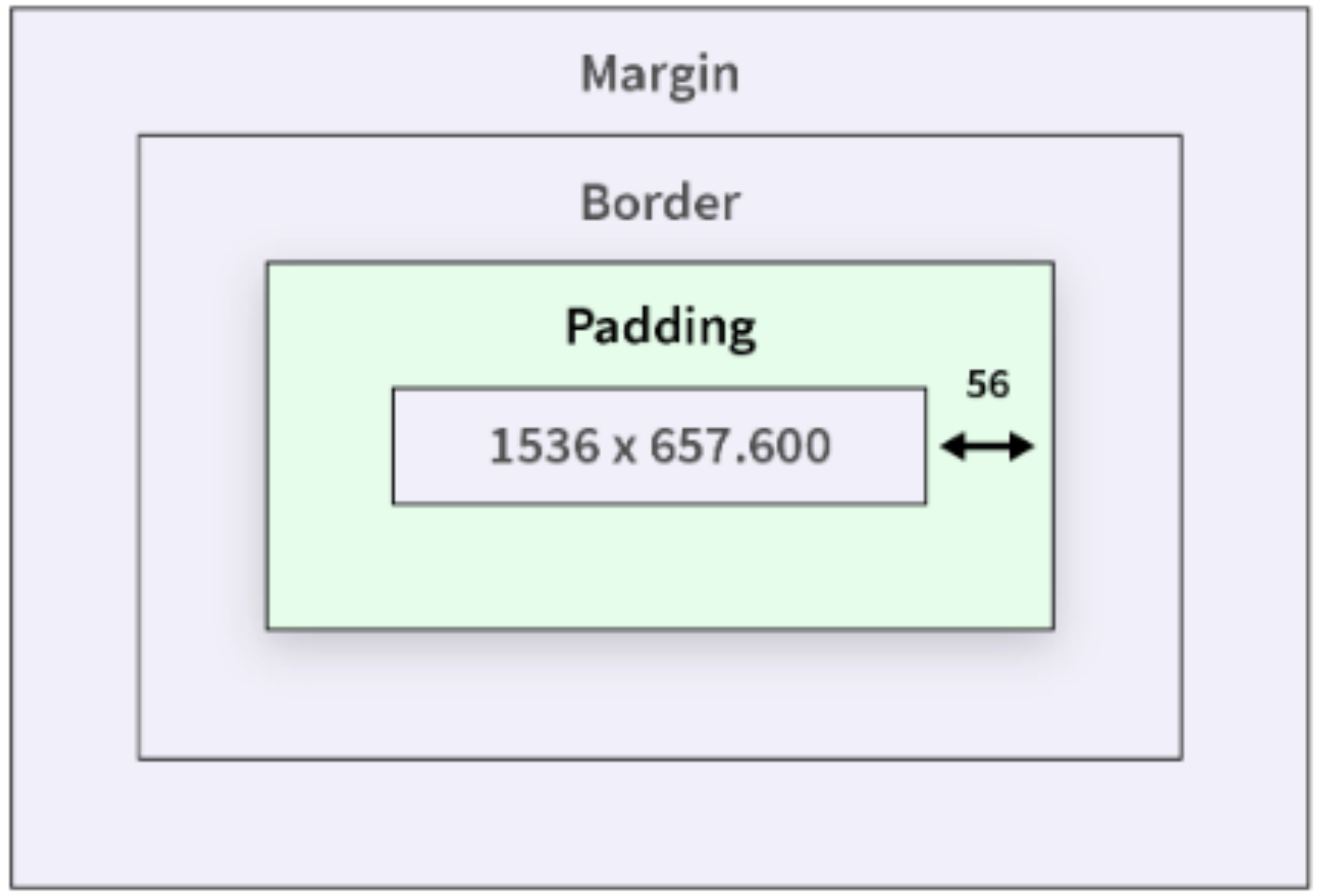
CSS Box Model & Selectors

Abhi Chakraborty (27th June 2022)

