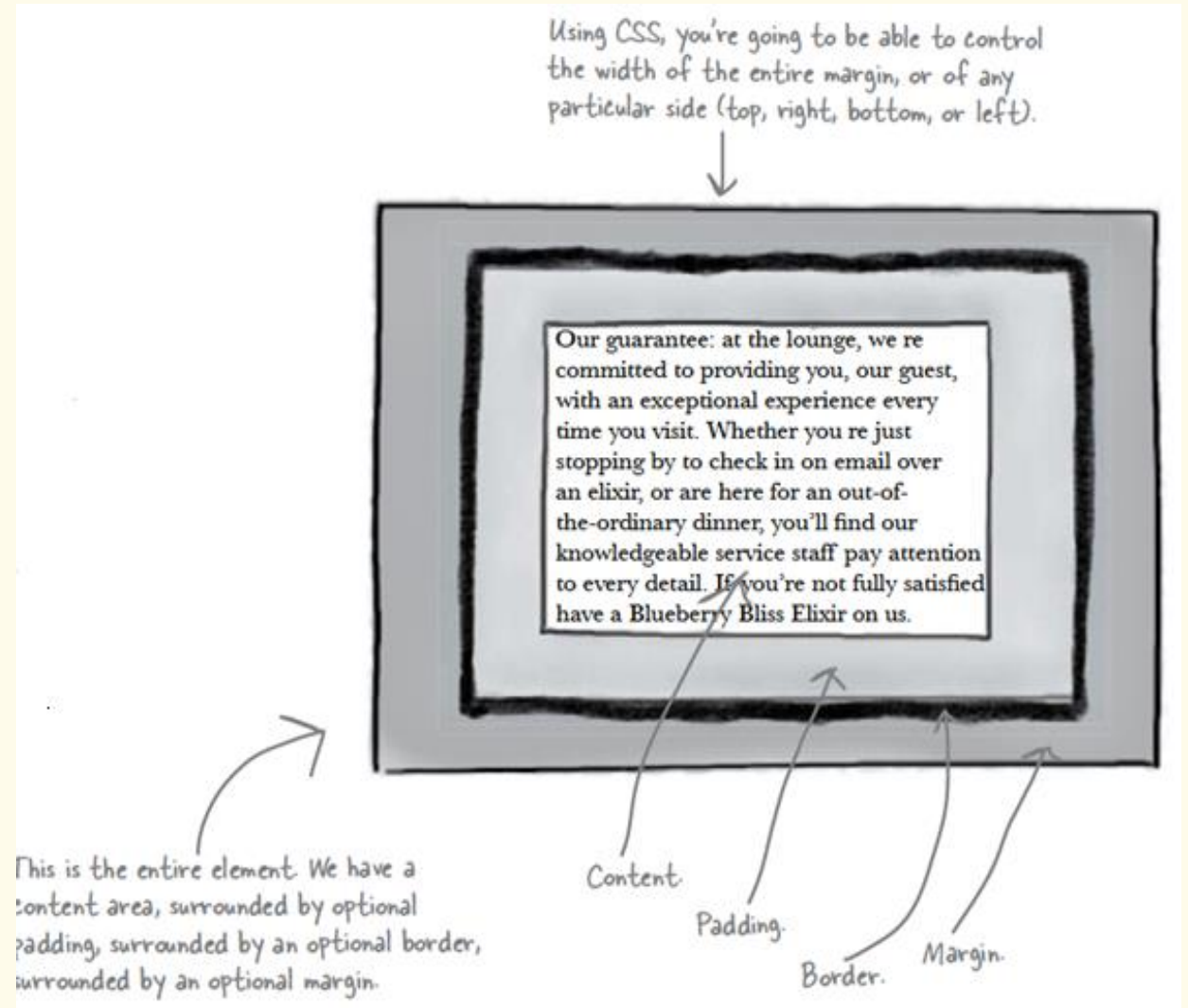
- Elements can have an optional border around them.
- The border surrounds the padding and because it takes the form of a line around the content, borders provide visual separation between content and other elements on the same page.
- Borders can be various widths, colours and styles.



CSS Box Model

Margin

- The margin is also optional and surrounds the border.
- The margin gives you a way to add space between your element and other elements on the same page.
- If two boxes are next to each other, the margins act as the space in between them.
- Like padding, margins are transparent and have no colour or decoration of their own.



Agenda

- CSS Box Model
- Block vs. inline display
- Applying box model properties
- Box-Sizing

Note: Block vs. Inline elements

- Block elements:

- Always start on a new line
- Take up 100% of the width available

- Examples:

- p, h1, div, blockquote, table

- Inline elements:

- Do not start on a new line
- Only take up as much width as needed (based on the contents)

- Examples:

- span, a, img, b, i

Block vs. Inline

BubbleUnder.com

Diving club for the south-west UK - **let's make a splash!**

- [Home](#)
- [About Us](#)
- [Contact Us](#)

Welcome to our **super-doooper Scuba site**

Glad you could drop in and share some air with us! You've passed your **underwater navigation skills** and successfully found your way to the start point - or in this case, our home page.

