

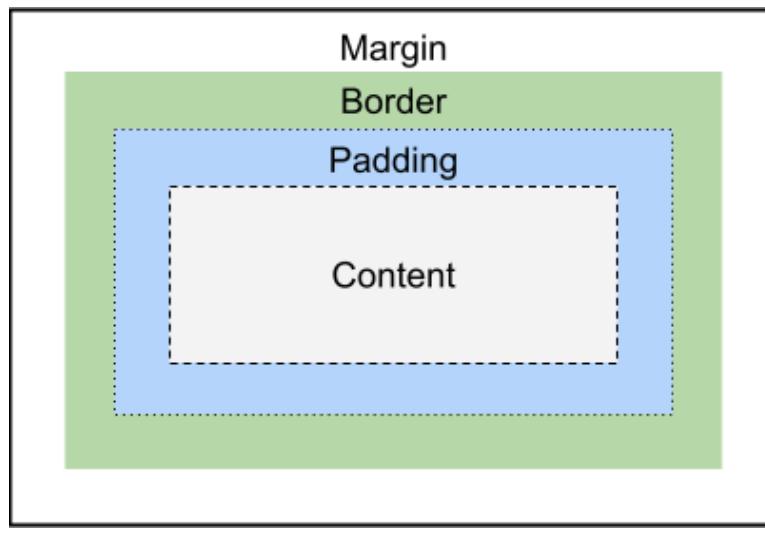
4.6 Box model

Box model components

HTML elements take up space on the webpage. The **box model** describes the size of each element as a series of nested boxes. The box model is important to understand when considering design and layout.

- **Content:** The innermost box contains the content of the element, such as text and images.
- **Padding:** The padding box contains the content box and adds a transparent area around the content.
- **Border:** The border box contains the padded content and adds an optionally colored area around the padding.
- **Margin:** The margin box contains all three boxes and adds a transparent area around the border.

Figure 4.6.1: Box model with labeled boxes.



[Feedback?](#)

The CSS properties that control the padding, border, and margin are:

- The **padding** property specifies the padding thickness. Ex: `padding: 5px;` creates a 5 pixel padding around the content.
- The **border** property specifies the border's thickness, style, and color. Ex: `border: 2px solid blue;` creates a solid blue border that is 2 pixels thick.

- The **margin** property specifies the margin thickness. Ex: **margin: 10px;** creates a 10 pixel margin.

The content, padding, margin, and border boxes are rendered according to the following rules:

- The content is displayed with a background according to the element's background color.
- The padding is transparent and will be displayed using the same color as the element's background.
- The border can be either colored or transparent. If transparent, the border will be displayed with the same color as the padding.
- The margin is transparent and will be displayed using the background color of the parent element.

PARTICIPATION ACTIVITY

4.6.1: Box model displayed.

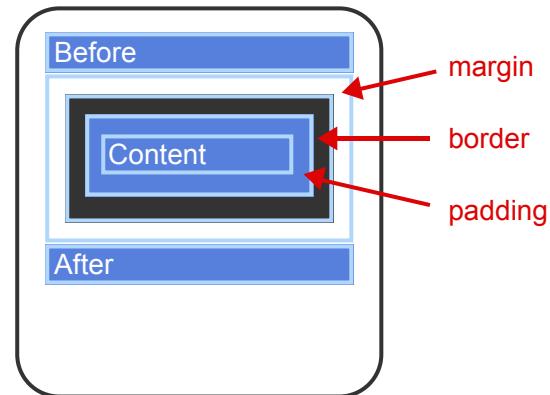

[Start](#)


2x speed

```
div { background-color: blue; }
#content {
  padding: 5px;
  border: 5px solid black;
  margin: 5px;
}
```



```
<div>Before</div>
<div id="content">Content</div>
<div>After</div>
```



Captions

- The box model describes the space each HTML element takes as a box. Each div element is displayed with a blue background, and each box is highlighted with a light blue outline.
- Padding surrounds the content element with transparent space, which is displayed using the element's background color.
- The border surrounds the content and padding and is colored black.
- The margin surrounds the border with transparent space and separates the element from other elements.

[Feedback?](#)
PARTICIPATION ACTIVITY

4.6.2: Choose the correct display color.