CSS Padding

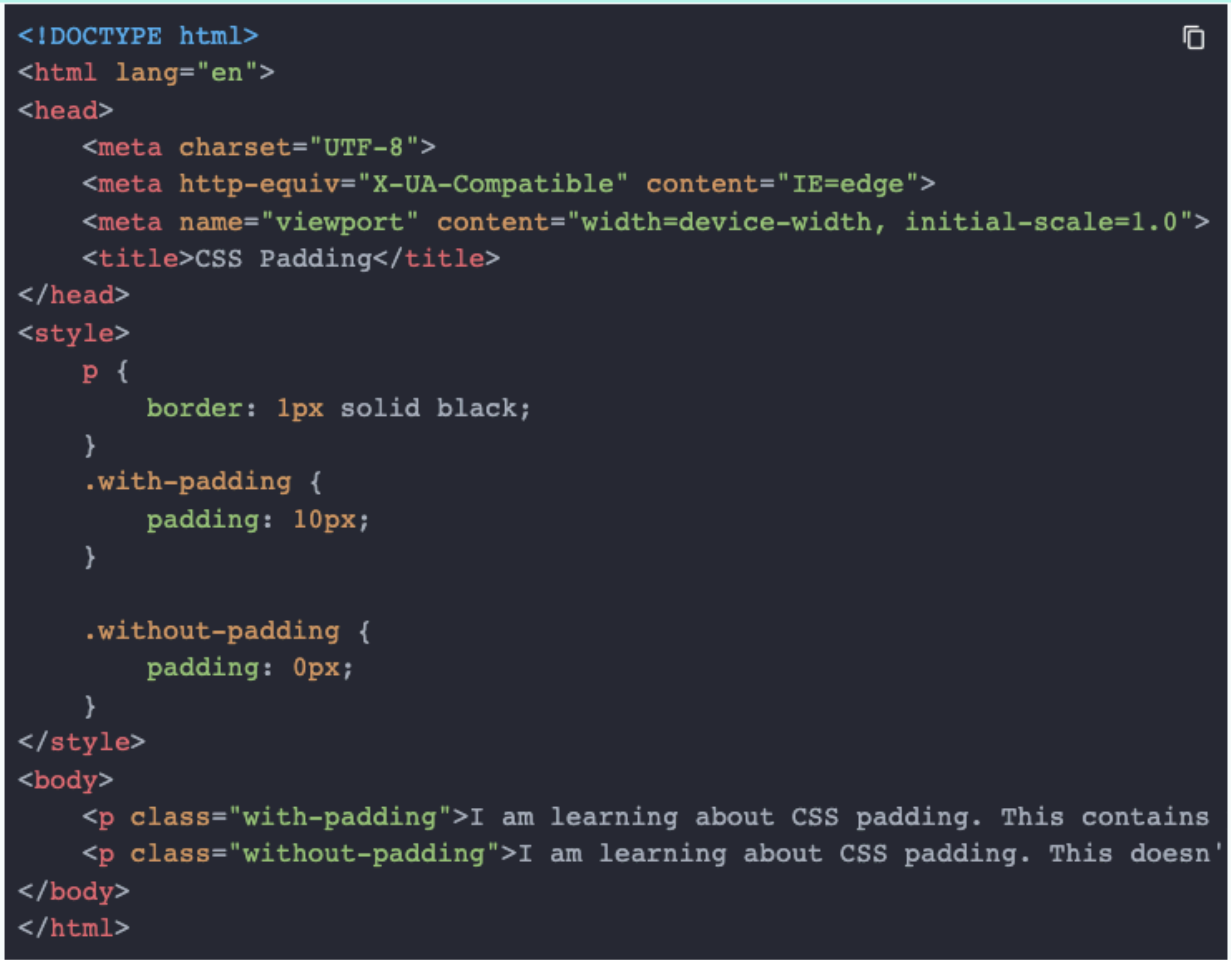
- CSS padding property is used to provide the space between the element's border and its content. It is different from the CSS margin in the way that it is affected by the background color of that element on which padding is set i.e if you set an element's background color it will be visible through the padding area. Margin creates extra space outside the element whereas padding creates extra space within the element



What is CSS Padding?

- CSS padding property is used to control the amount of space inside the HTML element between its content and border. You know about the margin that adds up the space outside an element's border but for padding the space is added inside the element's border. You can change the top, right, bottom, and left padding properties independently and you can also change all the padding properties at once using the shorthand property that you'll study later in this article
- Syntax : padding: size in px or percentage or inherit;
- The padding properties can be specified using the following values
- **length** - specifies padding in rem, em, px, etc
- **%** - specifies padding in the percentage of the width of the containing element
- **inherit** - specifies the padding should be inherited from the parent element.

