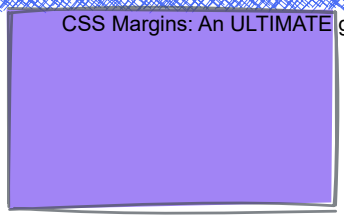


CSS Margins: An ULTIMATE guide

Published on September 26, 2022 by Gauri Shanker

✓ LAST UPDATED ON OCTOBER 16TH, 2023

Margin in CSS is the blank space outside of an element that separates it from other elements.

Margins are different from padding which creates space within the element.

In this comprehensive guide, we cover everything – margin declaration syntax, margin shorthand property, negative margins, collapsing margins, the difference

What is a Margin in CSS?

Margin in CSS is the blank space outside an element's borders.

Margin is generally used to put elements at a distance from each other. Margins are essential to produce a clean, uncluttered look for your web pages.

CSS Margin Syntax

CSS allows us to declare margin on all four sides — `top, right, bottom and left`.

```
div {
    margin-top: 20px;
    margin-right: 30px;
    margin-bottom: 40px;
    margin-left: 50px;
}
```

These lines of code will produce this output —

margin-bottom

CSS Margin Shorthand Property

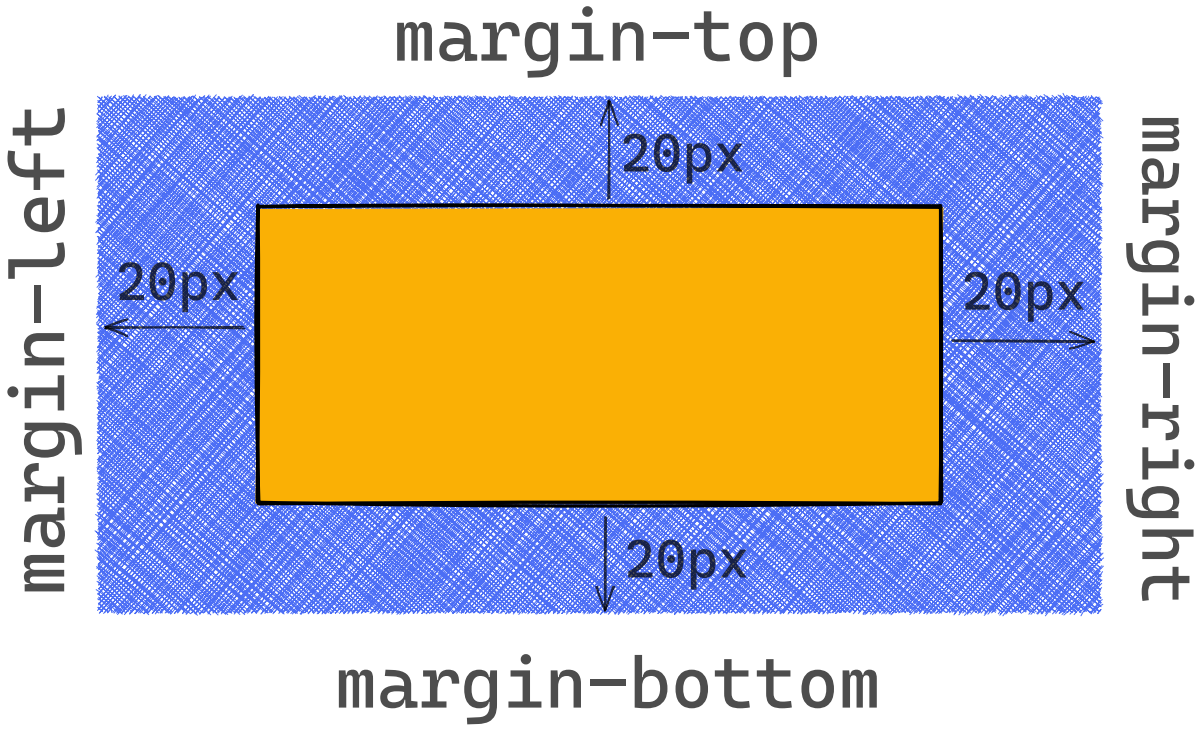
Writing all four properties every time can get tedious, therefore CSS comes with a shorthand property named `margin`.

