

grouping & box model

element structure

Simone Ferretti - Valentin Sorohan - Vincenzo Bonura

Flow

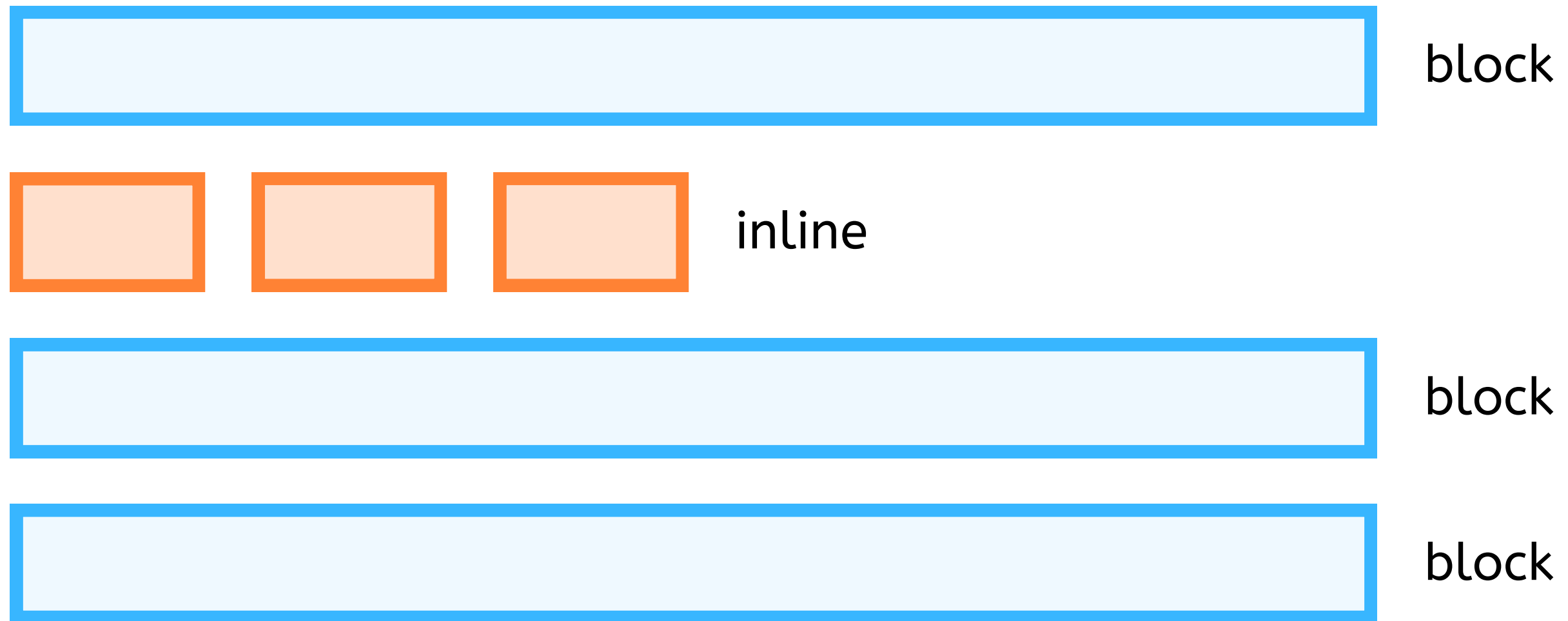
y a H H
o r o i
u e w !
?

Flow

Hi!

How are you?

Block or Inline



Block

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

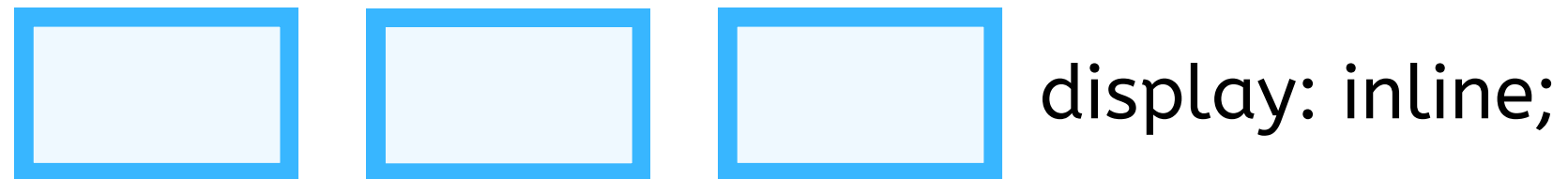
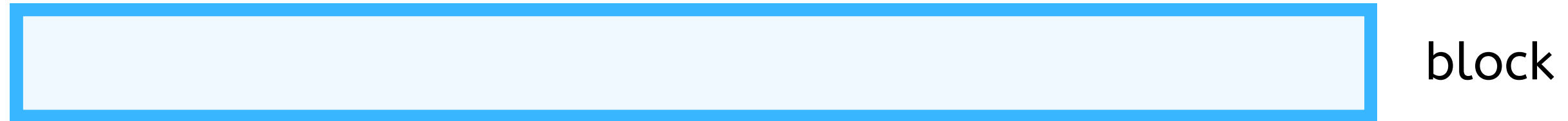
- The box will **break onto a new line**.
- The width and height properties are **respected**.
- Padding, margin and border will cause other elements to be pushed away from the box.
- If width is not specified, the box will **extend in the inline** direction to **fill the space** available in its container.

Inline

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

- The box will **not break** onto a new line.
- The width and height properties will **not apply**.
- Top and bottom padding, margins, and borders will apply but will not cause other inline boxes to move away from the box.
- Left and right padding, margins, and borders will apply and will cause other inline boxes to move away from the box.

Block, Inline or...



Is an **inline-block** ** the same as an **inline-block** *<div>*?

display: **block**;

The element generates a **block box**, generating **line breaks** both before and after the element when in the normal flow.

display: **inline**;

The element generates one or more **inline boxes** that **do not generate line breaks** before or after themselves. In normal flow, the next element will be on the same line if there is space.

display: **inline-block**;

The element generates a **block box** that will be **flowed with surrounding content** as if it were a single inline box (behaving much like a replaced element would).

display: **none**;

Turns off the display of an element so that **it has no effect on layout** (the document is rendered as though the element did not exist). All descendant elements also have their display turned off.

Css properties

Property: *text-decoration*

The `text-decoration` property sets the appearance of decorative lines on text

Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do

Properties

`text-decoration-color:` red

`text-decoration-style:` solid

`text-decoration-line:` underline

`text-decoration-thickness:` 3px

Shorthand property

`text-decoration:` solid underline red 3px

[MDN | Text-decoration](#)

Property: *text-align*

The `text-align` property sets the horizontal alignment of the content inside a block element or table-cell box.

Properties

`text-align: start;`
/*Aligns text according
to the direction of document*/

`text-align: left;`
/*Aligns the text to
the left of the container*/

`<html dir="rtl">`

Lorem ipsum dolor sit amet,
 consectetur adipiscing
 elit, sed

Lorem ipsum dolor sit amet,
 consectetur adipiscing
 elit, sed

Property: text-align

The `text-align` property sets the horizontal alignment of the content inside a block element or table-cell box.

Properties

```
text-align: end;  
/*Aligns text according  
to the direction of document*/
```

```
text-align: right;  
/*Aligns the text to  
the right of the container*/
```

```
<html dir="ltr">
```

```
    Lorem ipsum dolor sit amet,  
                                consectetur adipiscing  
                                elit, sed
```

```
    Lorem ipsum dolor sit amet,  
                                consectetur adipiscing  
                                elit, sed
```

[MDN | Text-align](#)

Property: cursor

The `cursor` property sets the mouse cursor, if any, to show when the mouse pointer is over an element.

```
.myClass {  
  cursor: help;  
  background-color: yellow;  
  text-align: center;  
}
```

I auto	↕ move	👉🚫 no-drop	↕ col-resize
🔍 all-scroll	👉 pointer	🚫 not-allowed	↕ row-resize
+ crosshair	🕒 progress	↔ e-resize	↗ ne-resize
🖱 default	I text	↑ n-resize	↖ nw-resize
🖱? help	↕ vertical-text	↓ s-resize	↘ se-resize
I inherit	🕒 wait	↔ w-resize	↙ sw-resize

Property: Width

- Sets the width of a **block-level** element or **img**
- Doesn't work for inline elements, unless their display property is changed
- Accepts a variety of length units

```
#sidebar {  
  width: 200px;  
  width: 20em; /* relative to font size */  
  width: 20%; /* relative to containing element width */  
  width: 20vw; /* relative to viewport: 1vw = 1% viewport width */  
}
```

The most used are: px, rem, em, vw, vh, % (percentage)

Property: Height

Works like width, with all the same units

```
.niceClass {  
  height: 200px;  
  height: 20em; /* relative to font size */  
  height: 20%; /* relative to containing element width */  
  height: 20vw; /* relative to viewport: 1vw = 1% viewport width */  
}
```

[The lengths of CSS](#)

[CSS Units](#)

[MDN | Values and Units](#)

What about **inline-size** and **block-size**?

Min & max properties

- Set upper or lower **limits** to the size of elements
- An element cannot be **smaller** than its **min-width** or **min-height**
- An element cannot be **larger** than its **max-width** or **max-height**

Want to play?



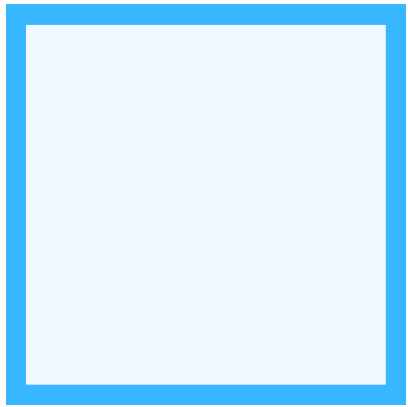
Box Model Playground

Site to test some CSS features

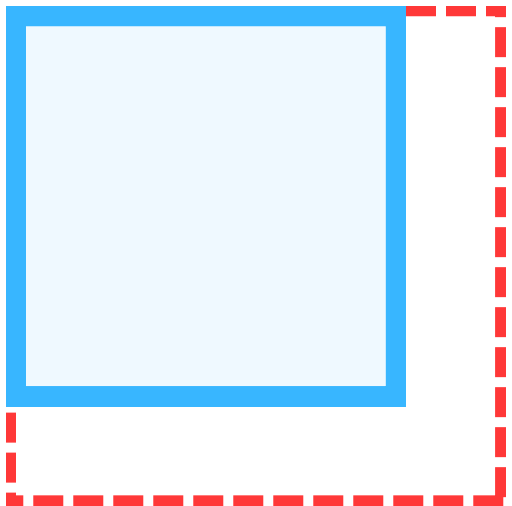
 vercel.app /

box-model-playground.vercel.app

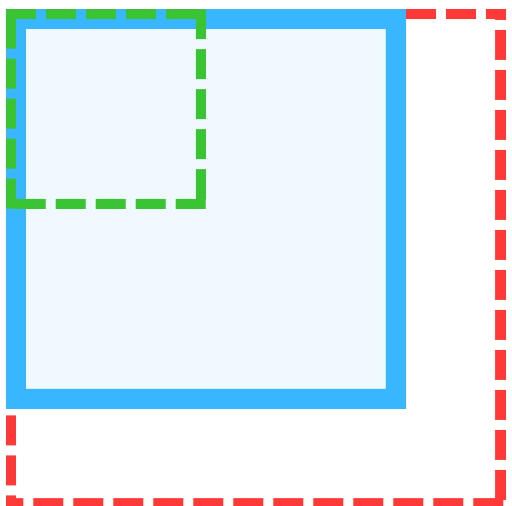
<div> with **fixed** class



<div> with **fixed** and **max** classes



<div> with **fixed**, **max** and **min** classes



What width and height will <div> have?

```
.fixed {  
  width: 1vh; /* 100px */  
  height: 1vh; /* 100px */  
}
```

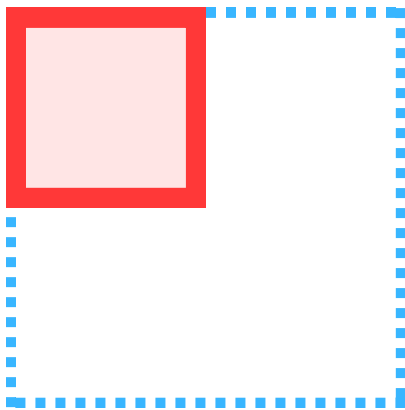
```
.max {  
  max-width: 150px;  
  max-height: 150px;  
}
```

```
.min {  
  min-width: 50px;  
  min-height: 50px;  
}
```

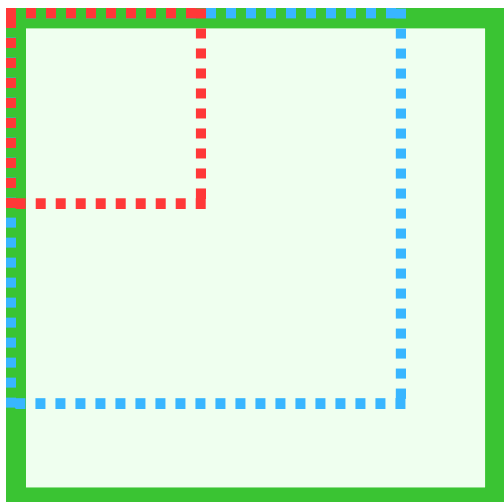

<div> with **fixed** class



<div> with **fixed** and **max** classes



<div> with **fixed**, **max** and **min** classes



What width and height will <div> have?

```
.fixed {  
  width: 1vh; /* 100px */  
  height: 1vh; /* 100px */  
}
```

```
.max {  
  max-width: 50px;  
  max-height: 50px;  
}
```

```
.min {  
  min-width: 150px;  
  min-height: 150px;  
}
```

Note: *max-width* “overrides” *width*, but *min-width* “overrides” *max-width*.