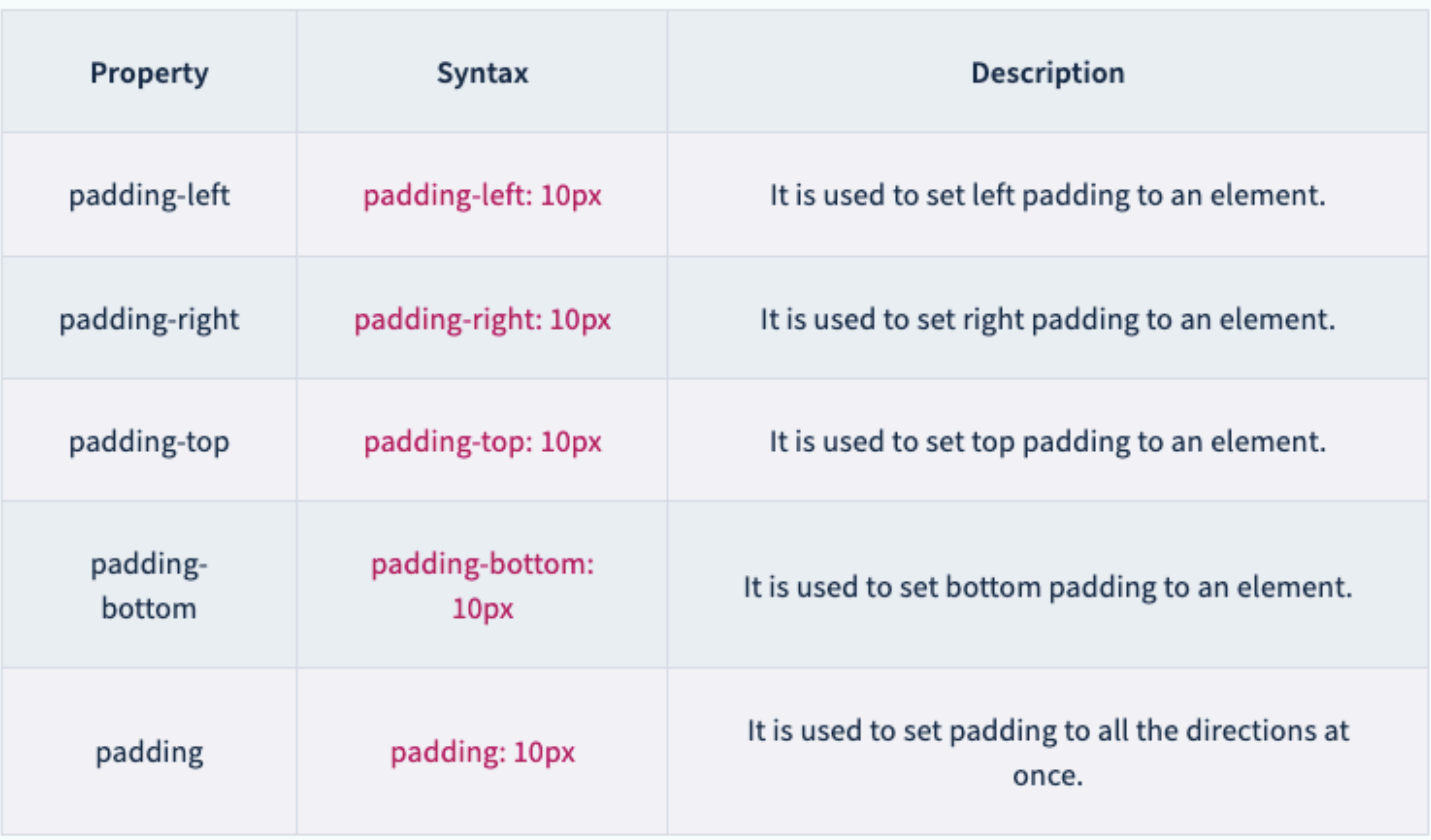
For example, take two elements, padding is added on one p element and the second p element doesn't have padding property.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS Padding</title>
</head>
<style>
    p {
        border: 1px solid black;
    }
    .with-padding {
        padding: 10px;
    }