Given the HTML and CSS below, choose which color each box will be displayed.

```
<style>
div {
    background: white;
    border: 2px solid blue;
}

span {
    background: yellow;
    border: 1px solid black;
    padding: 2px;
    margin: 2px;
}
</style>

<div>This is a <span>test</span> of the emergency broadcast system.
</div>
```

If unable to drag and drop, refresh the page.

yellow	span element's padding Padding is transparent, so the span's padding is displayed using the span's yellow background color.	Correct
black	span element's border The span's border has a color specified, so the browser will display the border using the specified color.	Correct
white	span's margin Margins are transparent, so the span's margin uses the same color as the background color of the span's parent, the div.	Correct
blue	div's border The div's border has a color specified, so the browser will display the border using the specified color.	Correct

Reset

Feedback?

Box sizing

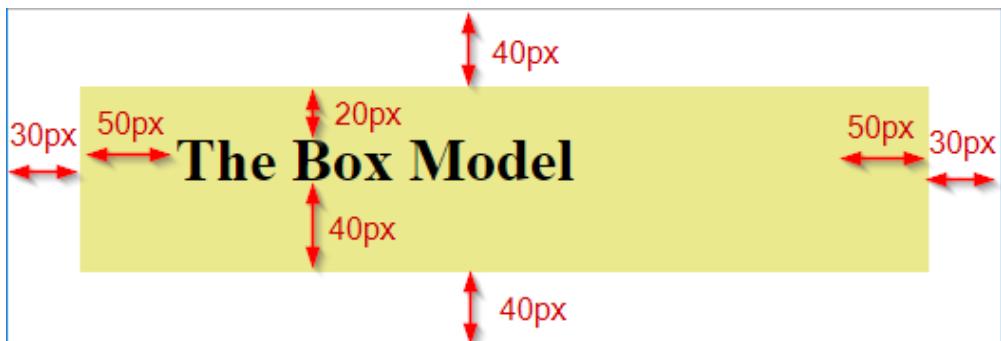
The `padding` and `margin` properties may have from 1 to 4 values:

- One value - Specifies uniform thickness around the box. Ex: `margin: 20px;` specifies 20px margin thickness around the box.
- Two values - Specifies top and bottom, right and left thickness. Ex: `margin: 10px 20px;` specifies 10px top and bottom margins and 20px right and left margins.
- Three values - Specifies top, right and left, and bottom thickness. Ex: `margin: 10px 30px 20px;` specifies 10px top margin, 30px right and left margins, and 20px bottom margin.
- Four values - Specifies top, right, bottom, and left thickness. Ex: `margin: 10px 30px 20px 40px;` specifies 10px top margin, 30px right margin, 20px bottom margin, and 40px left margin.

Example 4.6.1: Example with non-uniform margin and padding thickness.

```
<style>
  h1 {
    margin: 40px 30px;
    padding: 20px 50px 40px;
    background-color: khaki;
  }
</style>

<h1>The Box Model</h1>
```



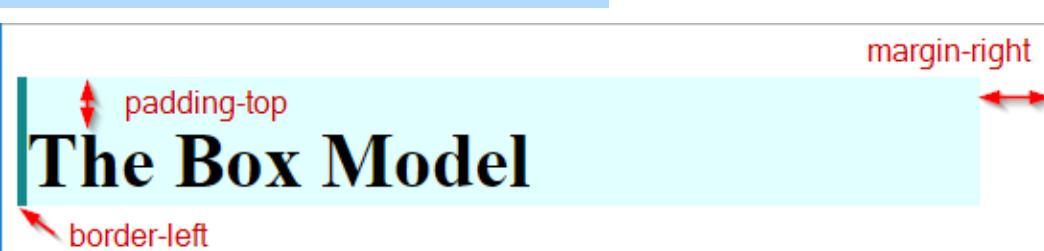
[Feedback?](#)

The `padding`, `border`, and `margin` CSS properties can apply only to one side by adding a `-top`, `-left`, `-bottom`, or `-right` suffix to the CSS property. Ex: `padding-top: 5px;` specifies 5 pixels of padding above the content.

Example 4.6.2: Example specifying one side of the box.

```
<style>
h1 {
    border-left: 5px solid darkcyan;
    margin-right: 30px;
    padding-top: 20px;
    background-color: lightcyan;
}
</style>

<h1>The Box Model</h1>
```

[Feedback?](#)**PARTICIPATION ACTIVITY**

4.6.3: Box model properties.



