Chapter 7: The Box Model

Parameter	Detail
`content-box`	Width and height of the element only includes content area.
`padding-box`	Width and height of the element includes content and padding.
`border-box`	Width and height of the element includes content, padding and border.
`initial`	Sets the box model to its default state.
`inherit`	Inherits the box model of the parent element.

Section 7.1: What is the Box Model?

The Edges

The browser creates a rectangle for each element in the HTML document. The Box Model describes how the padding, border, and margin are added to the content to create this rectangle.

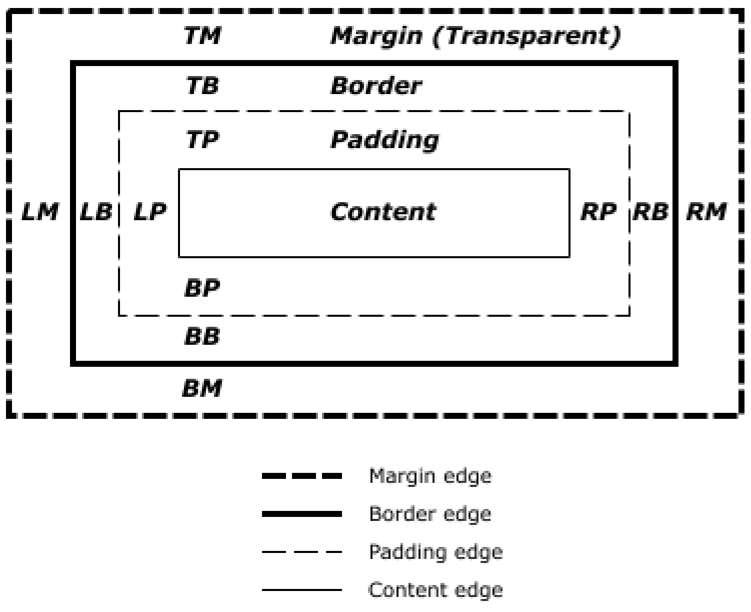


Diagram from CSS2.2 Working Draft

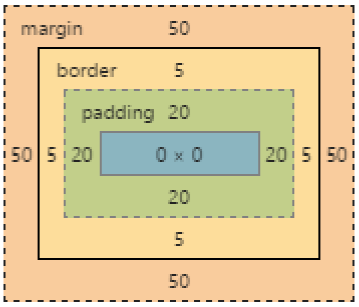
The perimeter of each of the four areas is called an *edge*. Each edge defines a *box.*

- The innermost rectangle is the **content box**. The width and height of this depends on the element's rendered content (text, images and any child elements it may have).
- Next is the **padding box**, as defined by the `padding` property. If there is no `padding` width defined, the padding edge is equal to the content edge.
- Then we have the **border box**, as defined by the `border` property. If there is no `border` width defined, the border edge is equal to the padding edge.
- The outermost rectangle is the **margin box**, as defined by the `margin` property. If there is no `margin` width defined, the margin edge is equal to the border edge.

Example

```
div {
    border: 5px solid red;
    margin: 50px;
    padding: 20px;
}
```

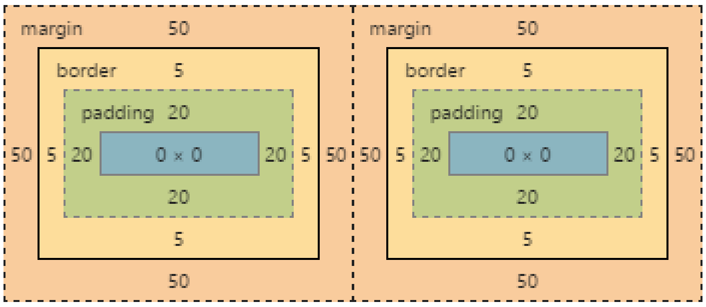
This CSS styles all `div` elements to have a top, right, bottom and left border of `5px` in width; a top, right, bottom and left margin of `50px`; and a top, right, bottom, and left padding of `20px`. Ignoring content, our generated box will look like this:



Screenshot of Google Chrome's Element Styles panel

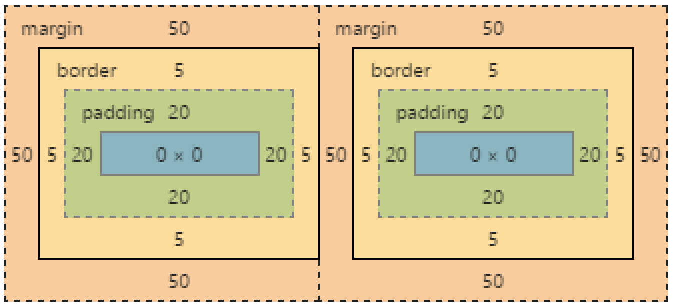
- As there is no content, the content region (the blue box in the middle) has no height or width (0px by 0px).
- The padding box by default is the same size as the content box, plus the 20px width on all four edges we're defining above with the `padding` property (40px by 40px).
- The border box is the same size as the padding box, plus the 5px width we're defining above with the `border` property (50px by 50px).
- Finally the margin box is the same size as the border box, plus the 50px width we're defining above with the `margin` property (giving our element a total size of 150px by 150px).