Calc()

- Property values can be determined by **calculation** using the CSS calc() function
- Different units can be combined together

```
#letsCalculate {  
  width: calc(10em - 10px);  
  height: calc(2em - 1rem);  
  min-height: calc(50% + 200px);  
  min-width: calc(2rem + 2px);  
}
```

```
#scaryStuff {  
  width: calc(0 + 20px);  
  height: calc(50% - 8px);  
  max-width: calc(1.25rem/125%);  
}
```

What about * and / ?

[MDN | calc\(\)](#)

Calc(): Are we brave enough?

```
#areWeBraveEnough {  
  --widthA: 100px;  
  --widthB: calc(var(--widthA) / 2);  
  --widthC: calc(var(--widthB) / 2);  
  width: var(--widthC);  
}
```

Checkpoint reached

Flow

Text-decoration

Text-align

Cursor

Max&Min

Calc

The box model

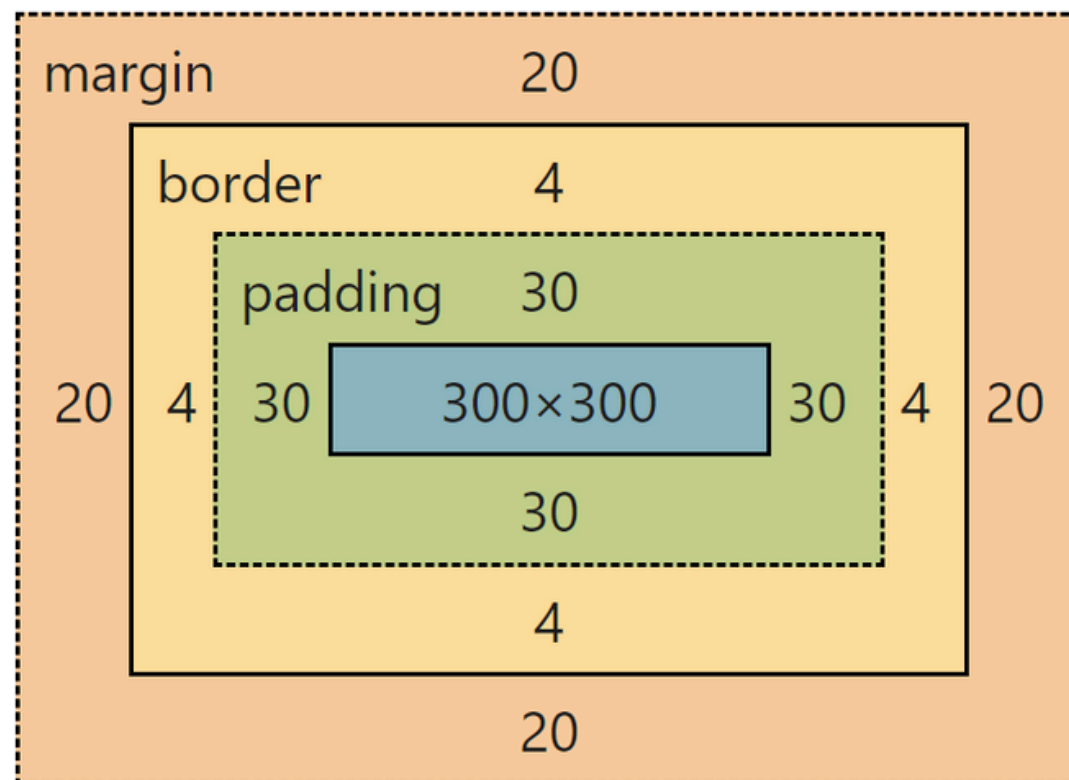
Overview

padding

border

box-sizing

margin



Content-box

This is the **core area** where the element's content resides, such as **text**, **images**, or **videos**.

Padding

Space between the **border** and the **content**

Border

The **edge** around the box

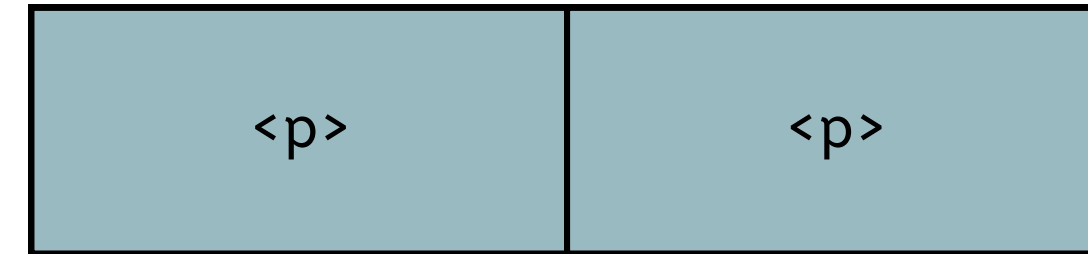
Margin

Transparent area around the box that **separates** it from other **elements**

[MDN | Box Model](#)

let's play!

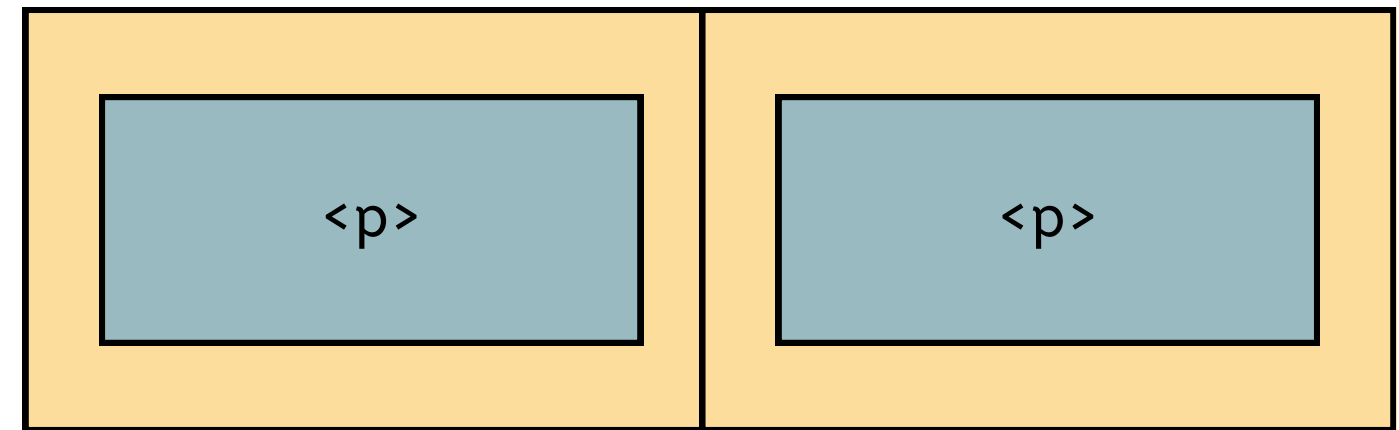
```
p {  
  display: inline-block  
}
```



if I want to place a limit on my
elements what should I do?

let's play!

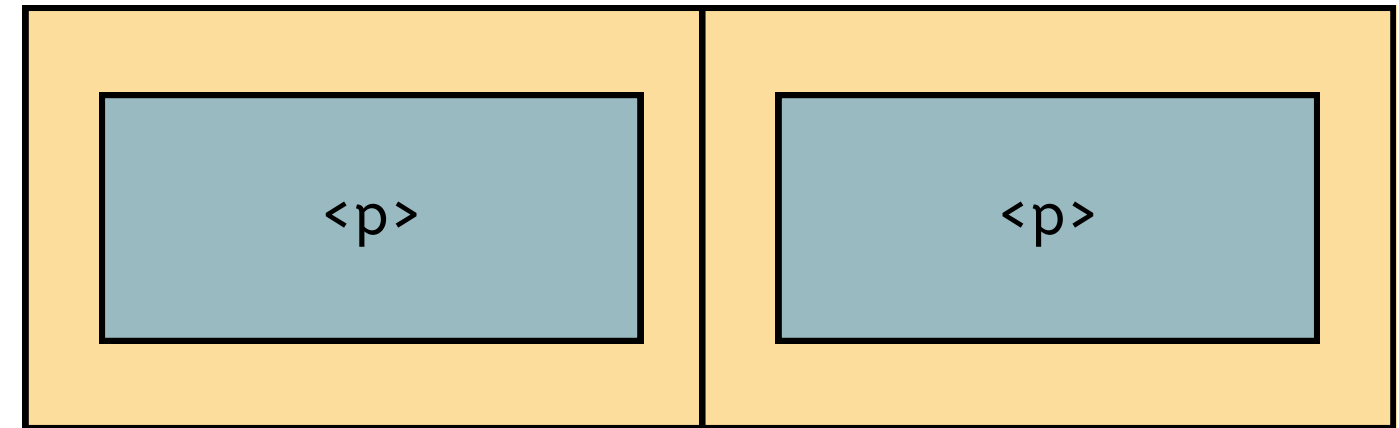
```
p {  
  display: inline-block;  
  border: - - - - -;  
}
```



BORDER

let's play!

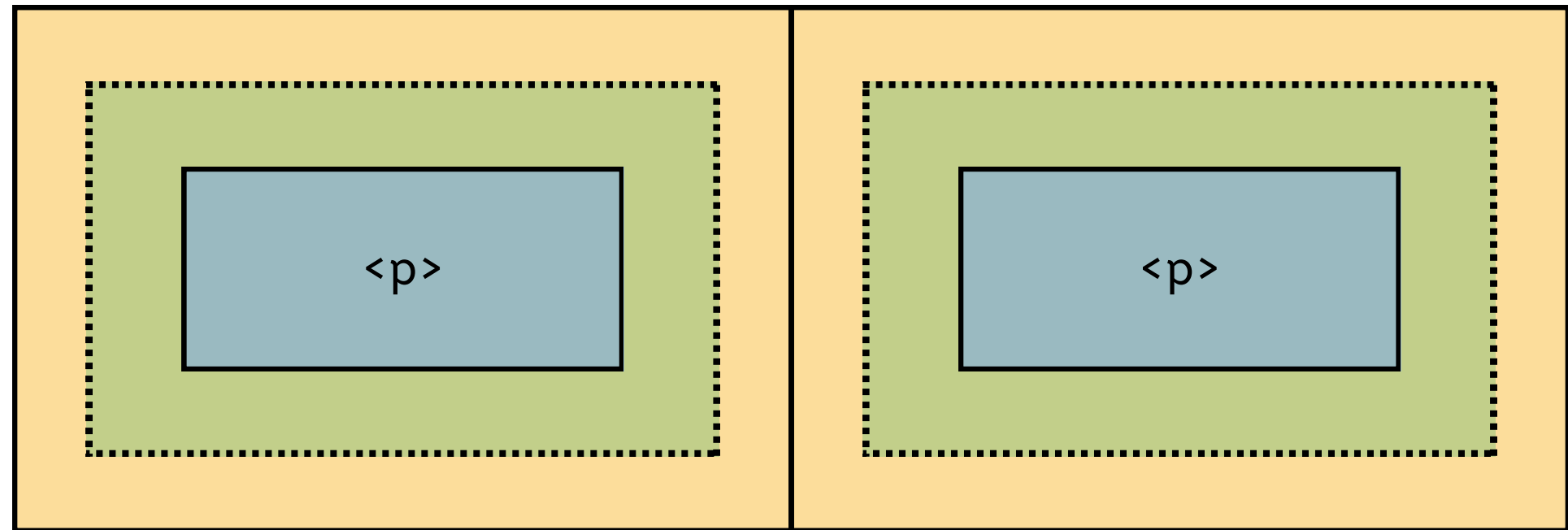
```
p {  
  display: inline-block;  
  border: - - - - -;  
}
```



if I want to place a distance inside
my elements what should I do?

let's play!