- 1) What CSS declaration sets the margin to have a uniform space of 20 pixels?

 //**Check****Show answer****Correct**

The margin property changes the element's margin. A single length value changes all margins (top, right, bottom, left) to have the same length.

- 2) What CSS declaration changes the top and bottom padding to be 20 pixels and right and left to be 30 pixels?

 //**Check****Show answer****Correct**

Specifying two **padding** values sets the element's top/bottom and right/left padding.

- 3) What CSS declaration changes the right padding to be 10 pixels wide?

 //**Check****Show answer****Correct**

The **padding-right** property changes only the element's right padding.

Content width and height

A block element's content spans the width of the enclosing parent element by default, but the content size can be changed with the **width** and **height** CSS properties:

- The **width** property specifies the content's width. Ex: `width: 20px;` makes the content 20px wide.
- The **height** property specifies the content's height. Ex: `height: 30px;` makes the content 30px high.

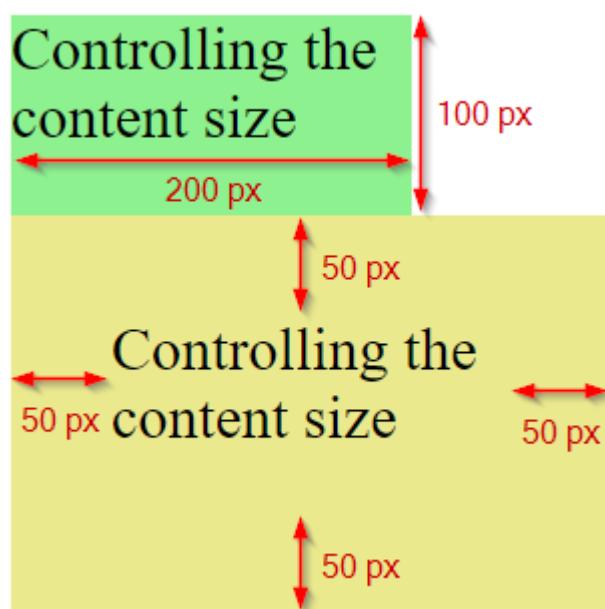
Example 4.6.3: Example using width and height properties.

Two `<div>` elements have the same **width** and **height** property values. However, the second `<div>` appears larger because of the additional padding.

```
<style>
#nopadding {
    width: 200px;
    height: 100px;
    background-color: lightgreen;
    font-size: 30px;
}

#padding {
    padding: 50px;
    width: 200px;
    height: 100px;
    background-color: khaki;
    font-size: 30px;
}
</style>

<div id="nopadding">Controlling the content size</div>
<div id="padding">Controlling the content size</div>
```



[Feedback?](#)

A common error is to use `width` or `height` on inline elements. An inline element like `` has a width and height that is equal to the size of the element's content. The width and height cannot be changed unless the inline element's `display` property is changed to `inline-block`.

Example 4.6.4: Example using width and height properties on an inline element.

```
<style>
#span1 {
    width: 80px; /* Ignored */
    height: 40px; /* Ignored */
    background-color: lightgreen;
}

#span2 {
    display: inline-block;
    width: 80px;
    height: 40px;
    background-color: khaki;
}
</style>

<p>
    <span id="span1">Span 1</span> and <span id="span2">Span
2</span>
</p>
```

Span 1 and Span 2

[Feedback?](#)

PARTICIPATION
ACTIVITY

4.6.4: Compute the size.





- 1) What is the total <div> width?

```
<style>
div {
    width: 200px;
    padding: 10px;
    border: 2px solid green;
    margin: 5px;
}
</style>

<div>How wide am I?
</div>
```

Correct

234

The total width of the element includes the content's width (200px) plus the thickness of padding ($2 \times 10\text{px}$), border ($2 \times 2\text{px}$), and margin ($2 \times 5\text{px}$) on left and right sides of the element.

234
pixels

[Check](#)

[Show answer](#)

- 2) What is the total <div> height?

```
<style>
div {
    height: 100px;
    margin: 10px 5px;
}
</style>

<div>How tall am I?
</div>
```

Correct

120

The total height of the element includes the content's height (100px) plus the thickness of padding (0px), border (0px), and margin on top (10px) and bottom (10px) sides of the element. Since the padding and border are not specified in the CSS rule, both have 0 thickness.