Display property

- We can change the default type using the **display** property
- There are several options for displaying elements:
 - Block
 - Inline
 - Block-inline
 - None (doesn't show the element)

Block, inline, inline-block

- Block elements stack one on top of the other (and we can set width and height properties)
- Inline elements stay in line; they don't take up any extra space (width and height can't be set)
- **Inline-block** elements stay in line AND we can set a width and height



Agenda

- CSS Box Model
- Block vs. inline display
- Applying box model properties
- Box-Sizing

Applying Borders

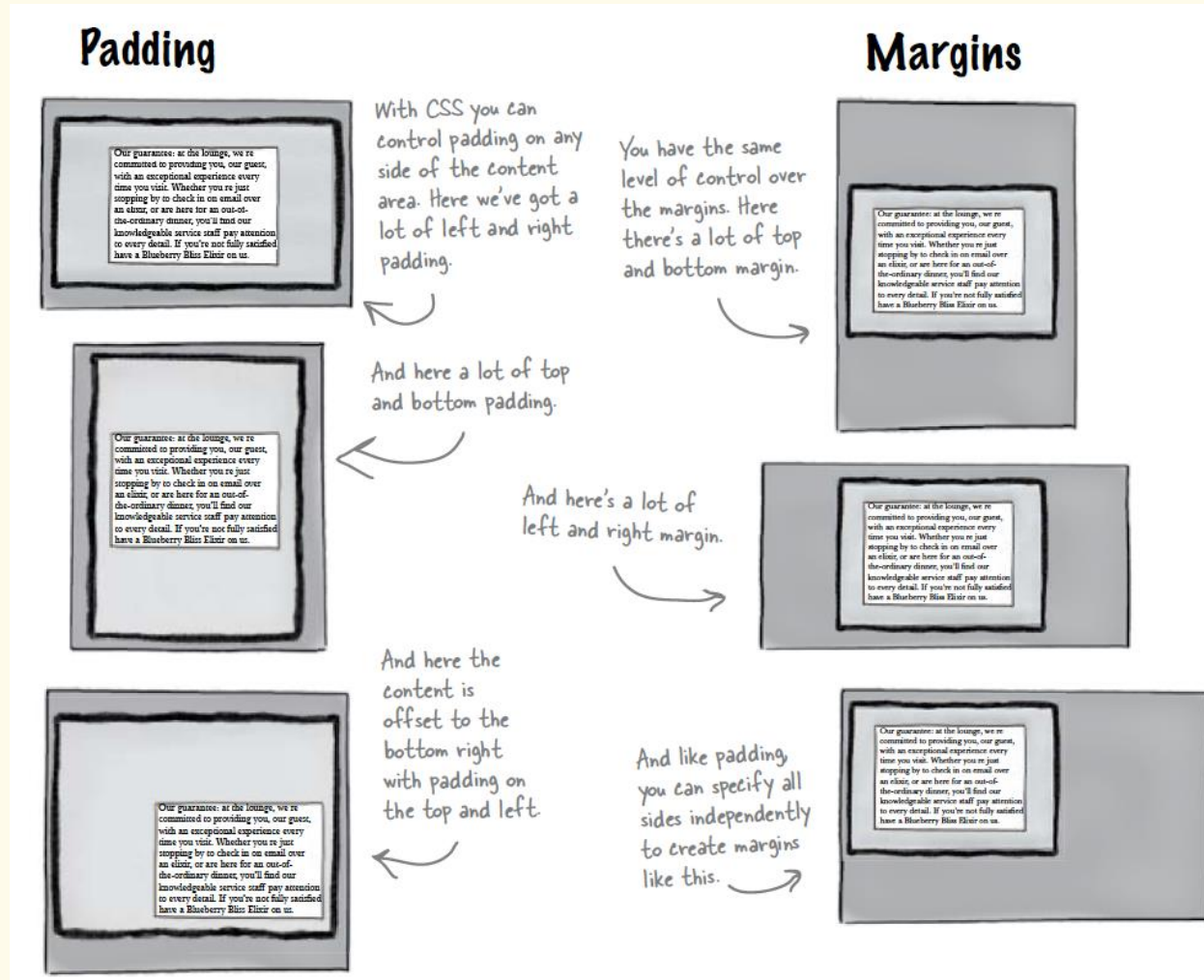
- Borders can have different properties:
 - Border width
 - Border style
 - Border colour
- Or you can specify all three properties using the shorthand “border” property

```
border-width: 2px;  
border-color: navy;  
border-style: solid;
```

```
border: 2px solid navy;
```

Applying Padding & Margins

- You can vary the padding and margin that is used on the top, bottom, left and right of each box.