Using this shorthand, you can set the values of all four properties at once, like this —

```
div {
    margin: 20px 30px 40px 50px;
}
```

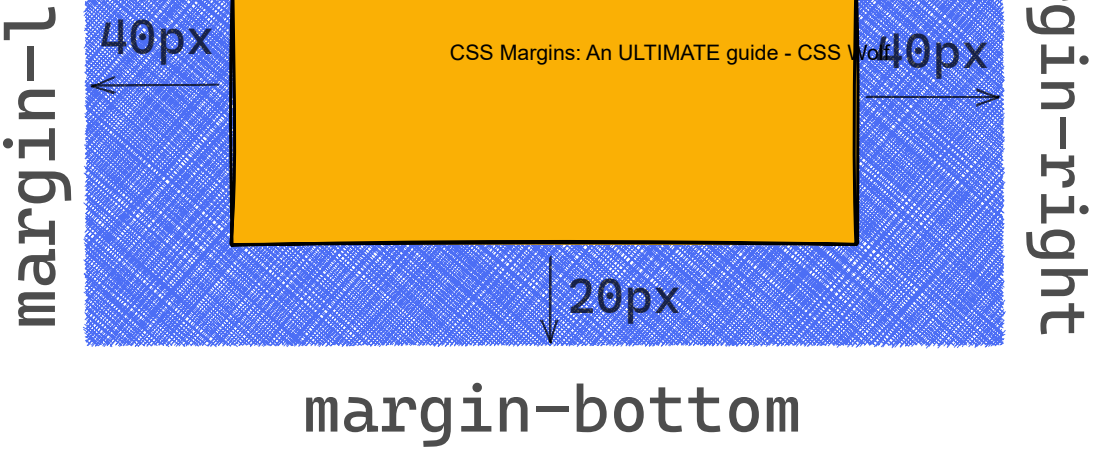
This margin shorthand property can take one or two or three or four values. The margins are applied differently in every case.

Let's discuss them one by one.