120
pixels

[Check](#)

[Show answer](#)



- 3) What is the total <div> width?

```
<style>
div {
    width: 100px;
    margin-left: 10px;
    margin-right: 15px;
    padding: 2px 5px
    10px 20px;
}
</style>

<div>How wide am I?
</div>
```

Correct

150

The total width of the element includes the content's width (100px), the left (10px) and right (15px) margin thickness, the padding right (5px) and left (20px) thickness, and twice the border thickness (0).

 //
pixels
Check**Show answer**

- 4) The default content width is 115px. What is the total width?

```
<style>
strong {
    width: 200px;
    margin: 5px;
}
</style>

<strong>How wide am I?
</strong>
```

Correct

125

Since is an inline element, the `width` property does not change the width. The total width is the content's width (115px) plus the margin thickness ($2 \times 5\text{px}$).

 //
pixels
Check**Show answer****Feedback?**

Chrome DevTools

The Chrome DevTools display the margin, border, padding, and content sizes for all elements. The screenshot below shows the size of the <h1> element with the content (509 × 37 pixels) at the center.

The screenshot shows a browser window titled "The Box Model" displaying the text "The Box Model" in a large, bold, dark red font. Below the browser window is the developer tools interface. The "Elements" tab is selected in the top navigation bar. The left panel shows the HTML structure:

```
<!doctype html>
<html>
  <head>...</head>
  <body>
    ... <h1>The Box Model</h1> == $0
  </body>
</html>
```

The right panel shows the "Styles" tab with the CSS properties for the selected `<h1>` element:

```
margin-top: 0px;
margin-bottom: 0px;
margin-left: 0px;
margin-right: 0px;
font-weight: bold;
```

Below the styles, there is a visual representation of the box model. It consists of a large orange dashed rectangle representing the margin. Inside it is a smaller yellow dashed rectangle representing the border. Inside that is a green dashed rectangle representing the padding. The innermost part is a blue solid rectangle representing the content area. The overall width of the box model is 509 pixels, and the height is 37 pixels.

Border styles

The **border** property combines the border width, style, and color. Each CSS property may also be specified separately:

- The **`border-width`** property specifies the border's width. Ex: `border-width: 5px;` specifies a border that is 5px thick.
- The **`border-style`** property specifies the border's style. Ex: `border-style: dashed;` specifies a border that is dashed.
- The **`border-color`** property specifies the border's color. Ex: `border-color: green;` specifies a border that is green.

Table 4.6.1: Common border styles.

Style	CSS example	Displayed
solid	<code>border: 2px solid black;</code>	
dashed	<code>border: 2px dashed red;</code>	

dotted	<code>border: 2px dotted blue;</code>	example
none	<code>border: 2px none green;</code>	example
double	<code>border: 5px double gray;</code>	example
groove	<code>border: 5px groove orange;</code>	example
ridge	<code>border: 5px ridge pink;</code>	example
inset	<code>border: 5px inset brown;</code>	example
outset	<code>border: 5px outset yellow;</code>	example

[Feedback?](#)**PARTICIPATION ACTIVITY****4.6.5: Modify the padding, border, and margin.**

Modify the CSS declarations for the `` and `<p>` tags to meet each requirement below:

1. Padding on the span: 5 pixels left, and 5 pixels right.
2. Border on the span: 2 pixels solid black above (use `-top` suffix), and 4 pixels solid gray below (use `-bottom`).
3. Border on the paragraph: 10 pixels, light green, ridge style (use `border-style`, `border-color`, and `border-width`).
4. Margin on the paragraph: uniform 5 pixels around.

[HTML](#) [CSS](#)

```
1 <p>
2 The typical filler text is the <span>Lorem Ipsum</span>.
3 According to lipsum.com, this text has been standard dummy text
4 </p>
```

Render webpageReset code**Your webpage**

The typical filler text is the Lorem Ipsum. According to lipsum.com, this text has been standard dummy text since the 1500s.

Expected webpage

The typical filler text is the Lorem I

Ipsum

. According to lipsum.com, this text has been standard dummy text since the 1500s.