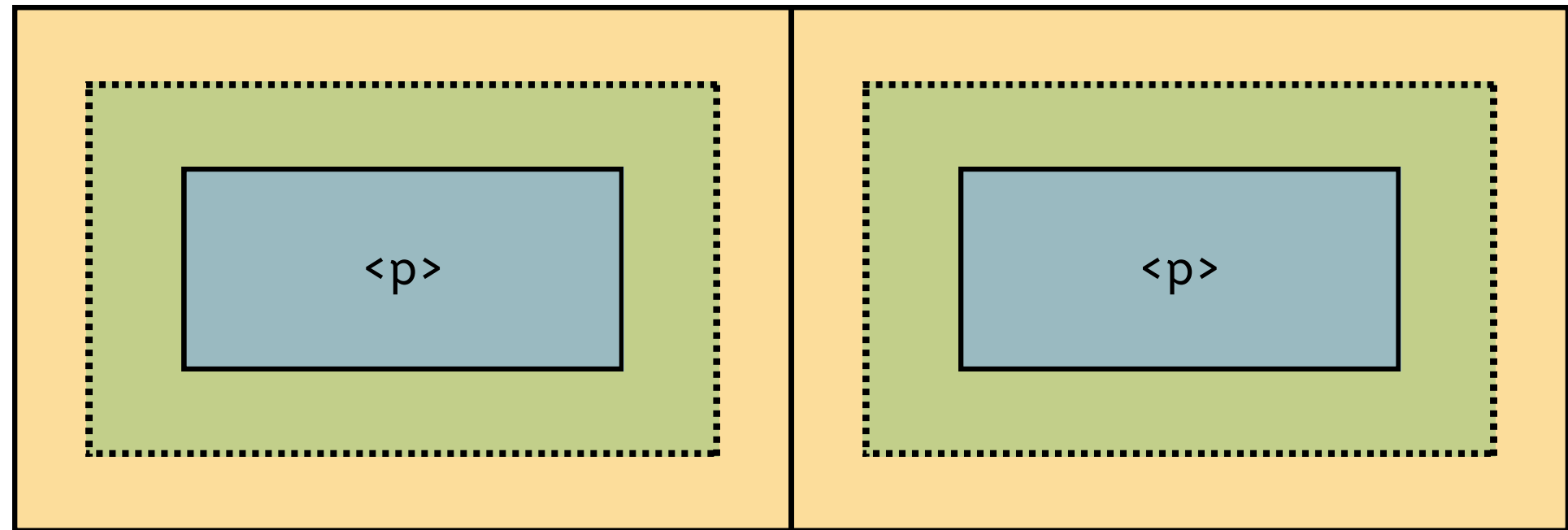
```
p {  
  display: inline-block;  
  border: - - - - -;  
  padding: - - - - -;  
}
```



PADDING

let's play!

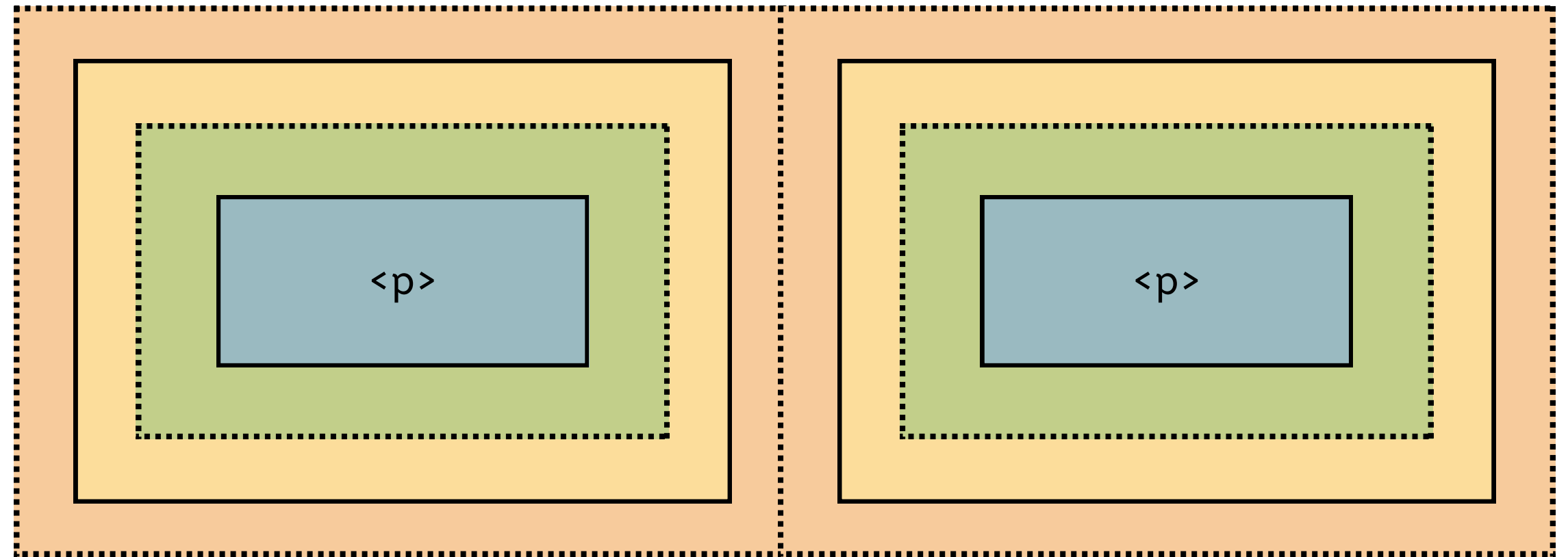
```
p {  
  display: inline-block;  
  border: - - - - -;  
  padding: - - - - -;  
}
```



if I want to place a distance outside
the elements what should I do?

let's play!

```
p {  
  display: inline-block;  
  border: - - - - -;  
  padding: - - - - -;  
  margin: - - - - -;  
}
```



MARGIN

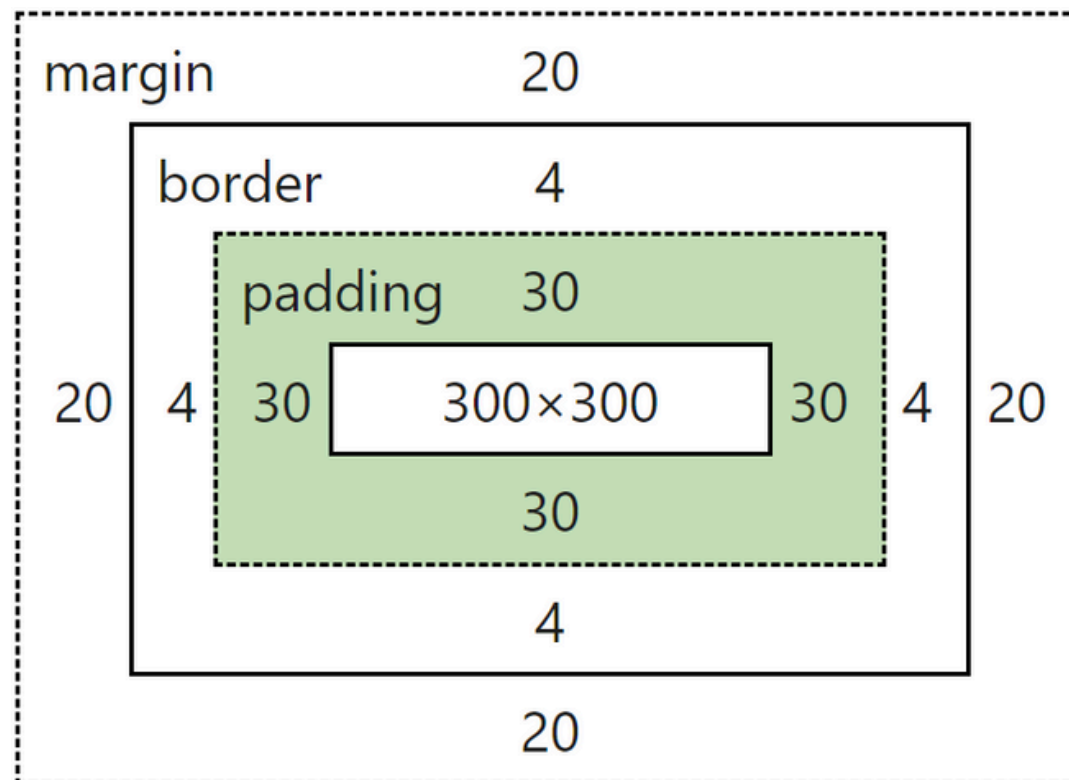
overview

Padding

border

box-sizing

margin



Properties

`padding-top: 10px;`

`padding-right: 15%;`

`padding-bottom: 10em;`

`padding-left: 15vw;`

Shorthand property

`padding: 10px; /* trbl */`

`padding: 10px 15%; /* tb-rl */`

`padding: 10px 15% 10em; /* t-rl-b */`

`padding: 10px 15% 10em 15vw; /* t-r-b-l */`

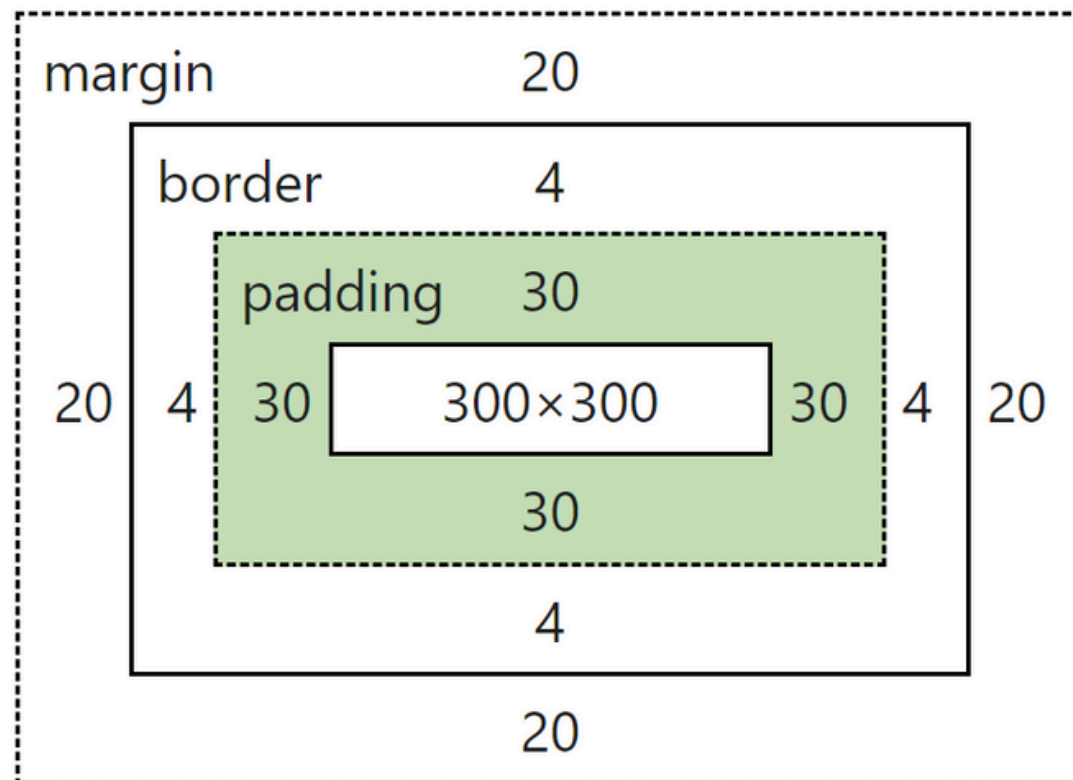
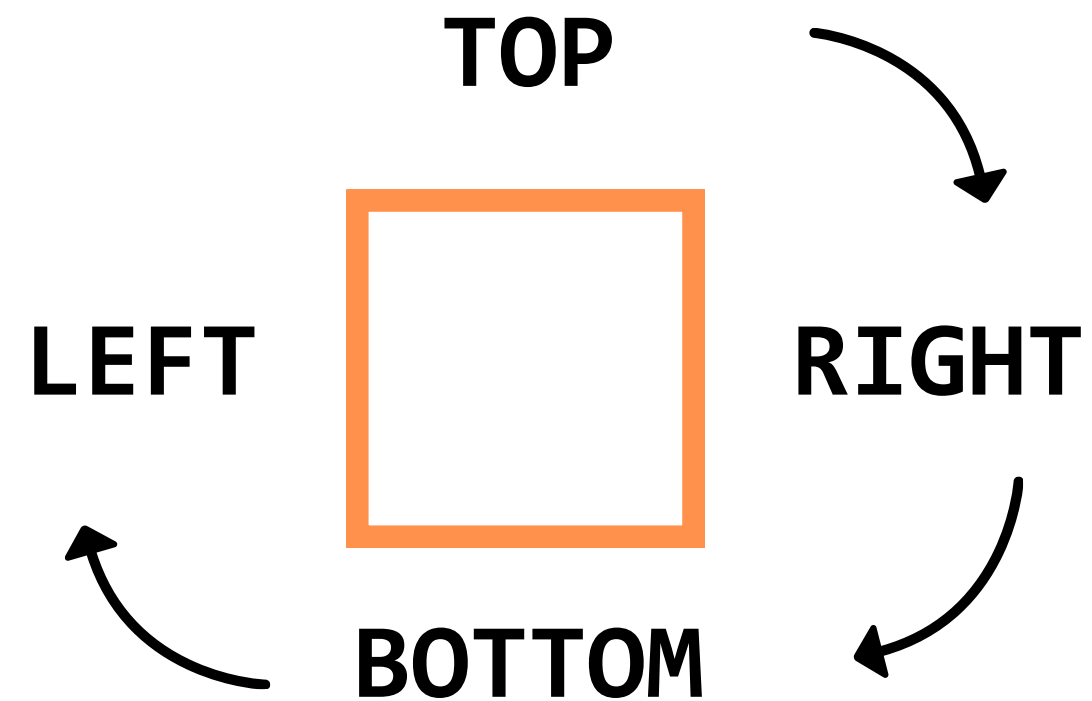
overview