    .without-padding {
        padding: 0px;
    }
</style>
<body>
    <p class="with-padding">I am learning about CSS padding. This contains
    <p class="without-padding">I am learning about CSS padding. This doesn'
</body>
</html>
```

CSS Padding Properties



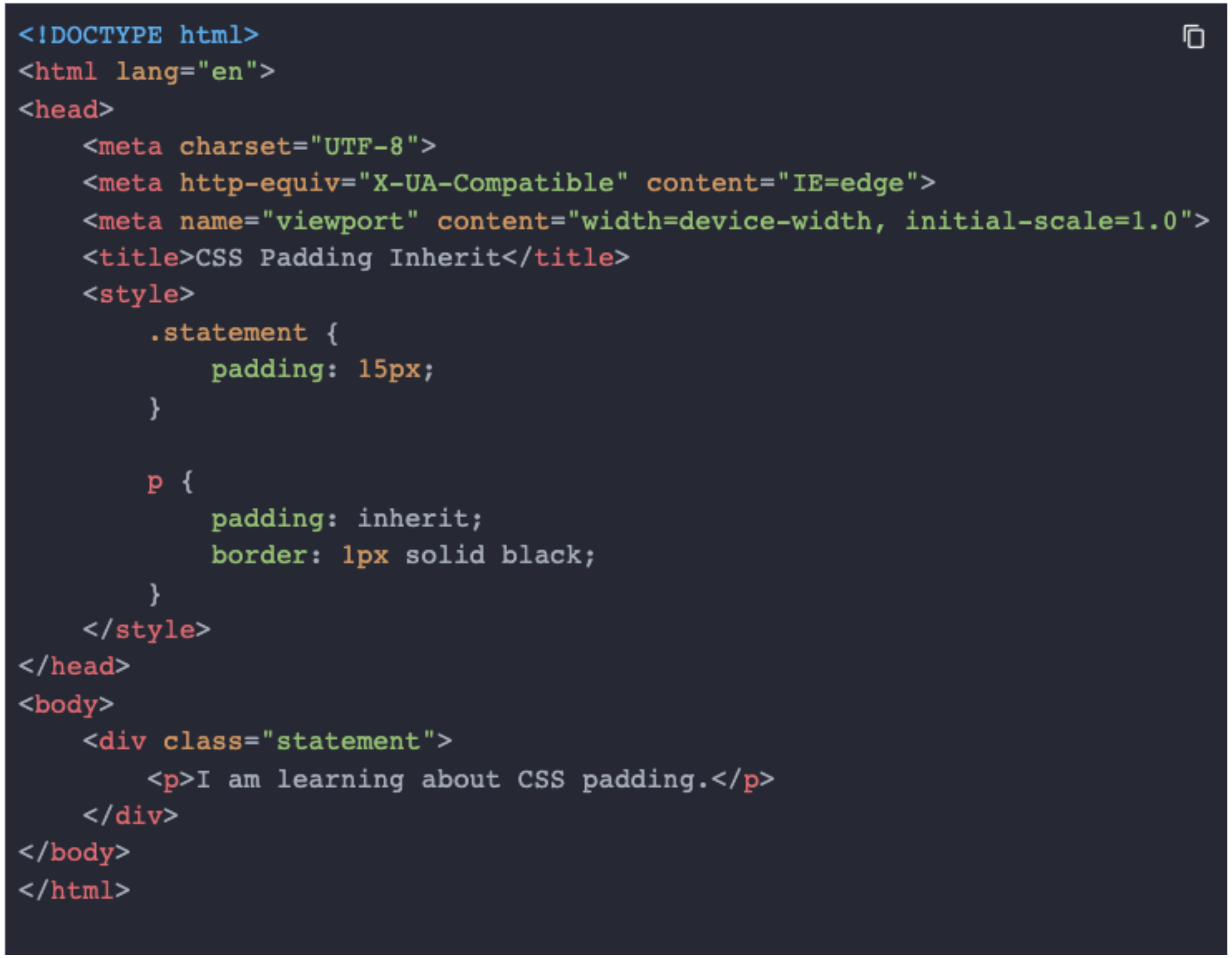
Property	Syntax	Description
padding-left	padding-left: 10px	It is used to set left padding to an element.
padding-right	padding-right: 10px	It is used to set right padding to an element.
padding-top	padding-top: 10px	It is used to set top padding to an element.
padding-bottom	padding-bottom: 10px	It is used to set bottom padding to an element.
padding	padding: 10px	It is used to set padding to all the directions at once.

CSS Padding Shorthand Property

- The padding declaration is a shorthand property to declare padding to all the four directions if padding in all four directions is the same so that you don't have to write padding-right, padding-left, padding-top, padding-bottom individually
- Ex: padding: 20px;
- If padding property has 2 values then the first value is applied to padding-top, padding-bottom of the element, and the second value is applied to padding-left, padding-right.