▼ View solution

» Explain

--- START FILE: HTML ---

```
<p>
The typical filler text is the <span>Lorem Ipsum</span>.
According to lipsum.com, this text has been standard dummy
text since the 1500s.
</p>
```

--- END FILE: HTML ---

--- START FILE: CSS ---

```
/* write your declarations below */
span {
    padding-left: 5px;
    padding-right: 5px;
    border-top: 2px solid black;
    border-bottom: 4px solid gray;
}

p {
    padding: 10px;
    border-style: ridge;
    border-color: lightgreen;
    border-width: 10px;
    margin: 5px;
}
```

--- END FILE: CSS ---

[Feedback?](#)

Margin collapsing and horizontal centering

Vertical margins of two elements can sometimes combine or collapse into a single vertical space. Horizontal margins never collapse. Ex: Adjoining margins of two vertically stacked block elements collapse. The resulting margin size equals the top element's bottom margin or the bottom element's top margin, whichever is larger.

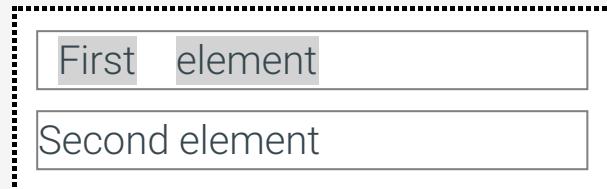
Example 4.6.5: Collapsed margins.

The following HTML demonstrates a collapsed margin. The vertical space between the inner div elements is collapsed to 10 pixels, while the horizontal space between the span elements is 20 pixels.

HTML and CSS

Rendered webpage

```
<style>
  .outer {
    border: 2px dotted black;
  }
  .inner {
    margin: 10px;
    border: 1px solid gray;
  }
  span {
    margin: 10px;
    background: lightgray;
  }
</style>
<div class="outer">
  <div class="inner">
    <span>First</span>
    <span>element</span>
  </div>
  <div class="inner">
    Second element
  </div>
</div>
```

[Feedback?](#)

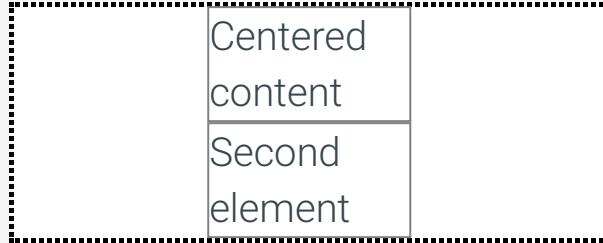
When an element has the `margin` property set to `auto`, the browser will compute the left and right margins to use up the remaining width of the parent element. The `auto` margin value is useful for horizontally centering an element in the parent element. A common error is to use `margin: auto` on elements that do not have the width specified or have `display: inline`, such as `span` elements.

Example 4.6.6: Horizontal centering.

HTML and CSS

Rendered webpage

```
<style>
  .outer {
    border: 2px dotted black;
  }
  .inner {
    margin: auto;
    width: 100px;
    border: 1px solid gray;
  }
</style>
<div class="outer">
  <div class="inner">
    Centered content
  </div>
  <div class="inner">
    Second element
  </div>
</div>
```

[Feedback?](#)**PARTICIPATION ACTIVITY****4.6.6: Margins.**

- 1) If two elements are stacked on top of each other and both have **margin: 20px**, the total margin between the two elements is 40px.

 True False**Correct**

Two vertical adjoining margins collapse into a single 20px margin.



- 2) If two elements are horizontally adjacent to each other and both have **margin: 20px**, the total margin between the two elements is 40px.

 True False**Correct**

Horizontal margins do not collapse, so the margins combine to 40px.



- 3) The text is horizontally centered in the web browser.

```
<body>
  <div style="width:300px; margin:auto;">Some text</div>
</body>
```

- True
 False



Correct

The `<div>` is horizontally centered, but the text is left-aligned in the `<div>`. The `text-align:center` property horizontally centers text.

[Feedback?](#)

CHALLENGE ACTIVITY

4.6.1: Box model.



530096.4000608.qx3zqy7

[Start](#)



1



2



3



4



5

[CSS](#) [HTML](#)

```
1 span {
2
3   /* Your solution goes here */
4
5 }
```

1

2

3

4

5

[Check](#)

[Next](#)

View your last submission