Margin shorthand with two values

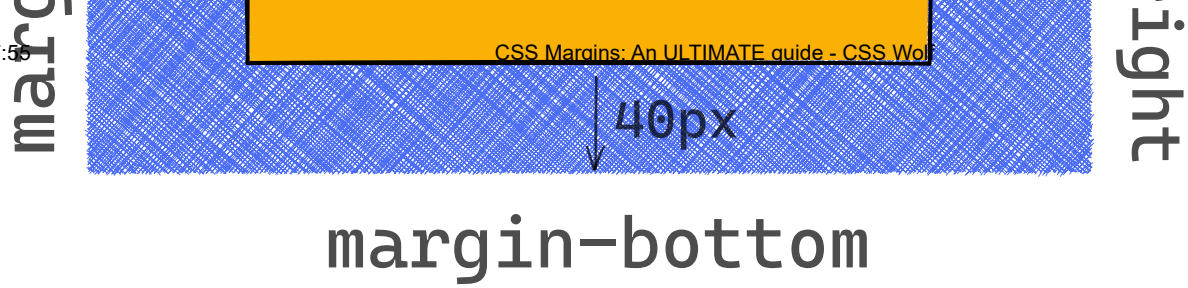
```
div {
    margin: 20px 40px;
}
```



Margin shorthand with three values

```
div {
    margin: 20px 30px 40px;
}
```

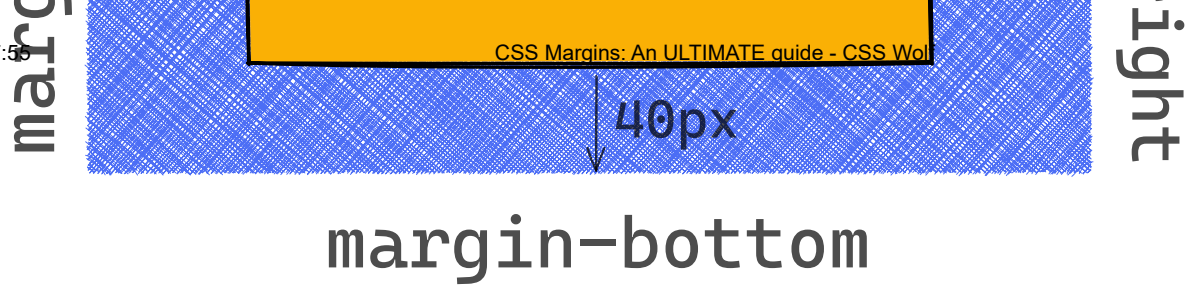
When the `margin` shorthand property is given three values, the first value (`20px`) is applied to the top, the second value (`30px`) is applied to the left and right, and the third value (`40px`) is applied to the bottom.



Margin shorthand with four values

```
div {
    margin: 20px 30px 40px 50px;
}
```

When the `margin` shorthand property is given four values, they are applied in a clockwise manner, starting from the top.



> The pattern in which these shorthand values are applied is not unique to the `margin` property. In fact, the declaration of a few other shorthand properties such as `padding` and `border-radius` work almost the same way.

Acceptable values for CSS Margin

Margin property in CSS defines the space between an element's border and other elements therefore it only accepts values in length units.

There are several ways the margin property can take values. Let's discuss them one by one.

Default value

```
    </div>
</div>
```

I have given the `.parent div` some margins. On the child element, I have declared `margin: inherit`. I have also given some borders to these `div`s so that we can clearly see them.

```
.grandpa {
    border: 5px solid lightpink;
}

.parent {
    margin: 40px 80px;
    border: 2px solid lightblue;
}

.child {
    margin: inherit;
    border: 2px solid lightgreen;
    padding: 30px; /* sets width and height of 60px each */
}
```

This is what's happening —

Setting `margin:inherit` *makes the child inherit margins from its parent.*

Margins can be set using fixed length values

As we already stated that the `margin` property defines the space outside an element's borders, Obviously, it accepts fixed length values like pixels (`px`), points (`pt`), centimeters (`cm`) etc.

Margins can be set using percentages

```
<div class="parent">
    <div class="child">Child</div>
</div>
```

I set the width to `400px` on the parent. Also, I have set a margin `20%` on the child. I have also set a few borders around the elements so that we can clearly see them.