Applying Padding & Margins

- Specify all four separately:

```
padding-top: 5px;  
padding-right: 12px;  
padding-bottom: 5px;  
padding-left: 10px;
```

- Use shorthand “padding” property to do all four at once:

```
padding: 5px 12px 5px 10px;
```

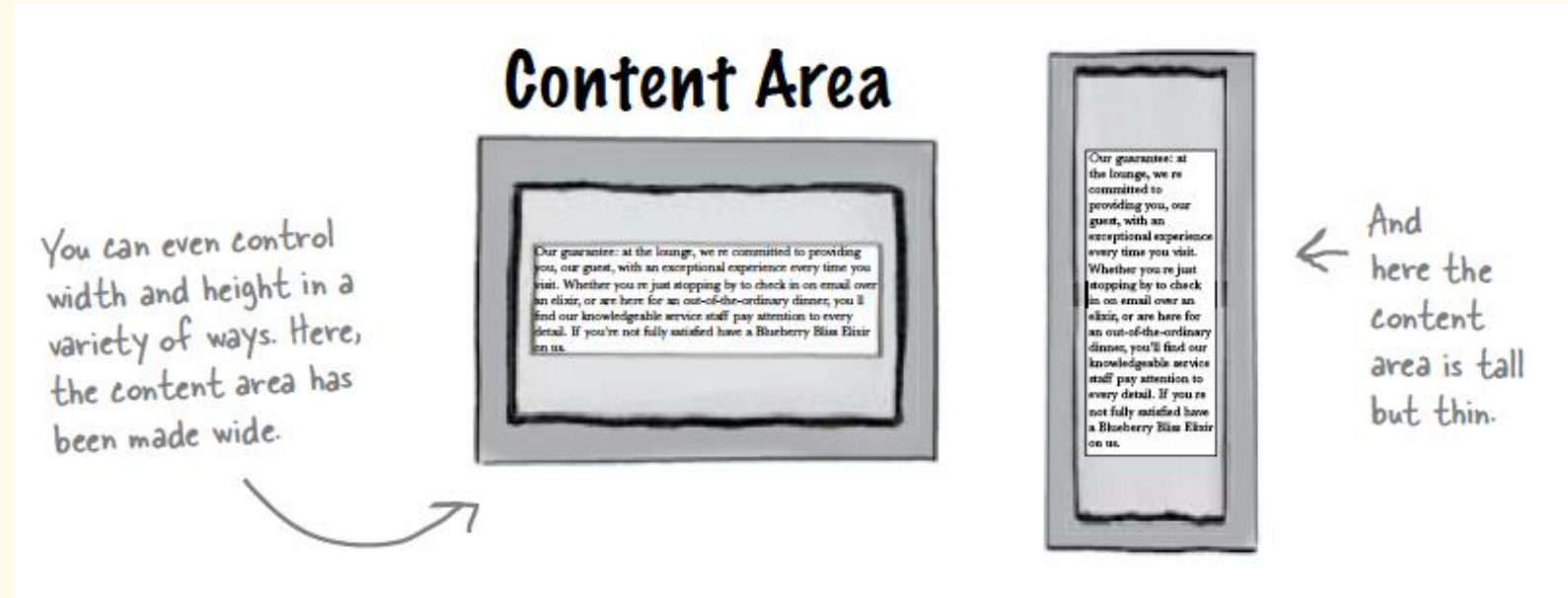
- Specify two values: one for top/bottom, second for left/right:

```
padding: 5px 10px;
```

- The same can be done for margins.

Variations: Content

- You can also set the width and height of content areas.



```
div{  
  width: 50%;  
  height: 800px;  
}
```

Agenda

- CSS Box Model
- Block vs. inline display
- Applying box model properties
- Box-Sizing

Box-Sizing

- By default, the width and height of an element is calculated like this:
 - $\text{width} + \text{padding} + \text{border} = \text{actual width of an element}$
 - $\text{height} + \text{padding} + \text{border} = \text{actual height of an element}$
- This means: When you set the width/height of an element, the element often appears bigger than you have set (because the element's border and padding are added to the element's specified width/height).

Box-Sizing

```
.div1 {  
  width: 300px;  
  height: 100px;  
  border: 1px solid blue;  
}  
  
.div2 {  
  width: 300px;  
  height: 100px;  
  padding: 50px;  
  border: 1px solid red;  
}
```

This div is smaller (width is 300px and height is 100px).

This div is bigger (width is also 300px and height is 100px).

Box-Sizing

- The CSS **box-sizing** property allows us to include the padding and border in an element's total width and height, so the element appears as you expect. To achieve this, we add the following to our web pages:

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

Box-Sizing

```
.div1 {  
  width: 300px;  
  height: 100px;  
  border: 1px solid blue;  
  box-sizing: border-box;  
}  
  
.div2 {  
  width: 300px;  
  height: 100px;  
  padding: 50px;  
  border: 1px solid red;  
  box-sizing: border-box;  
}
```

Both divs are the same size now!

Hooray!