CSS Box Model

Everything in CSS has a box around it, and understanding these boxes is key to being able to create layouts with CSS.

**Block and inline boxes**

In CSS we broadly have two types of box — **block boxes** and **inline boxes**. These characteristics refer to how the box behaves in terms of page flow, and in relation to other boxes on the page:

If a box is defined as a block, it will behave in the following ways:

1. The box will extend in the inline direction to fill the space available in its container. In most cases this means that the box will become as wide as its container, filling up 100% of the space available.
2. The box will break onto a new line.
3. The [width](https://developer.mozilla.org/en-US/docs/Web/CSS/width) and [height](https://developer.mozilla.org/en-US/docs/Web/CSS/height) properties are respected.
4. Padding, margin and border will cause other elements to be pushed away from the box

Unless we decide to change the display type to inline, elements such as headings (e.g. <h1>) and <p>s all use block as their outer display type by default.

If a box has an outer display type of inline, then:

1. The box will not break onto a new line.
2. The [width](https://developer.mozilla.org/en-US/docs/Web/CSS/width) and [height](https://developer.mozilla.org/en-US/docs/Web/CSS/height) properties will not apply.
3. Padding, margin and borders will apply but will not cause other inline boxes to move away from the box.

The <a> element, used for links, <span>, <em> and <strong> are all examples of elements that will display inline by default.

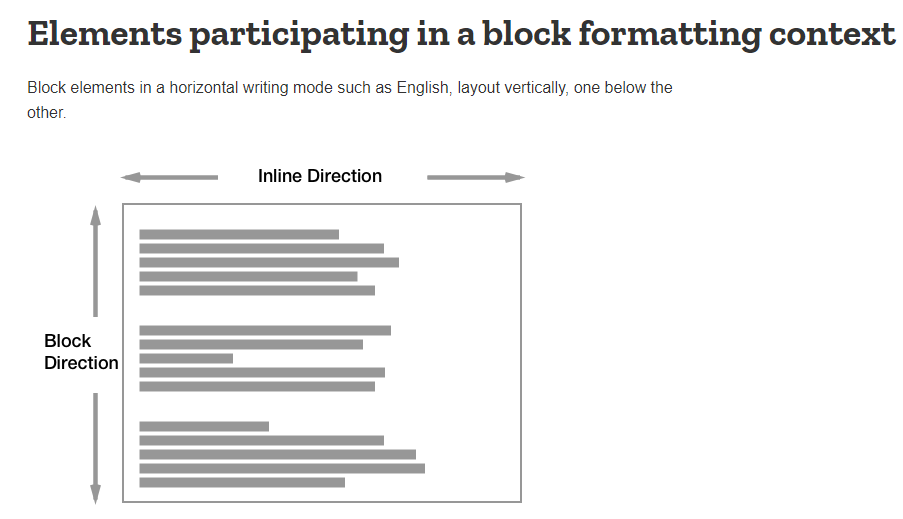
The type of box applied to an element is defined by [display](https://developer.mozilla.org/en-US/docs/Web/CSS/display) property values such as block and inline, and relates to the **outer** value of display.

## Aside: Inner and outer display types

At this point we'd better also explain **inner** and **outer** display types. As mentioned above, boxes in CSS have an outer display type, which details whether the box is block or inline.

Boxes also have an inner display type, however, which dictates how elements inside that box are laid out. By default the elements inside a box are laid out in [**normal flow**](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Normal_Flow), which means that they behave just like any other block and inline elements (as explained above).

We can however, change the inner display type by using display values like flex. If we set display: flex; on an element, the outer display type is block, but the inner display type is changed to flex. Any direct children of this box will become flex items, and will laid out according to the rules set out in the [Flexbox](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox) spec, which you'll learn about later on.