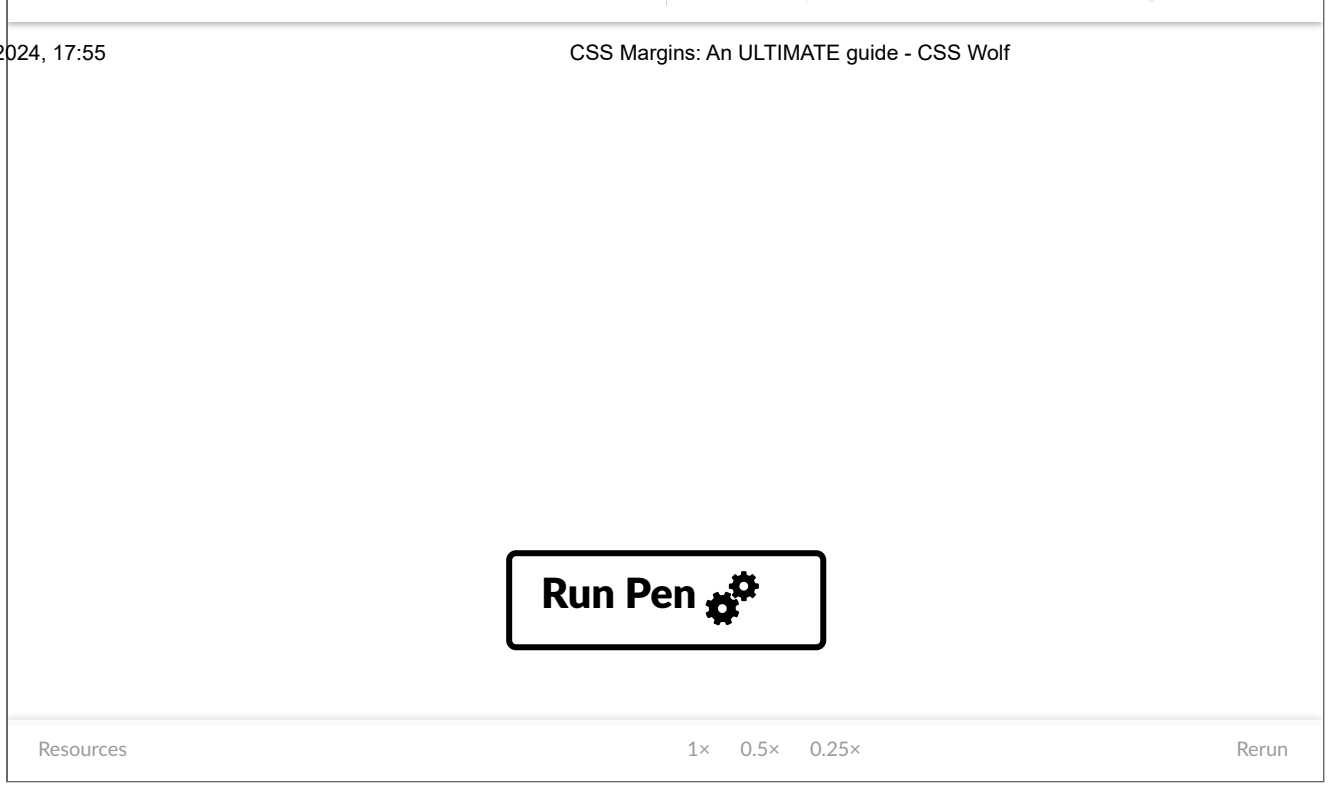
```
.parent {
    border: 5px solid lightblue;
    width: 400px;
}

.child {
    margin: 20%;
    border: 5px solid green;
}
```

This is what we expected. We have given the child element a margin of `20%` which equals `80px` when calculated relative to the width (`400px`) of its containing block.

You might have observed that we haven't given any height to the parent `div` . Even if we explicitly set a height to it, the margin on the child is still calculated relative to its width.

To demonstrate this, I give the parent `div` a height of `300px` in the example above. I have also introduced a second child to better convey my point.

```
<div class="parent">
    <div class="child">Child</div>
    <div class="child">Second Child</div>
</div>
```



If you use a screen-measuring tool, you will observe that the `child` element has the same margins as before. This is because the width of the parent block remained the same, only the height changed.

Margin can be set to `auto`

The margin property in CSS accepts `auto` as one of the values. The browser then automatically calculates the margins.

```
    <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Quae quic
    <p>lorem20Lorem ipsum dolor sit, amet consectetur adipisicing elit. Qu
    <p>lorem20Lorem ipsum dolor sit, amet consectetur adipisicing elit. Qu
</div>
```

You decide that —

- the paragraphs on the page should occupy about `80%` of the width,
- they should have top and bottom margins of `25px`, and
- they should be horizontally centered on the page.

You write the following code —

```
.page {
    border: 3px solid black;
}

p {
    width: 80%;
    /* Set vertical margins to 20px and horizontal margins to auto */
```

Resources 1× 0.5× 0.25× Rerun

You will observe that the paragraphs are automatically centered on the page even if we have not explicitly centered them.