Padding

border

box-sizing

margin



Shorthand property

padding: 10px; /* trbl */

padding: 10px 15%; /* tb-rl */

padding: 10px 15% 10em; /* t-rl-b */

padding: 10px 15% 10em 15vw; /* t-r-b-l */

[MDN | Padding](#)

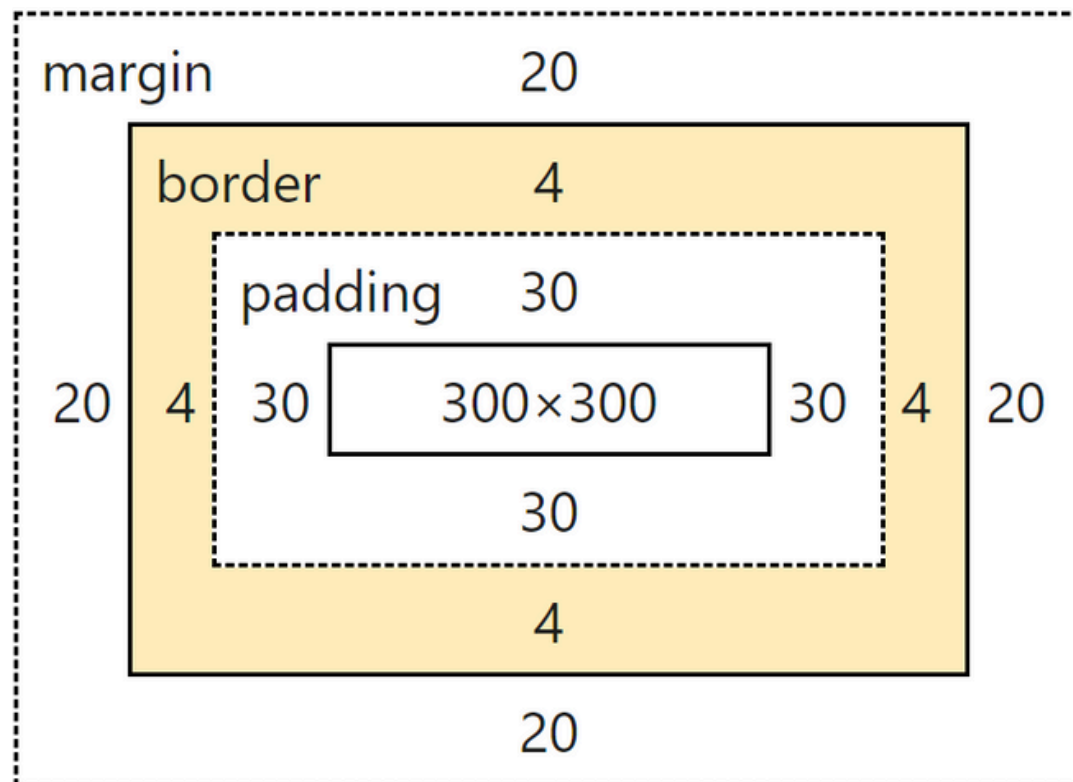
overview

padding

Border

box-sizing

margin



Properties

border-width: 10px;

border-style: solid;

border-color: #ff9900;

border-radius: 6px;

Specific edge properties

border-top-width: 15px;

border-right-style: dotted;

border-bottom-color: #0099ff;

Specific corner property

border-top-left-radius: 50%;

Shorthand property

border: 10px solid #ff9900;

border-top: 2px dotted #0050a0;

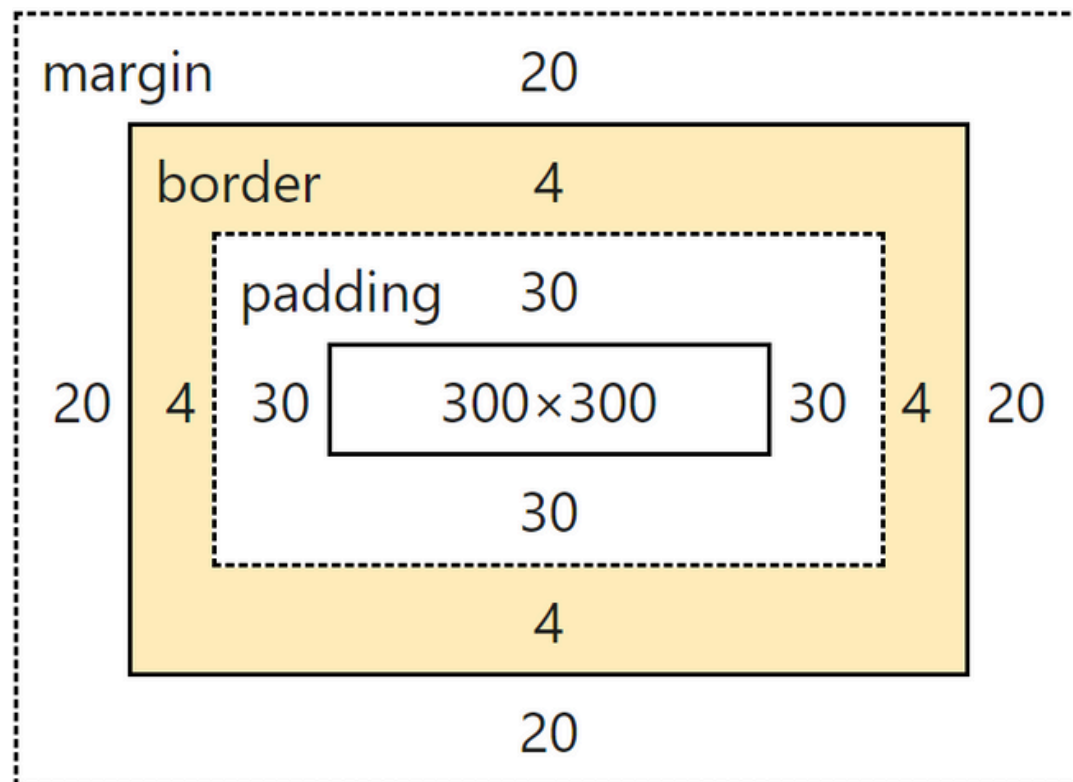
overview

padding

Border

box-sizing

margin



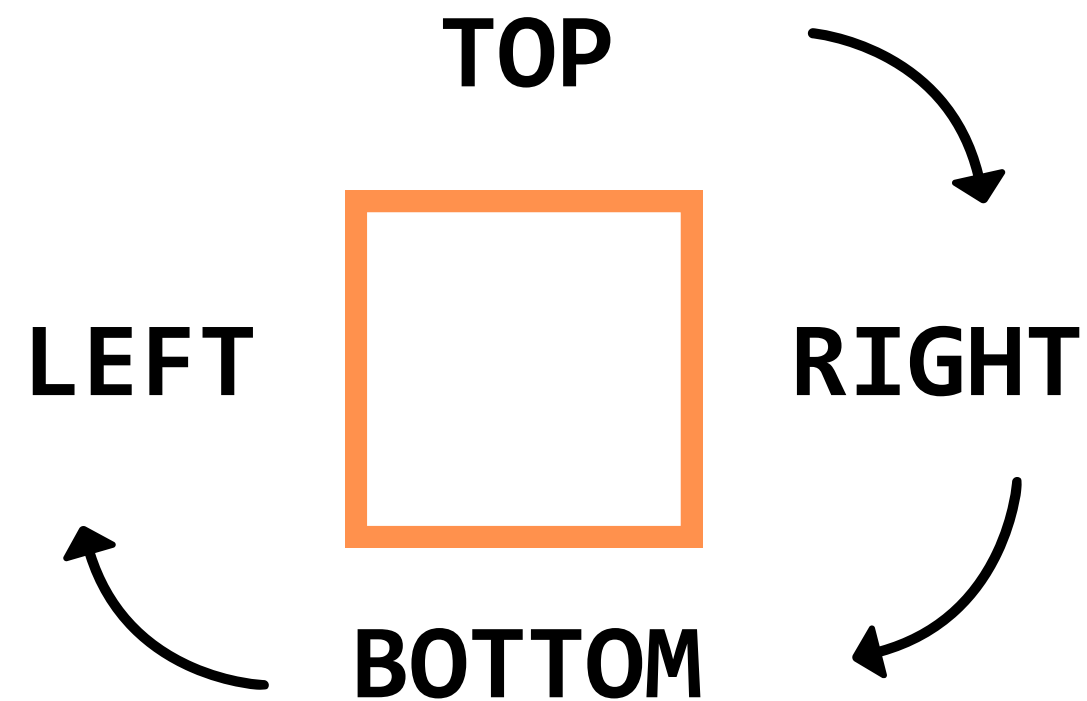
Properties

border-width: 10px 15px 5px 20px;

border-style: solid dotted;

border-color: #ff9900 #ffffff #0099ff;

border-radius: 6px 2vw 30px 3em;



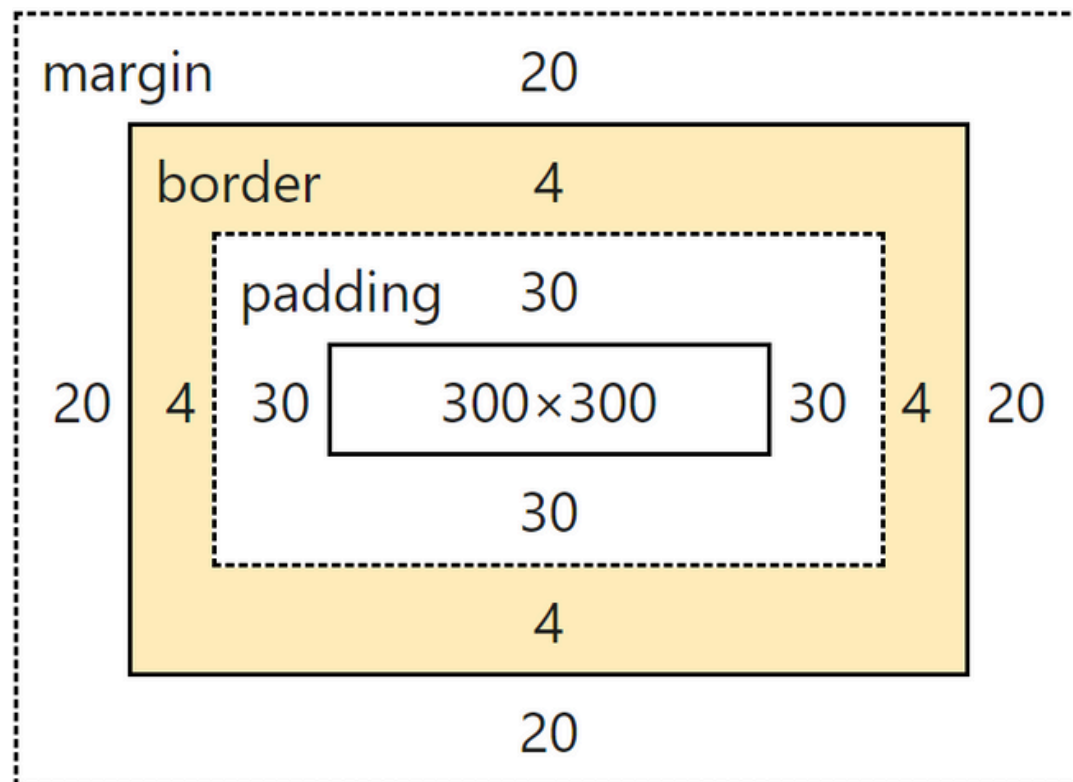
overview

padding

Border

box-sizing

margin

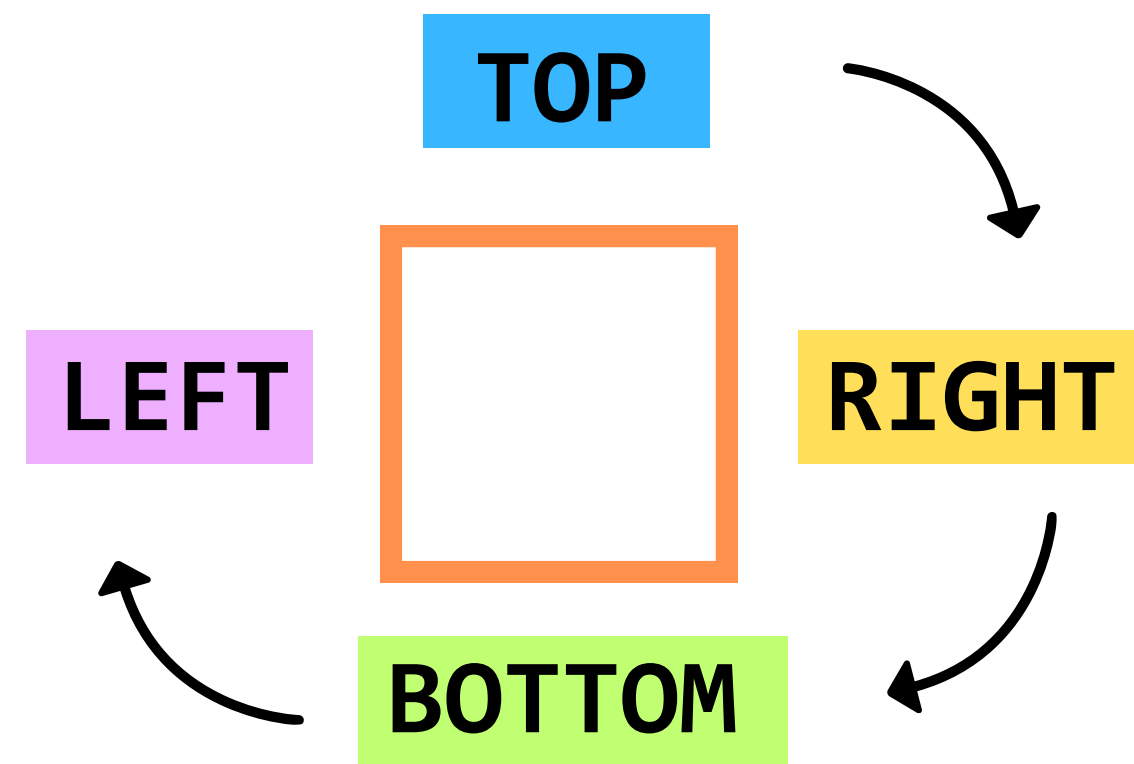


Edges properties

border-width: 10px 15px 5px 20px;