- Ex: padding: 10px 20px;
- If three values are provided to padding property then the first value is applied to padding-top, the second value is applied to padding-right and padding-left, the third value is applied to padding-bottom.
- Ex: padding: 10px 15px 20px;
- If four values are provided to the padding property, the first value is applied to padding-top, the second to padding-right, the third to padding-bottom, and the fourth value is applied to padding-left
- Ex: padding: 10px 15px 20px 25px
-

CSS Padding from Parent

To inherit the padding from the parent element you can use padding: inherit as discussed above.

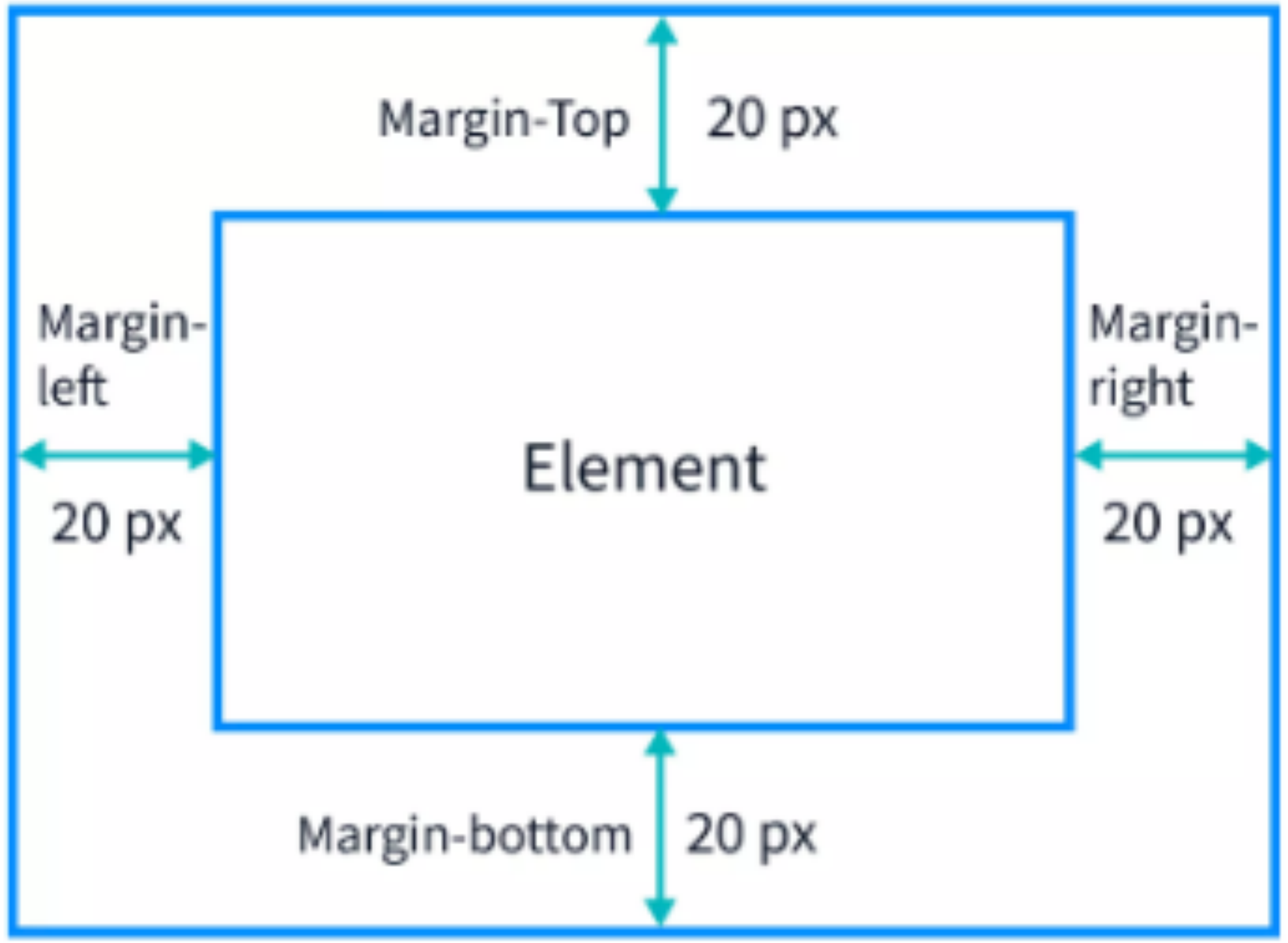


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS Padding Inherit</title>
    <style>
        .statement {
            padding: 15px;
        }

        p {
            padding: inherit;
            border: 1px solid black;
        }
    </style>
</head>
<body>
    <div class="statement">
        <p>I am learning about CSS padding.</p>
    </div>
</body>
</html>
```