Now lets give our element a sibling with the same style. The browser looks at the Box Model of both elements to work out where in relation to the previous element's content the new element should be positioned:



The content of each of element is separated by a 150px gap, but the two elements' boxes touch each other.

If we then modify our first element to have no right margin, the right margin edge would be in the same position as the right border edge, and our two elements would now look like this:



Section 7.2: box-sizing

The default box model (`content-box`) can be counter-intuitive, since the `width` / `height` for an element will not represent its actual width or height on screen as soon as you start adding `padding` and `border` styles to the element.

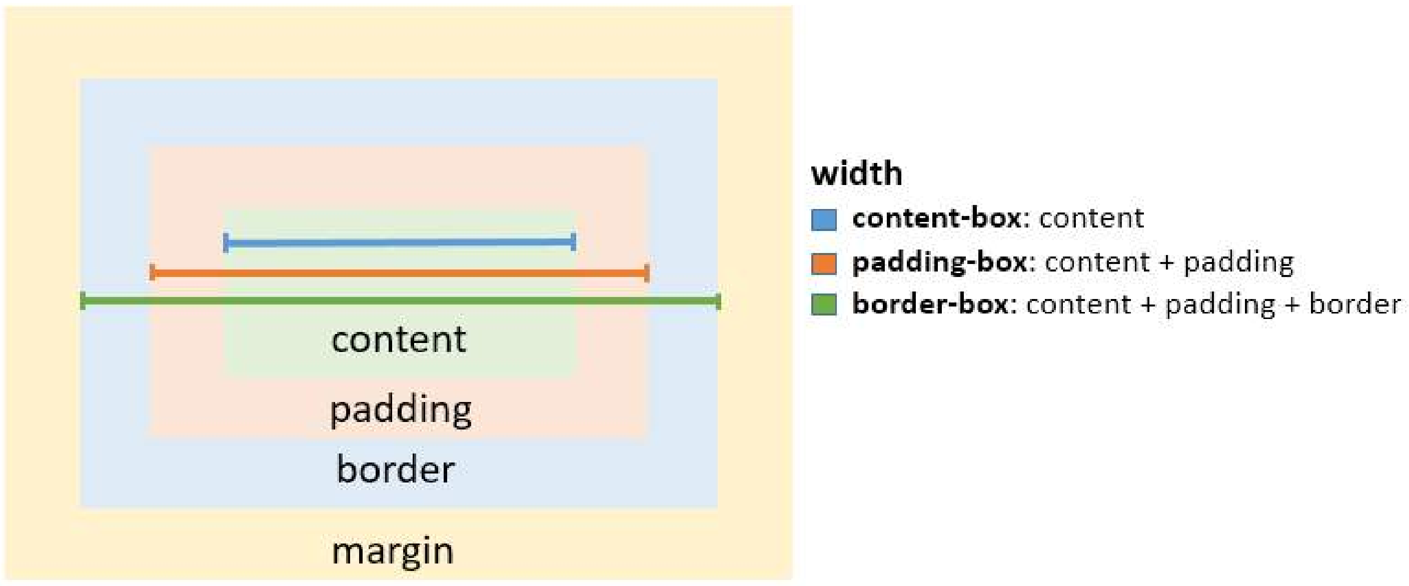
The following example demonstrates this potential issue with `content-box`:

```
textarea {
    width: 100%;
    padding: 3px;
    box-sizing: content-box; /* default value */
}
```

Since the padding will be added to the width of the textarea, the resulting element is a textarea that is wider than 100%.

Fortunately, CSS allows us to change the box model with the `box-sizing` property for an element. There are three different values for the property available:

- `content-box`: The common box model - width and height only includes the content, not the padding or border
- `padding-box`: Width and height includes the content and the padding, but not the border
- `border-box`: Width and height includes the content, the padding as well as the border


content
padding
border
margin
width
content-box: content
padding-box: content + padding
border-box: content + padding + border

To solve the `textarea` problem above, you could just change the `box-sizing` property to `padding-box` or `border-box`. `border-box` is most commonly used.

```
textarea {
    width: 100%;
    padding: 3px;
    box-sizing: border-box;
}
```

To apply a specific box model to every element on the page, use the following snippet:

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

*, *:before, *:after {
    box-sizing: inherit;
}
```

In this coding `box-sizing:border-box;` is not directly applied to `*`, so you can easily overwrite this property on individual elements.