border-style: solid dotted;

border-color: #ff9900 #ffffff #0099ff;



[MDN | Border](#)

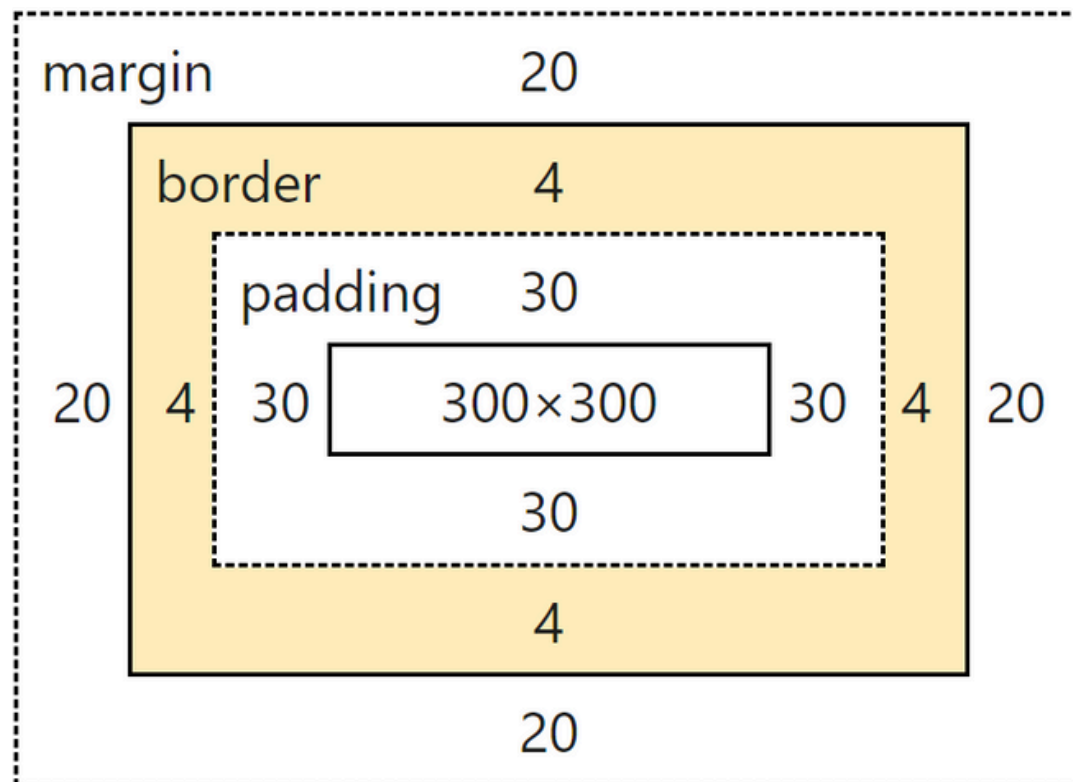
overview

padding

Border

box-sizing

margin



Corners properties

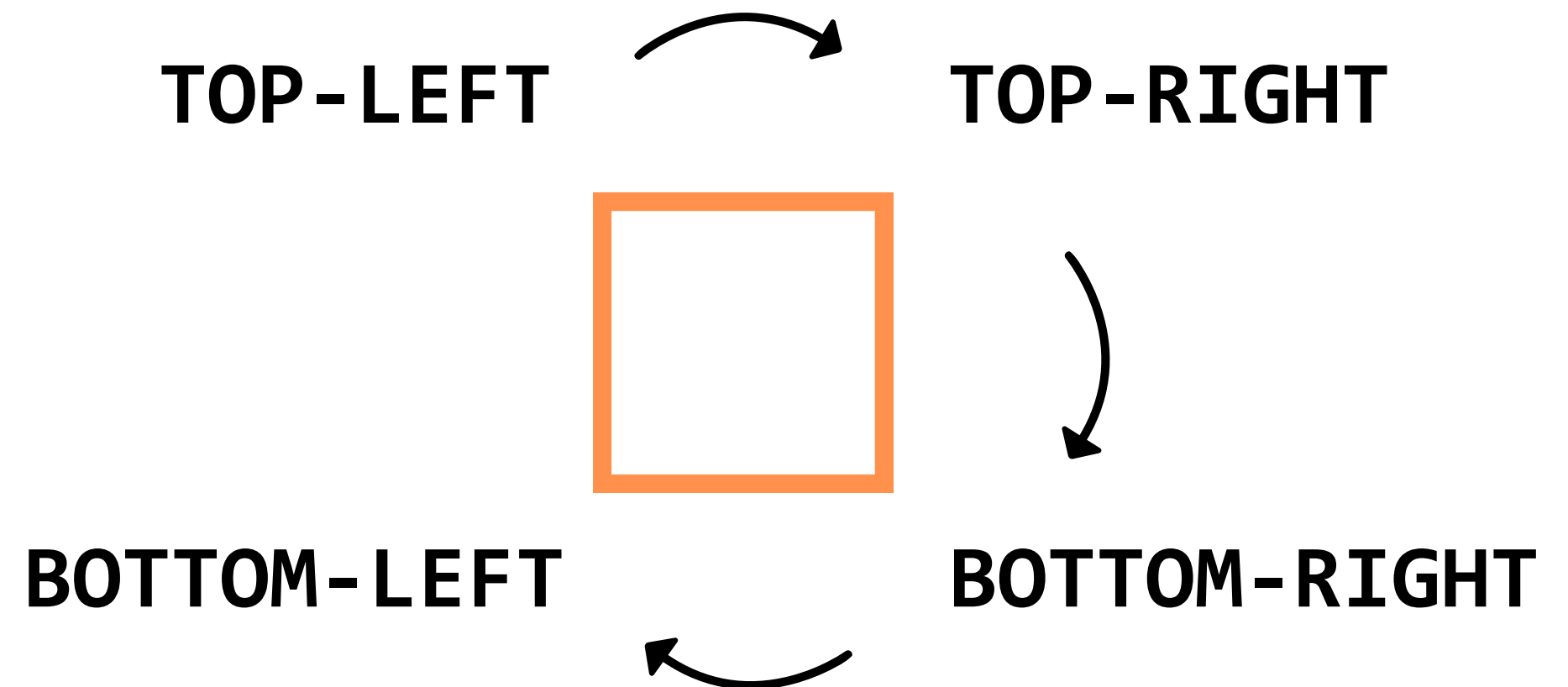
`border-top-left-radius: 6px;`

`border-top-right-radius: 2vw;`

`border-bottom-right-radius: 30px;`

`border-bottom-left-radius: 3em;`

`border-radius: 6px 2vw 30px 3em;`



[MDN | Border-radius](#)

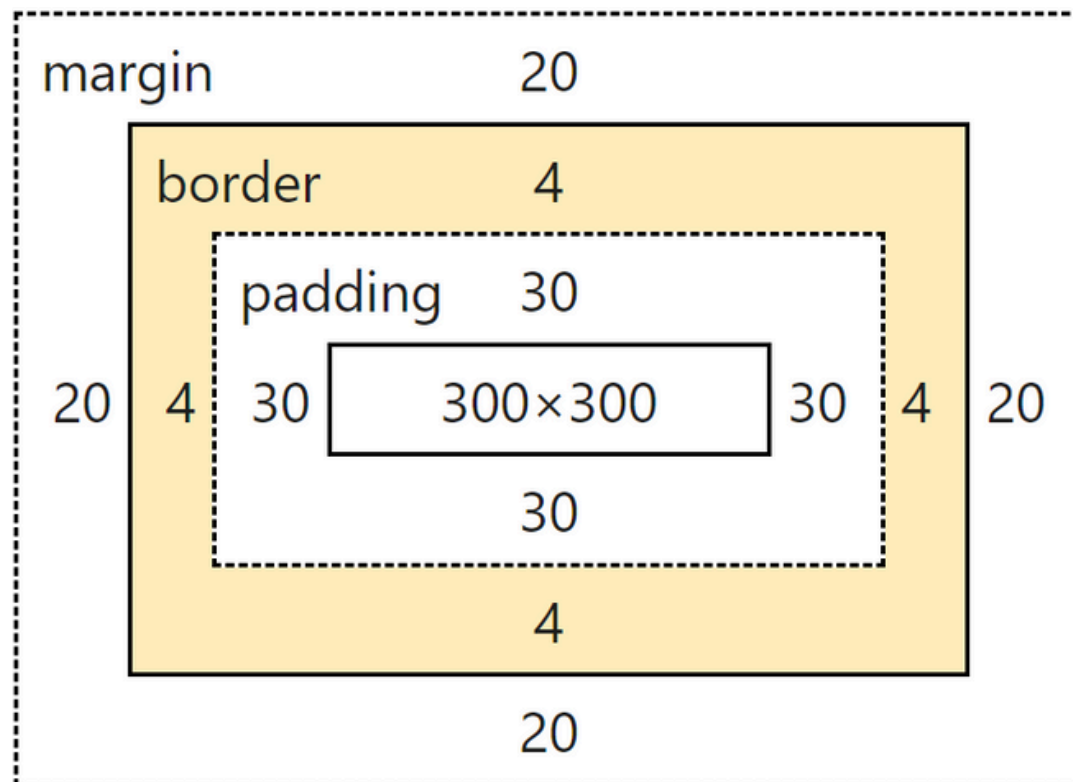
overview
padding
border

Box-sizing

margin

The box-sizing property

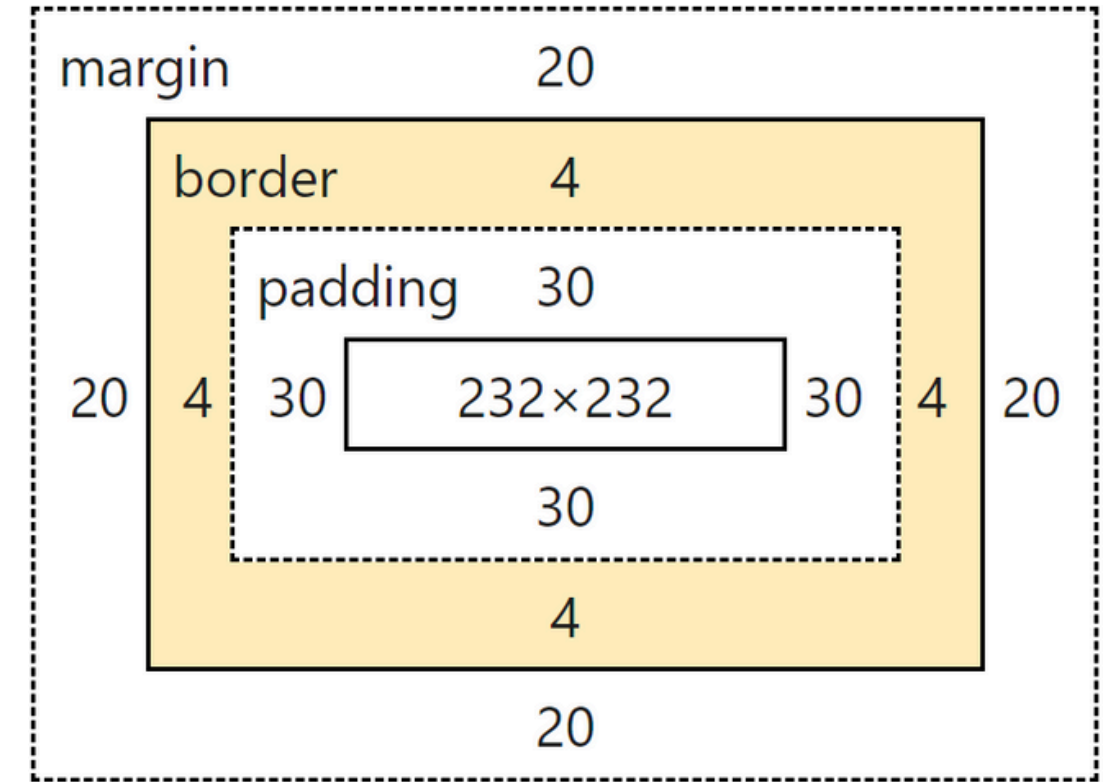
Defines whether the **width** and **height** of an element should **include padding** and **borders** or **not**