CSS Margins

- **Margin is a CSS property that defines the empty space around an HTML element**. Margins ensure that the specified region around an element remains unoccupied by any neighbouring element. For eg, in the figure below, we observe that the element has a margin of 20 pixels on all four sides
-



Margin Syntax

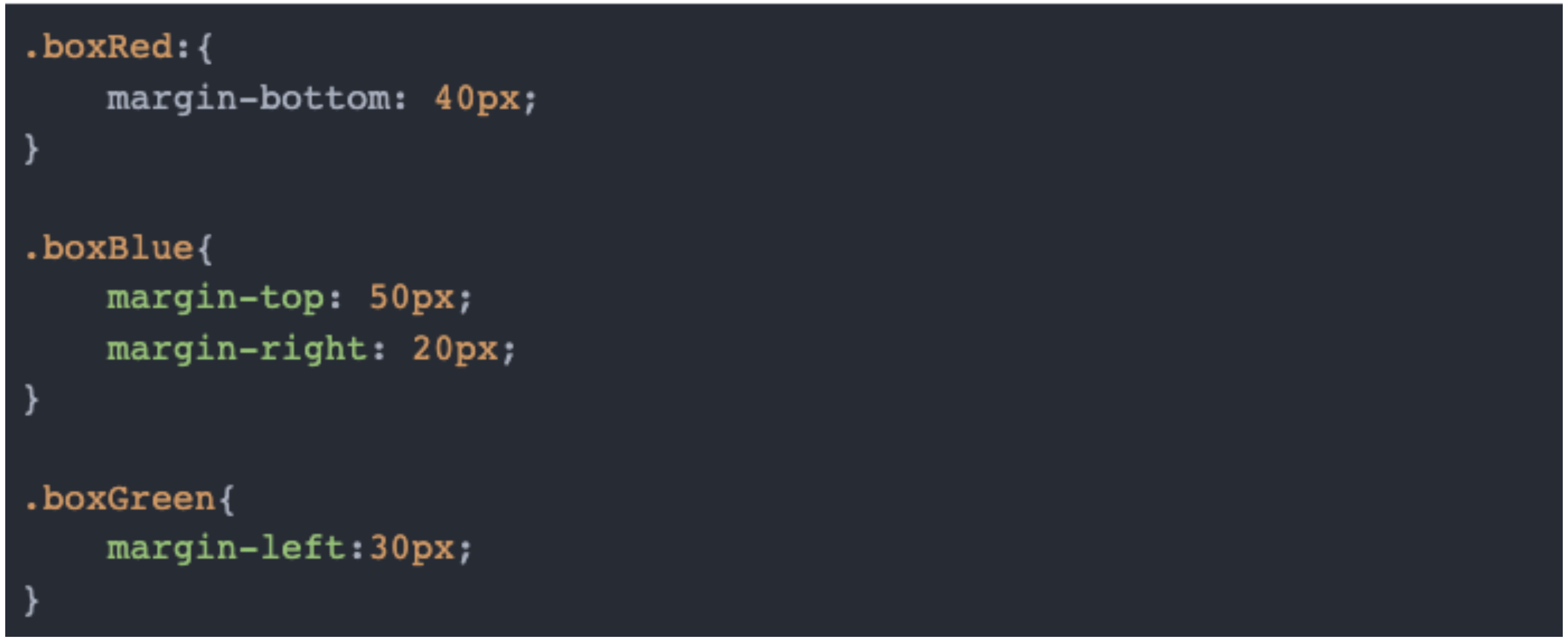
- Margin in CSS is assigned using the margin property name and a margin value. The margin property can be any of the following and corresponds to the direction of the HTML element on which it applies the margin value
- The CSS margin value can be:
- a fixed value margin-top: {value} (*value* can be given as px, pt, em, rem, etc.)
- A percentage of the parent component’s dimensions margin-top: {percentage}
- or a keyword margin-top: {keyword} (*keyword* can be auto, inherit, initial, revert, unset)
- The default margin value is *auto*. This leaves it up to the browser to calculate the margin value automatically