Try resizing the browser window. You will see that the paragraphs always remain centered and within the viewing area. That's because setting horizontal margins to `auto` gives the browser a free hand to automatically calculate the margins and give us the best possible result.

Remember that there are some caveats to using this method —

1. The element-to-be-centered (paragraph in this case) must be block-level.
2. It must not have a width of `100%` otherwise, the browser won't have any horizontal space to create margins. (Or you can think of it like this — *the*

```
    <p class="width-auto">This paragraph has width auto.</p>
    <p class="width-80p">This paragraph has width 80%</p>
</div>
```

The first one has `width: 100%`, the second one `width: auto` and the third one `width: 80%`.. Also, I have set the same margin on all paragraphs — `20px` vertical and `auto` horizontal.

```
p {
    margin: 20px auto;
    border: 3px solid lightgreen;
}

.width-100p {
    width: 100%;
}

.width-auto {
    width: auto;
}

.width-80p {
```

You will observe that only the paragraph with the width `80%` is centered on the page.

> ⚠ **A word of caution:** Note that there are still a few cases where `margin:auto` might not work as expected. Check out this excellent thread on StackOverflow for complete details.

Negative margins

The margin property in CSS also accepts negative values.

> Negative values draw the element closer to its neighbors than it would be by default.

pull
elements
closer
than
they
would
have
been in
normal
flow.

You will observe that the red block has shifted to the left and encroached upon the margin space of the green block. Similarly, the blue block has also shifted to the left and is now overlapping the red block partially.

I have also put together an animation showing the workings of negative margin in action.

The margin on the red block is alternating between `0px` and `-40px`.

Can CSS Margins be Animated?

CSS margin property *is* animatable (we saw a demo just above 👆).

Nevertheless, I will take another example.

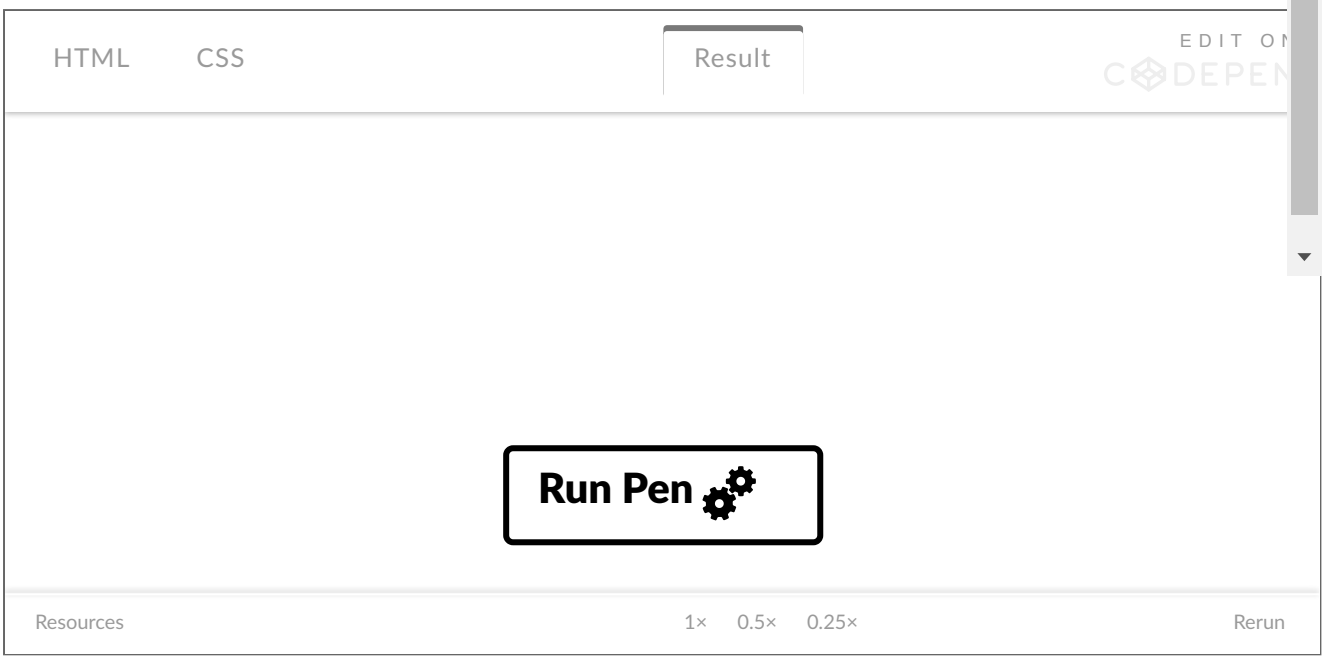
I have taken two span elements —

```
<span class="green"> GREEN </span>
<span class="red"> RED </span>
```