◀ **box-sizing: content-box;**

height: 300px;
width: 300px;
padding: 30px;
border: 4px solid black;

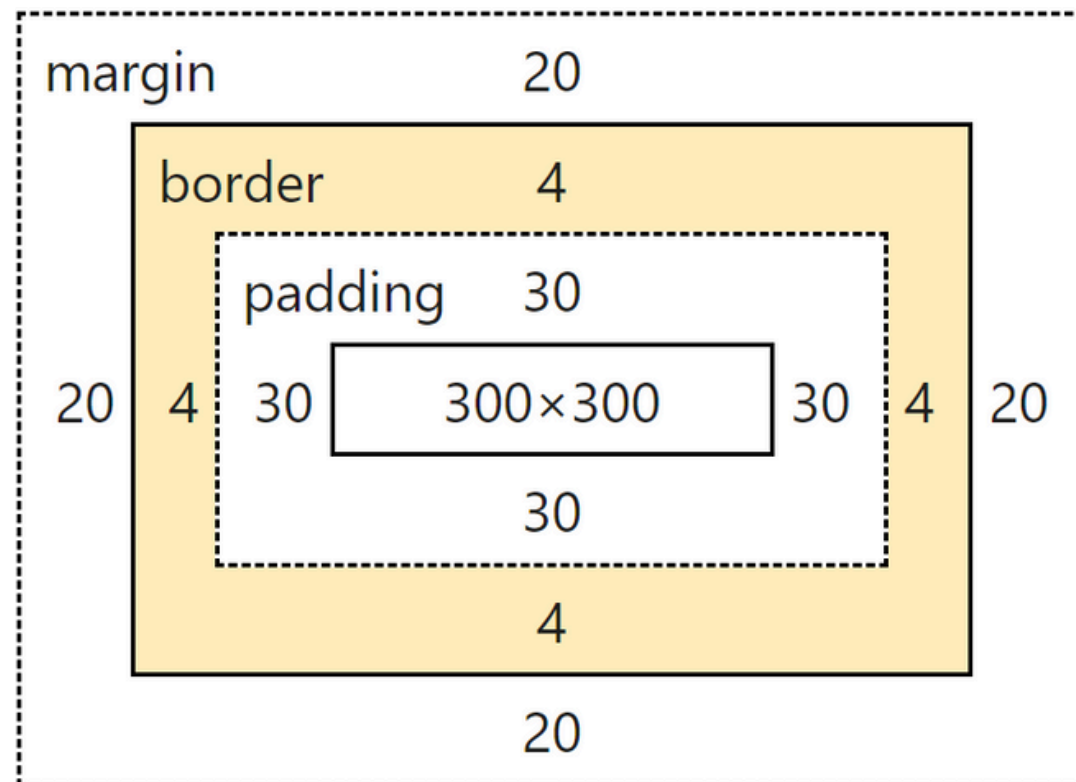
box-sizing: border-box; ▶



overview
padding
border

Box-sizing

margin



Setting box-sizing

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

OLD METHOD

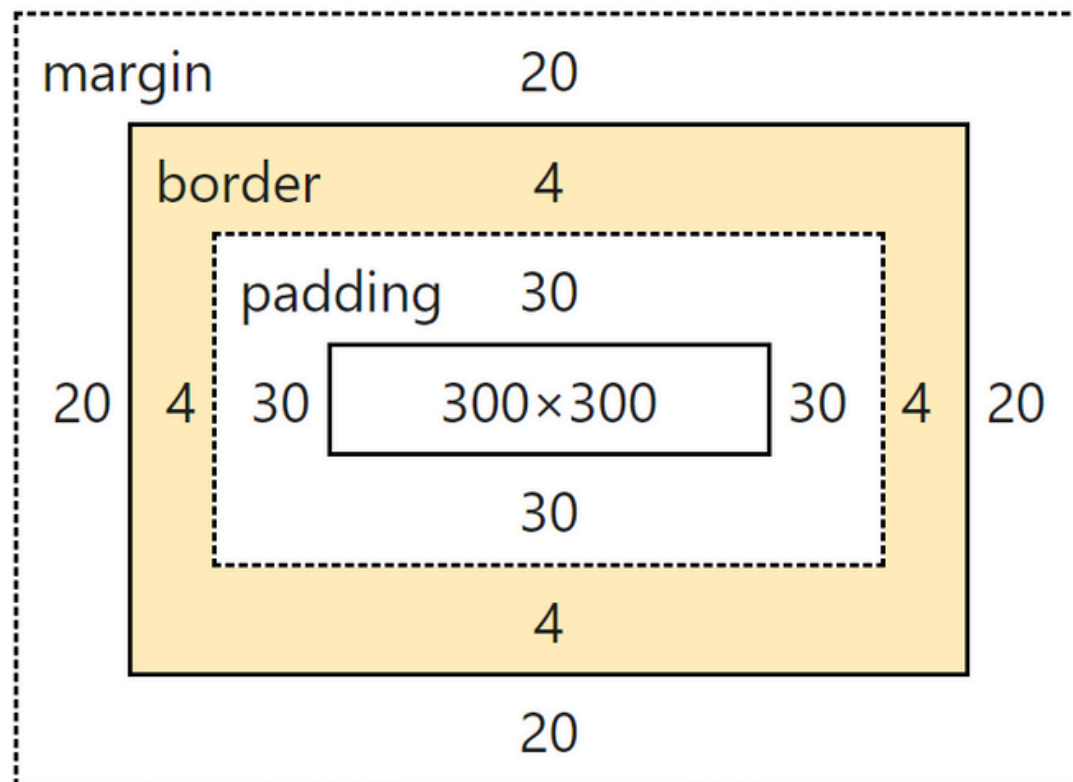
overview

padding

border

Box-sizing

margin



Universal Box Sizing

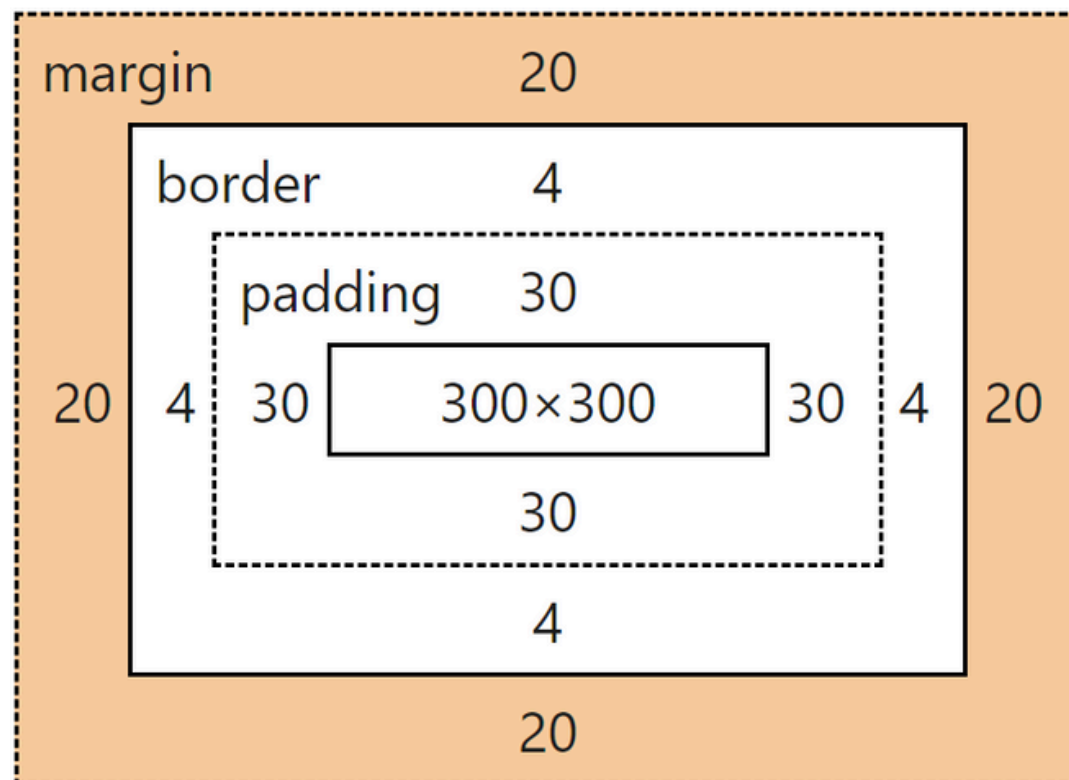
```
*, *::before, *::after {  
    box-sizing: border-box;  
}
```

Universal Box Sizing with Inheritance

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

[MDN | Box-sizing](https://developer.mozilla.org/en-US/docs/Web/CSS/box-sizing)

overview
padding
border
box-sizing
Margin



Properties

```
margin-top: 10px;  
margin-right: 3em;  
margin-bottom: 5%;  
margin-left: 10vw;
```

Shorthand property

```
margin: 10px; /* trbl */  
margin: 10px 15%; /* tb-rl */  
margin: 10px 15% 10em; /* t-rl-b */  
margin: 10px 15% 10em 15vw; /* t-r-b-l */
```

auto margins

```
margin: 10px auto; /* align center */  
margin: 10px 10px 10px auto; /* align right */
```

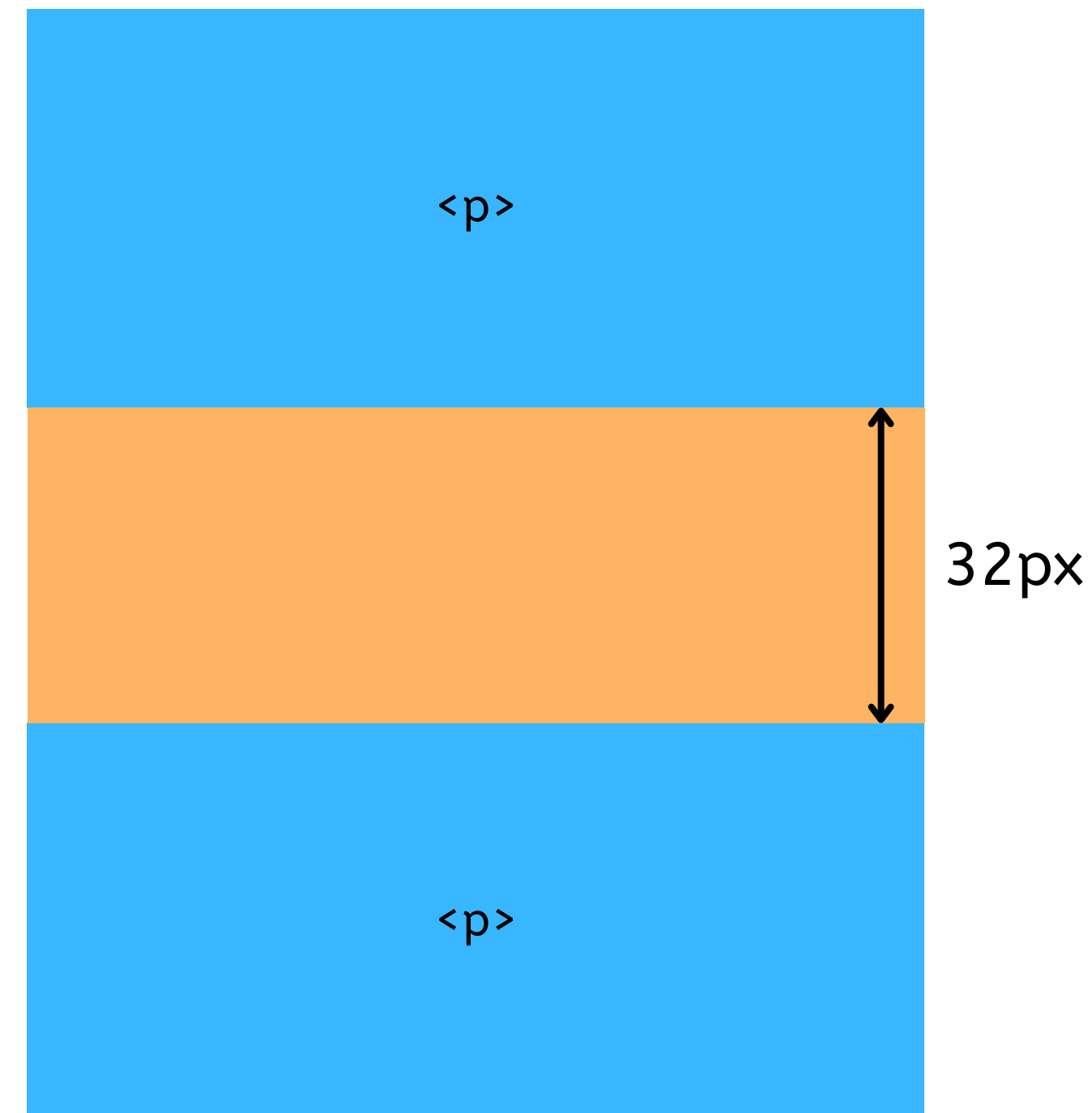
let's play!
(again)

```
html

<p>Paragraph One</p>
<p>Paragraph Two</p>

css

p {
  margin-top: 32px;
  margin-bottom: 32px;
}
```