CSS Margin Shorthand Property

- Shorthand margin property is a short-cut way of margin assignment which enables us to assign multiple margin properties on an element with a single declaration. It comes in handy when the same margin value needs to be set on certain sides of the element. These are short, crisp, and save us some extra lines of code.
- Instead of specifying the margin value for each of the sides separately, we can use different shorthand CSS margin notations to save us the extra effort
- The single value is set as margin on all four sides
- The first value is set as vertical margin (top & bottom) while the second is set as horizontal margin (right & left)
- The first value is set as top margin, second as horizontal margin (right & left) while the third is set as bottom margin
- The values are assigned to margins on sides starting from top and moving in a clockwise direction, i.e. the four values are set as top, right, bottom and left margin respectively

Collapsing margins in CSS

- Sometimes the top and bottom margins of two adjacent elements are collapsed into one margin whose thickness is equal to the larger of the two margin values. The concept of collapsing margins in CSS is related to the vertical margins (*margin-top* and *margin-bottom* properties). It does not happen for the horizontal margins



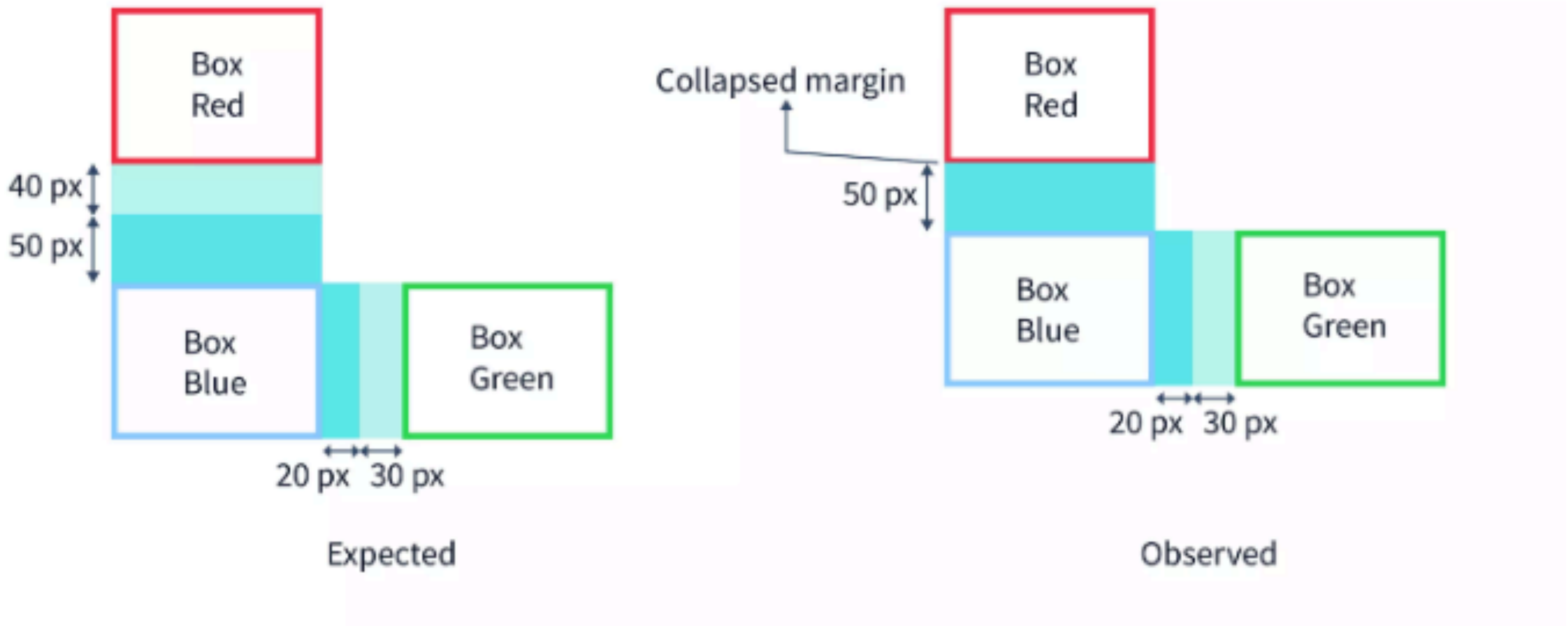
```
.boxRed:{
    margin-bottom: 40px;
}

.boxBlue{
    margin-top: 50px;
    margin-right: 20px;
}

.boxGreen{
    margin-left:30px;
}
```