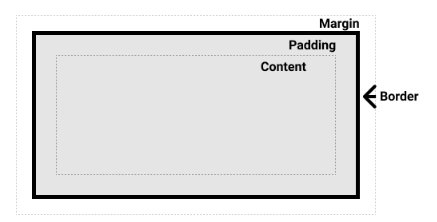


**Parts Box:**

Making up a block box in CSS we have the:

1. **Content box**: The area where your content is displayed, which can be sized using properties like [width](https://developer.mozilla.org/en-US/docs/Web/CSS/width) and [height](https://developer.mozilla.org/en-US/docs/Web/CSS/height).
2. **Padding box**: The padding sits around the content as white space; its size can be controlled using [padding](https://developer.mozilla.org/en-US/docs/Web/CSS/padding) and related properties.
3. **Border box**: The border box wraps the content and any padding. Its size and style can be controlled using [border](https://developer.mozilla.org/en-US/docs/Web/CSS/border) and related properties.
4. **Margin box**: The margin is the outermost layer, wrapping the content, padding and border as whitespace between this box and other elements. Its size can be controlled using [margin](https://developer.mozilla.org/en-US/docs/Web/CSS/margin) and related properties.

The below diagram shows these layers:

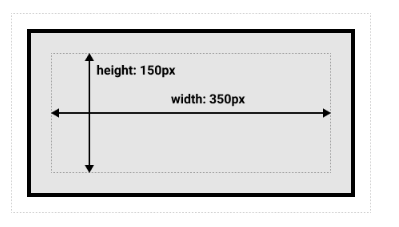


**Standard CSS Box model**

By default, browsers use the standard box model.

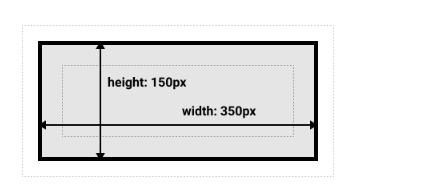
**Width & Height**

In the standard box model, if you give a box a width and a height, this defines the width and height of the content box.



**Alternative CSS Box model**

By default, browsers use the standard box model. If you want to turn on the alternative model for an element you do so by setting box-sizing: border-box on it. By doing this you are telling the browser to take the border box as the area defined by any size you set.



**Margins**

We can control all margins of an element at once using the [margin](https://developer.mozilla.org/en-US/docs/Web/CSS/margin) property, or each side individually using the equivalent longhand properties:

1. [margin-top](https://developer.mozilla.org/en-US/docs/Web/CSS/margin-top)
2. [margin-right](https://developer.mozilla.org/en-US/docs/Web/CSS/margin-right)
3. [margin-bottom](https://developer.mozilla.org/en-US/docs/Web/CSS/margin-bottom)
4. [margin-left](https://developer.mozilla.org/en-US/docs/Web/CSS/margin-left)

**Border**

To set the width, color, and style of each side individually, you can use:

1. [border-top](https://developer.mozilla.org/en-US/docs/Web/CSS/border-top)
2. [border-right](https://developer.mozilla.org/en-US/docs/Web/CSS/border-right)
3. [border-bottom](https://developer.mozilla.org/en-US/docs/Web/CSS/border-bottom)
4. [border-left](https://developer.mozilla.org/en-US/docs/Web/CSS/border-left)

To set the color, style, or width of all sides, use the following:

1. [border-width](https://developer.mozilla.org/en-US/docs/Web/CSS/border-width)
2. [border-style](https://developer.mozilla.org/en-US/docs/Web/CSS/border-style)
3. [border-color](https://developer.mozilla.org/en-US/docs/Web/CSS/border-color)

**Padding**

We can control the padding on each side of an element individually using the [padding](https://developer.mozilla.org/en-US/docs/Web/CSS/padding) property, or each side individually using the equivalent longhand properties:

1. [padding-top](https://developer.mozilla.org/en-US/docs/Web/CSS/padding-top)
2. [padding-right](https://developer.mozilla.org/en-US/docs/Web/CSS/padding-right)
3. [padding-bottom](https://developer.mozilla.org/en-US/docs/Web/CSS/padding-bottom)
4. [padding-left](https://developer.mozilla.org/en-US/docs/Web/CSS/padding-left)

## Using display: inline-block

There is a special value of display, which provides a middle ground between inline and block. This is useful for situations where you do not want an item to break onto a new line, but do want it to respect width and height and avoid the overlapping seen above.

An element with display: inline-block does a subset of the block things we already know about:

1. The width and height properties are respected.
2. padding, margin, and border will cause other elements to be pushed away from the box.