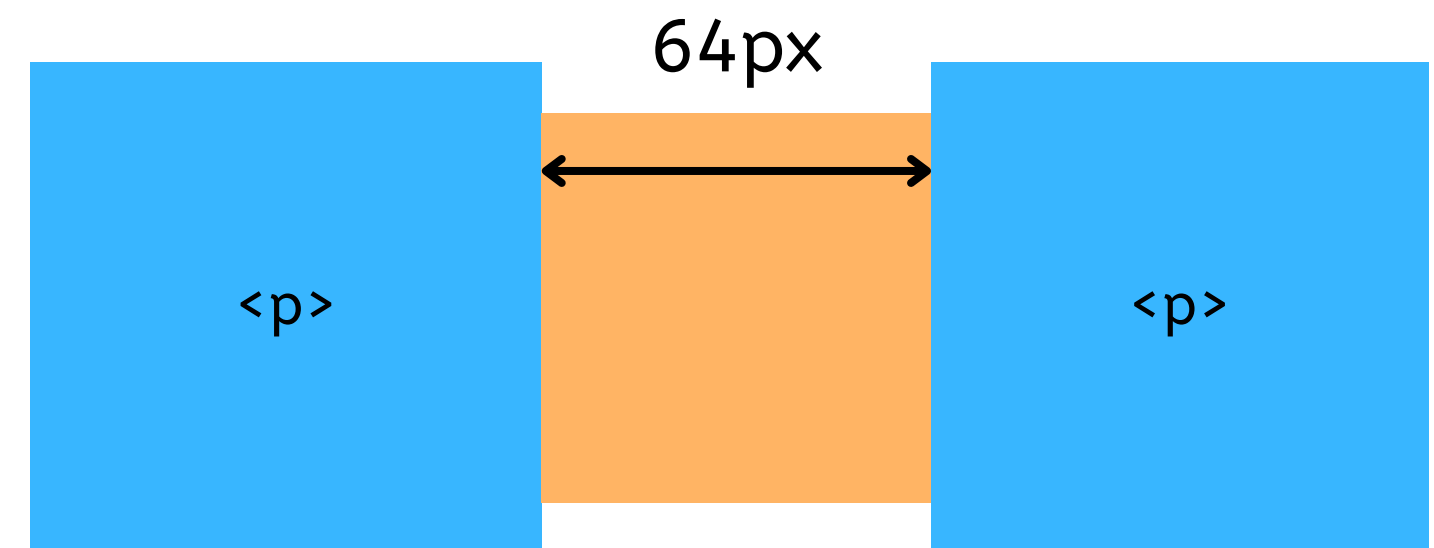



```
html

<p>Paragraph One</p>
<p>Paragraph Two</p>

css

p {
  margin-left: 32px;
  margin-right: 32px;
}
```

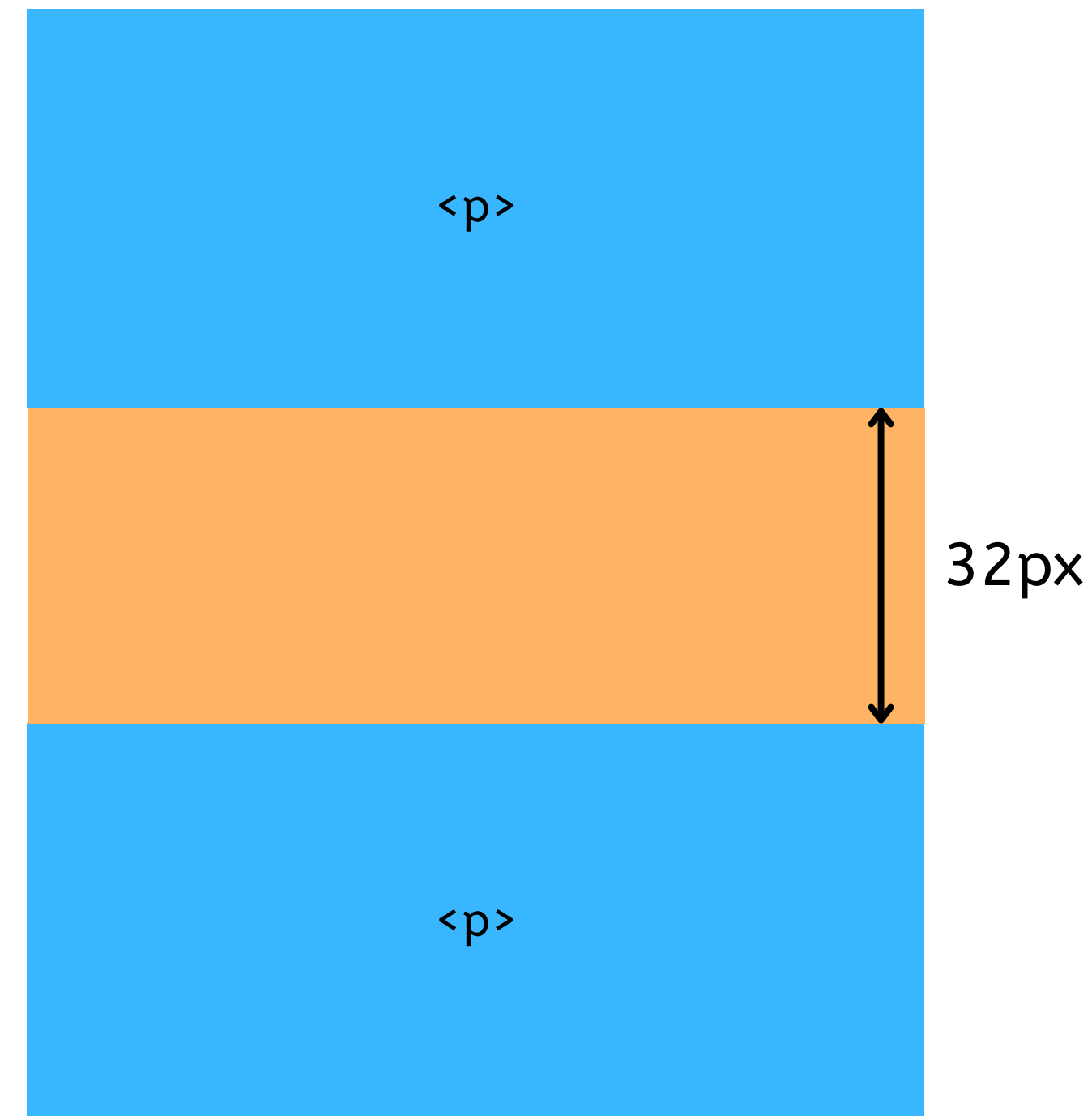


```
html

<p>Paragraph One</p>
<p>Paragraph Two</p>

css

p {
  margin-top: 24px;
  margin-bottom: 32px;
}
```

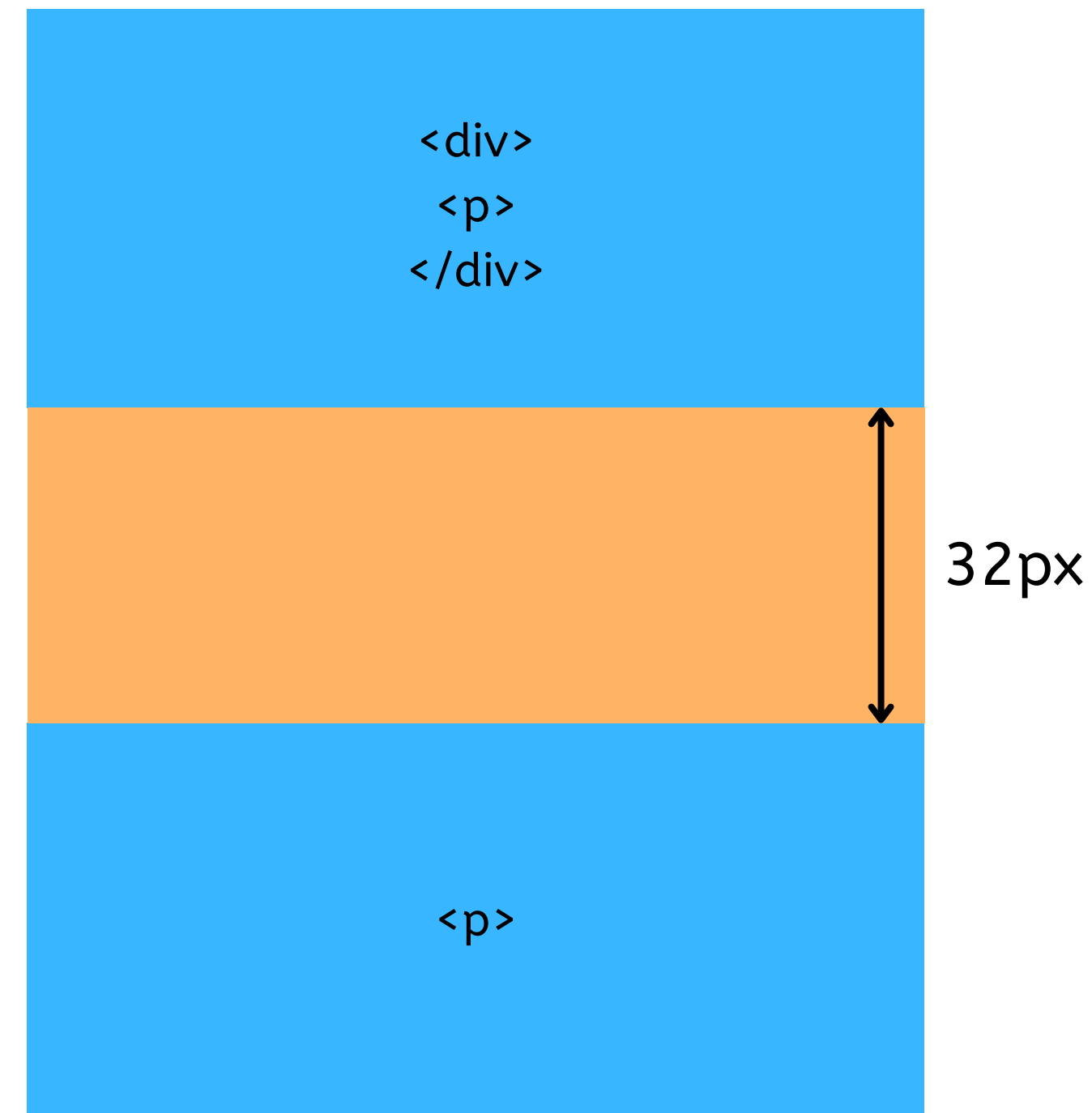


```
html

<div>
  <p>Paragraph One</p>
</div>
<p>Paragraph Two</p>

css

p {
  margin-top: 32px;
  margin-bottom: 32px;
}
```

