Box sizing : border box

Why ?

Although this is the default method of calculating the space occupied by an element in HTML, this isn’t very favorable when it comes to designing our elements. Say you have an element that needs to have a fixed total width of 300px. You would first specify its width property in CSS as 300px. However, if you wish to add a border or padding to this element, since they add to the total width of the element, you would also need to reduce the width property of the element accordingly in order to maintain the fixed total width of 300px. In order to overcome this inconvenience, we have a property in CSS called box-sizing.

[The CSS box-sizing property. In HTML, by default, every element is a… | by Kabir Nazir | Level Up Coding (gitconnected.com)](https://levelup.gitconnected.com/css-box-sizing-property-and-how-it-works-f7ac1c3a76b0)

Min height

The min-height property is used to set the minimum height of an element. It overrides the height property if the value of the height property becomes smaller than its value, thus preventing the height of an element from becoming smaller than the min-height value

the min-height property provides a minimum height for an element.

This occurs when the viewport shrinks and the element's height cannot be reduced beyond a defined height unit.

Height : 100%;

In simple words, **height: 100%;** means that the element's height should be the same as the height of its parent container. It's like saying, "Make this element as tall as its parent container."

Imagine you have a box (the element) inside another larger box (the parent container). If you set the height of the smaller box to **100%**, it will stretch to fill the entire height of the larger box, making them the same height.

For example, consider the following HTML and CSS:

HTML:

htmlCopy code

<div class="parent"> <div class="child">Hello, I'm a child!</div> </div>

CSS:

cssCopy code

.parent { height: 300px; /\* The parent box is 300px tall \*/ background-color: lightblue; } .child { height: 100%; /\* The child box will be as tall as its parent (300px) \*/ background-color: lightgreen; }

In this example, the child box will be 300 pixels tall, just like its parent box. The **height: 100%;** rule ensures that the child box takes up the entire height of its parent.

Difference between background size cover and contain

?

**Keyword values**

Using the keyword values contain and cover, we can change the size of a background image. Let’s look at some examples of each keyword value.

**contain**

When we set the contain value on the background-size property, the background image will resize to ensure the image is fully visible. This keyword value scales the image to fit within its container as much as possible without cropping or stretching.

If the container element is larger than the image, this will result in image tiling (in which multiple copies are displayed side-by-side) to fill the negative space, unless the background-repeat property is set to no-repeat:

In the example above, the container element is larger than the image. So, if we hadn’t set the background-repeat property to no-repeat, our image would tile in an attempt to cover the entire container. It does this, rather than stretching, in order to preserve image quality.

**cover**

The cover value, as the name implies, ensures that the picture is scaled to fill the full container, leaving no empty space, while maintaining its ratio. If the proportions of the background image and the element differ, the image’s width or height will be cropped:

Notice how the above image is stretched to completely fill the container, even if its width or height is cropped.

***N.B.,*** background-size *can be set to* auto*; this is the property’s default value and will result in the image being displayed as its original size*

https://www.smashingmagazine.com/2021/10/object-fit-background-size-css/