I have applied some styles to these blocks and written a single keyframe. Here's my CSS and keyframe definition —

```
span {
    /* gives width and height of 100px each */
    padding: 50px;
    display: inline-block;
}

.green {
```

Here's the final result —



Observe that I have animated the margin property *only on the green block, not the red one*. The red block is moving only to occupy the void space vacated by the green block during the animation.

Although you can animate the `margin` property, it is not advised to do so in your real-world projects.

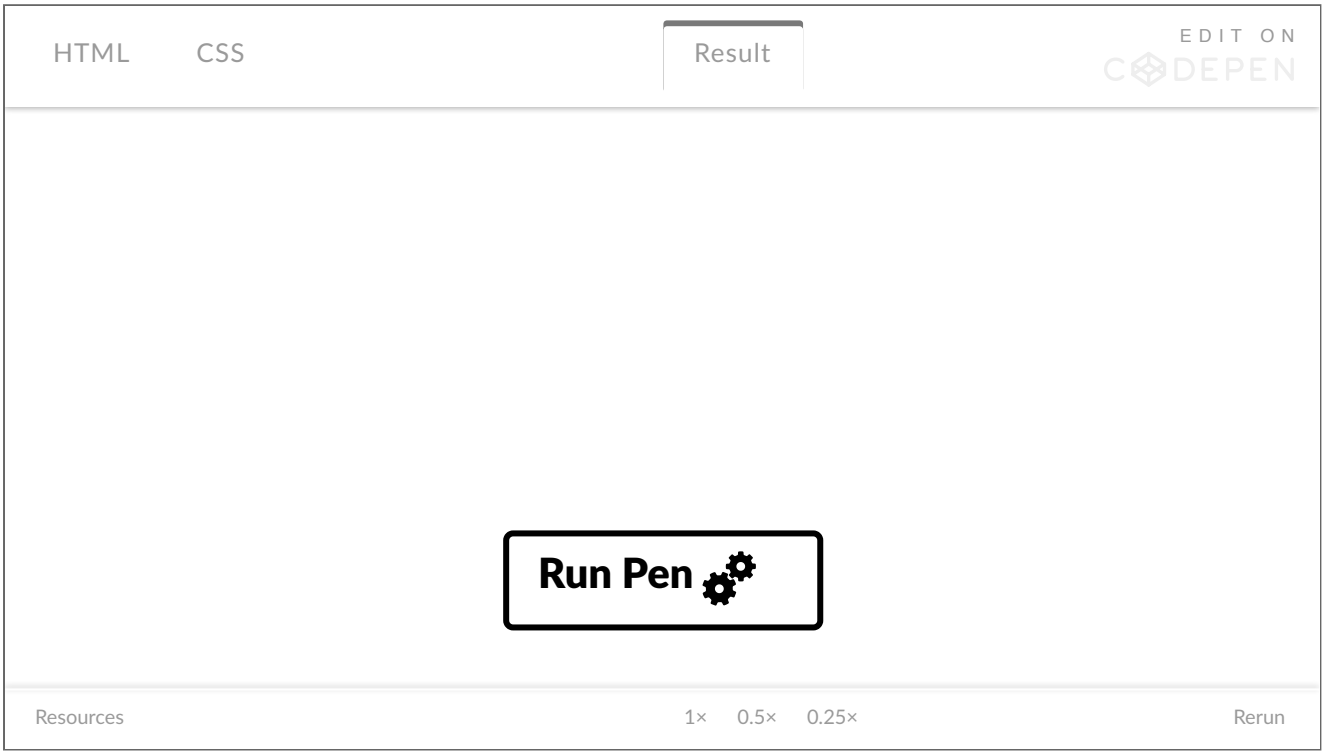
```
        <p></p>
        <p></p>
    </div>
```

```
p {
    margin: 30px;
}
```