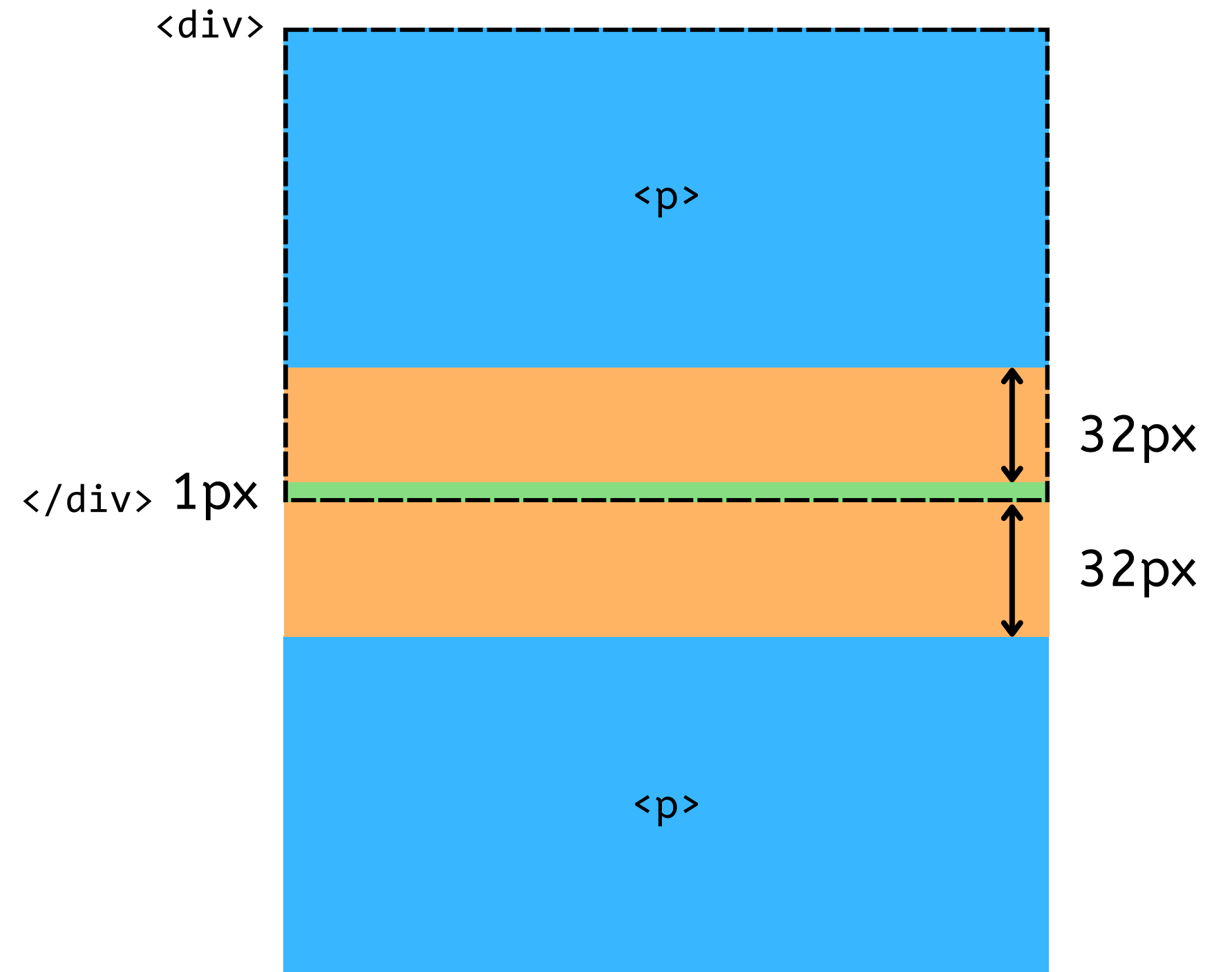


html

```
<div>
  <p>Paragraph One</p>
</div>
<p>Paragraph Two</p>
```

css

```
p {
  margin-top: 32px;
  margin-bottom: 32px;
}
div {
  padding-bottom: 1px;
}
```

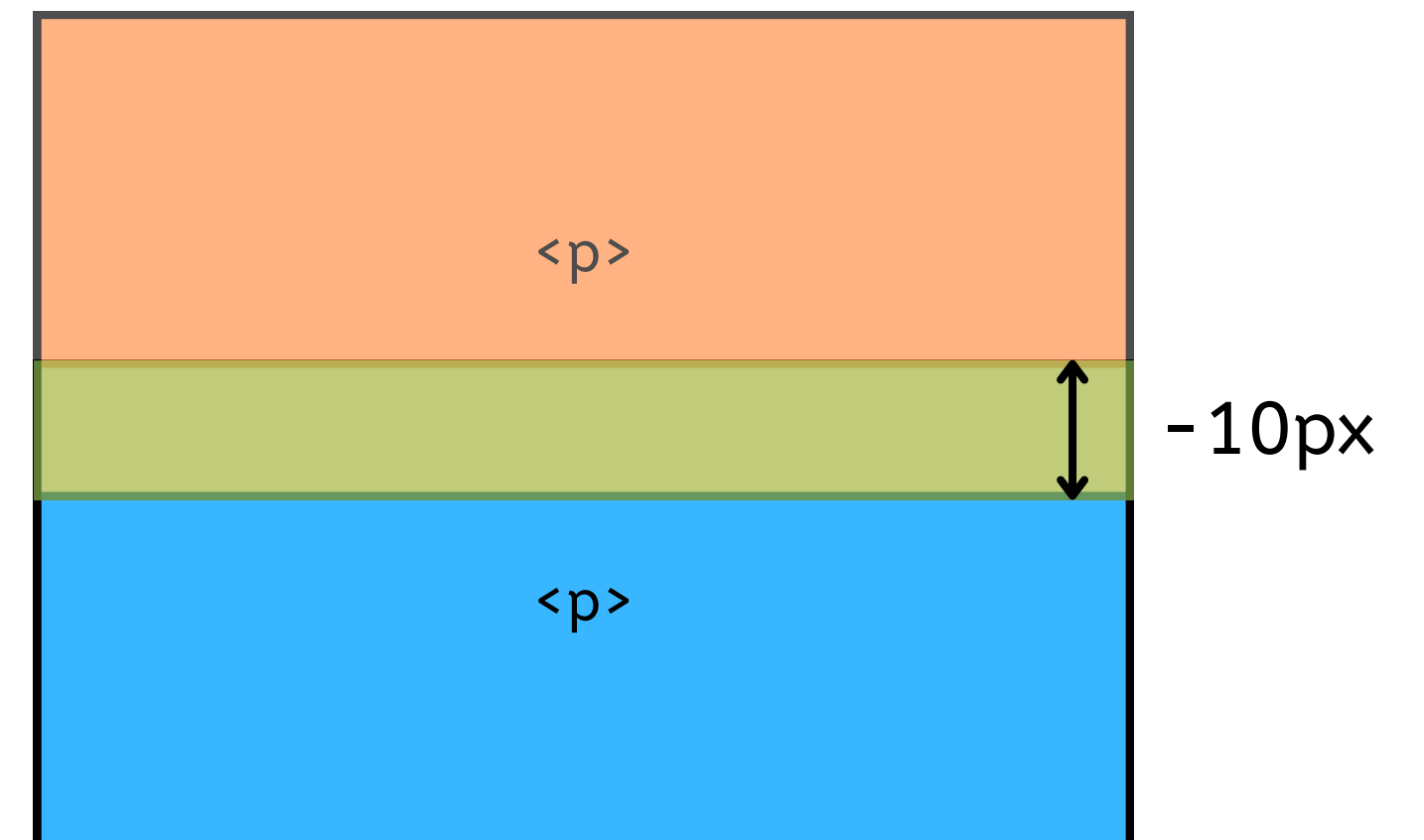


```
html

<p>Paragraph One</p>
<p>Paragraph Two</p>

css

p {
  margin-top: -8px;
  margin-bottom: -10px;
}
```

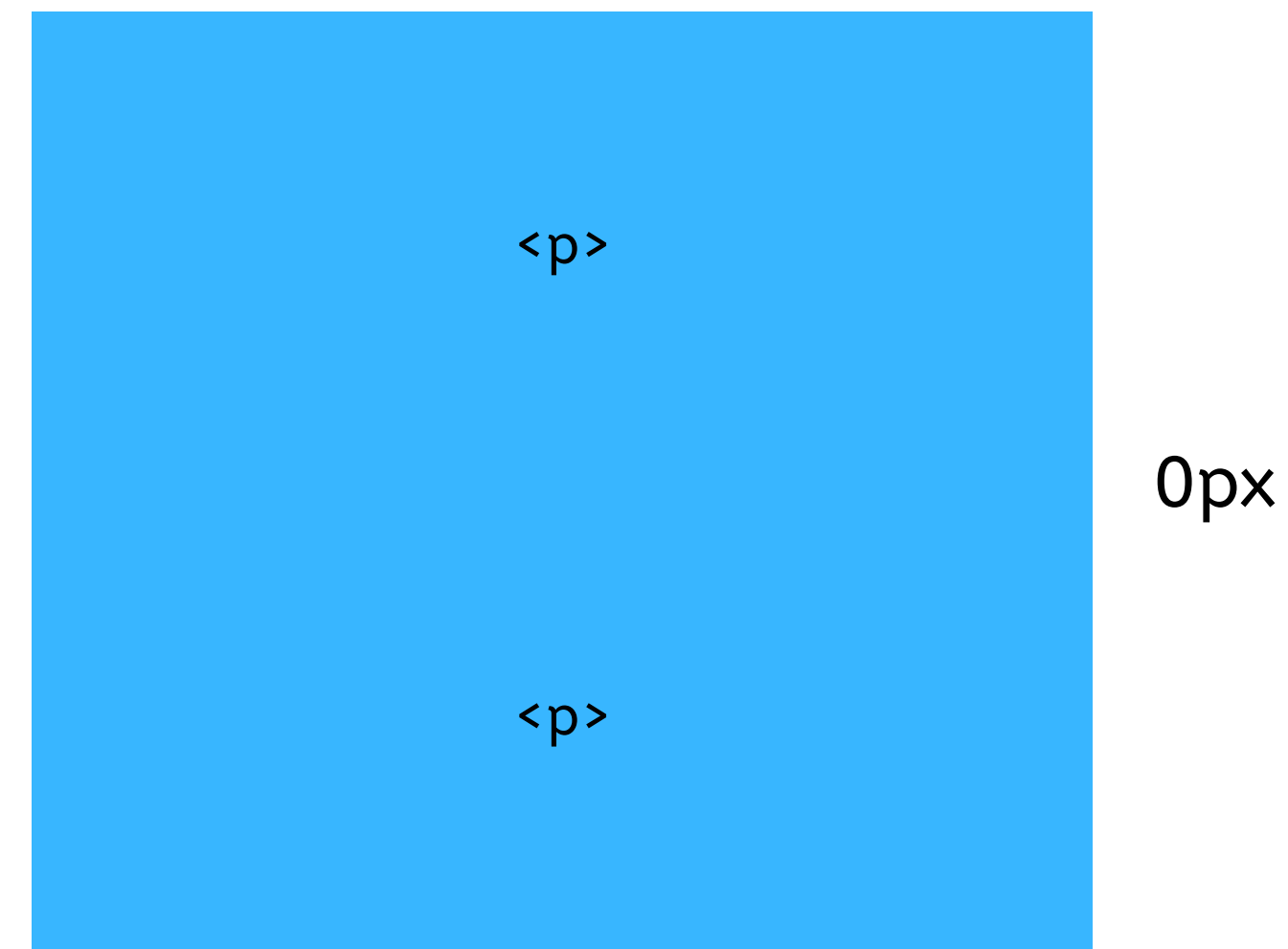


```
html

<p>Paragraph One</p>
<p>Paragraph Two</p>

css

p {
  margin-top: 10px;
  margin-bottom: -10px;
}
```



Margin collapse: Let's wrap up

- Margin is meant to **increase the distance between siblings**.
- Margin will always try and increase distance between siblings, even if it means **transferring margin to the parent** element.
- Margins **only collapse when they're touching**. If there's any sort of gap or barrier between margins, they won't collapse.
- Between two margins with same sign, the **most significant** one gets chosen over the other.
- Opposite signs cancel each other.
- Adding a padding or border to the parent breaks collapsing
- Only vertical margins collapse!
- Margins of elements taken out-of-flow will never collapse (spoiler)

What we have learned (hopefully)

Flow

Text-decoration

Text-align

Cursor

Max&Min

Calc

Box Model

Padding

Border

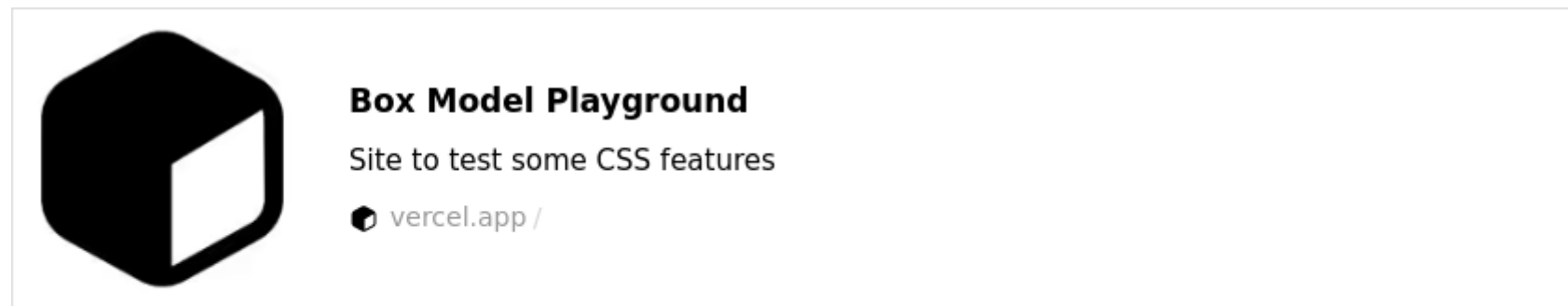
Box-Sizing

Margin

Margin-collapse

Now what?

We made a playground!



box-model-playground.vercel.app