If margins didn't collapse, this code would have created an unintended effect — any two adjacent paragraphs would have `60px` of space between them.

Collapsing margins prevent any unintended blank space between adjacent elements.

To avoid this extra space, the browser intelligently collapses these vertical margins and applies the greater of the two (in this case, they both are equal).



CSS Margin vs Padding

Both, margin and padding are used to create blank space but there is a difference — margin is the space *outside* an element's border whereas padding is the space *within* the element's border.

The yellow-shaded region shows margin whereas the blue-shaded region shows padding.

There are a few more subtle differences between CSS Margin and Padding. Let's discuss them.

Padding can be styled but not margin

As padding is considered part of an element, styles such as `background-color` also affect the padding area. Not so with margins which are always transparent.

Padding doesn't collapse but the margin does

As we discussed above, vertical margins sometimes collapse but padding never does.

Margin can be negative but not padding.

As we discussed above, the negative values are acceptable for the margin property. Not so with padding.

and its contents (or children).

Use padding to space an element and its contents.

And use margin when you want to separate sibling elements.

Use margins when putting up space between an element and its siblings.

TL; DR

In this article, we discussed everything about the `margin` property in CSS, how it is declared, its shorthand syntax, the values it can accept, and much more.

Here's the quick summary of all that in digestible Q&A format.

What is a CSS Margin?

Margin is the blank space outside an element's border. It does not count in an element's size.

How do you use declare margin in CSS?

There are four properties in CSS to declare margins — `margin-top`, `margin-right`, `margin-bottom` and `margin-left`.

Negative margins draw the element closer to other elements than it would have normally been.

Why do margins collapse in CSS?

When the browser detects that you are creating more space between two adjacent elements than you intended, it intelligently collapses two margins and keeps only the greater of the two.

How to change the margin color in CSS?

Margins are always transparent.

There is no property in CSS to apply styles on the margins. You can however use `box-shadow` or `outline` property to achieve some similar effects.

What does `margin:auto` mean in CSS?

The `auto` value instructs the browser to calculate margins by itself.

It is most useful when we want to center an element that is less wide than its container.

3. **fixed length —** like pixels (`px`), points (`pt`), centimeter (`cm`) etc.
4. **Percentages —** when defined in percentage, margins are calculated relative to the *width of the containing element*.
5. **Inherit —** the child will inherit margins from its parent.
6. **Negative values —** negative margins pull an element closer to other elements than it would have been in normal flow.

Wrapping up

This wraps up our discussion of the CSS margin property. I hope this guide helped you grasp and retain the concepts.

If you found it useful, don't forget to share it with your friends.

Did I forget something or make a typo? Kindly drop a line in the comments and I will immediately correct it.

Thanks for reading.

CSS

css margin

About

I am Gauri Shanker, a coding enthusiast. I learn new things almost every day and share them through this blog.

Articles on this blog may contain some affiliate links. It means if you buy something from these links, I may get a commission at no extra cost to you.

Follow me on Twitter and Pinterest.

Popular Posts

How to Move a Fullscreen App to Second Monitor on Mac

How to Change Case of Text on Clipboard on Mac

How to Select Text Vertically on a Mac

How to Access Menu Items From the Keyboard on Mac