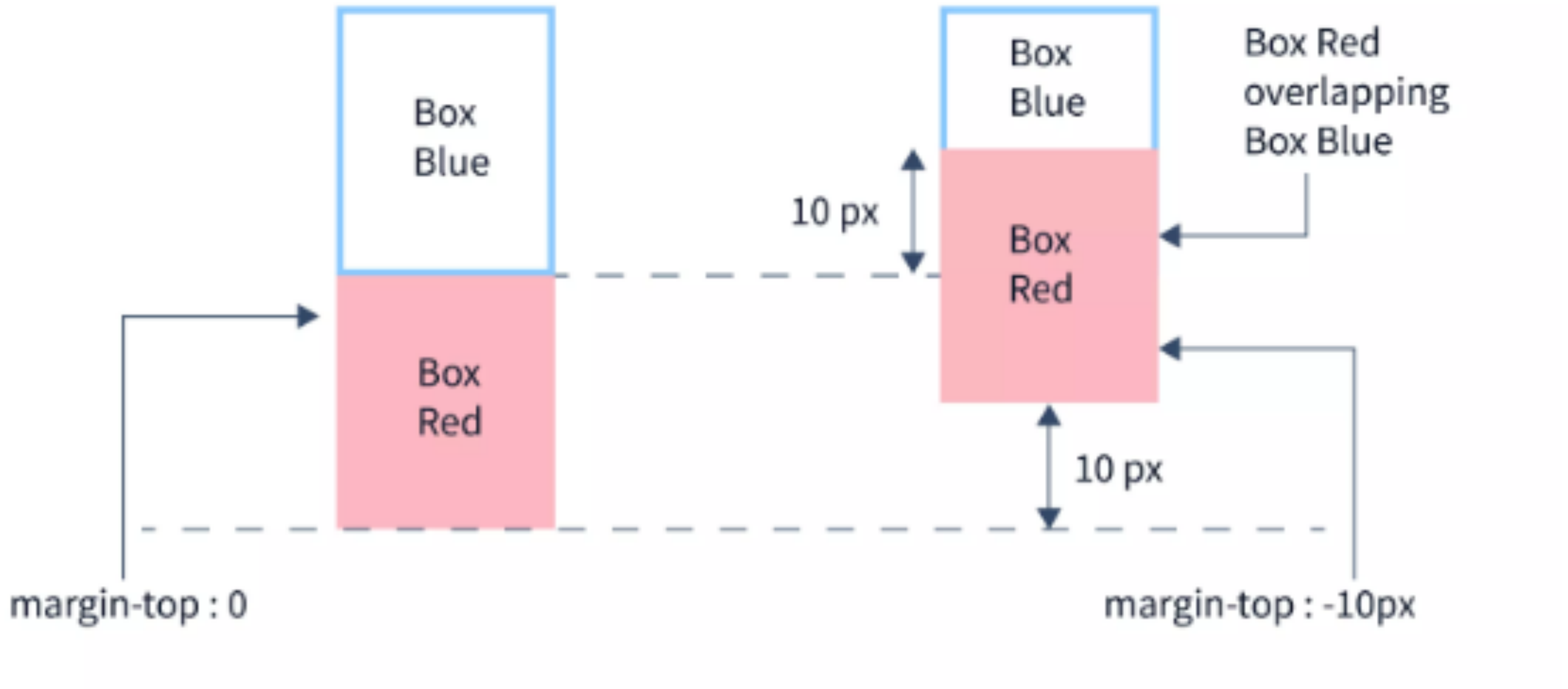
Collapsing margins in CSS

- In the above example, we have set a *margin-bottom* of 40px on **boxRed** and a *margin-top* of 50px on **boxBlue**. Also, **boxBlue** has a *margin-right* of 20px while **boxGreen** on it's right has a *margin-left* of 30px. We observe that, in the horizontal direction, the margin between **boxBlue** and **boxGreen** is 50px (20px + 30px) which is exactly as per our expectation. In the vertical direction, however, the expected vertical spacing between **boxRed** and **boxBlue** is 90px (40px + 50px), but due to collapsing of margins in CSS, the effective spacing is 50px (the maximum of 40px and 50px)

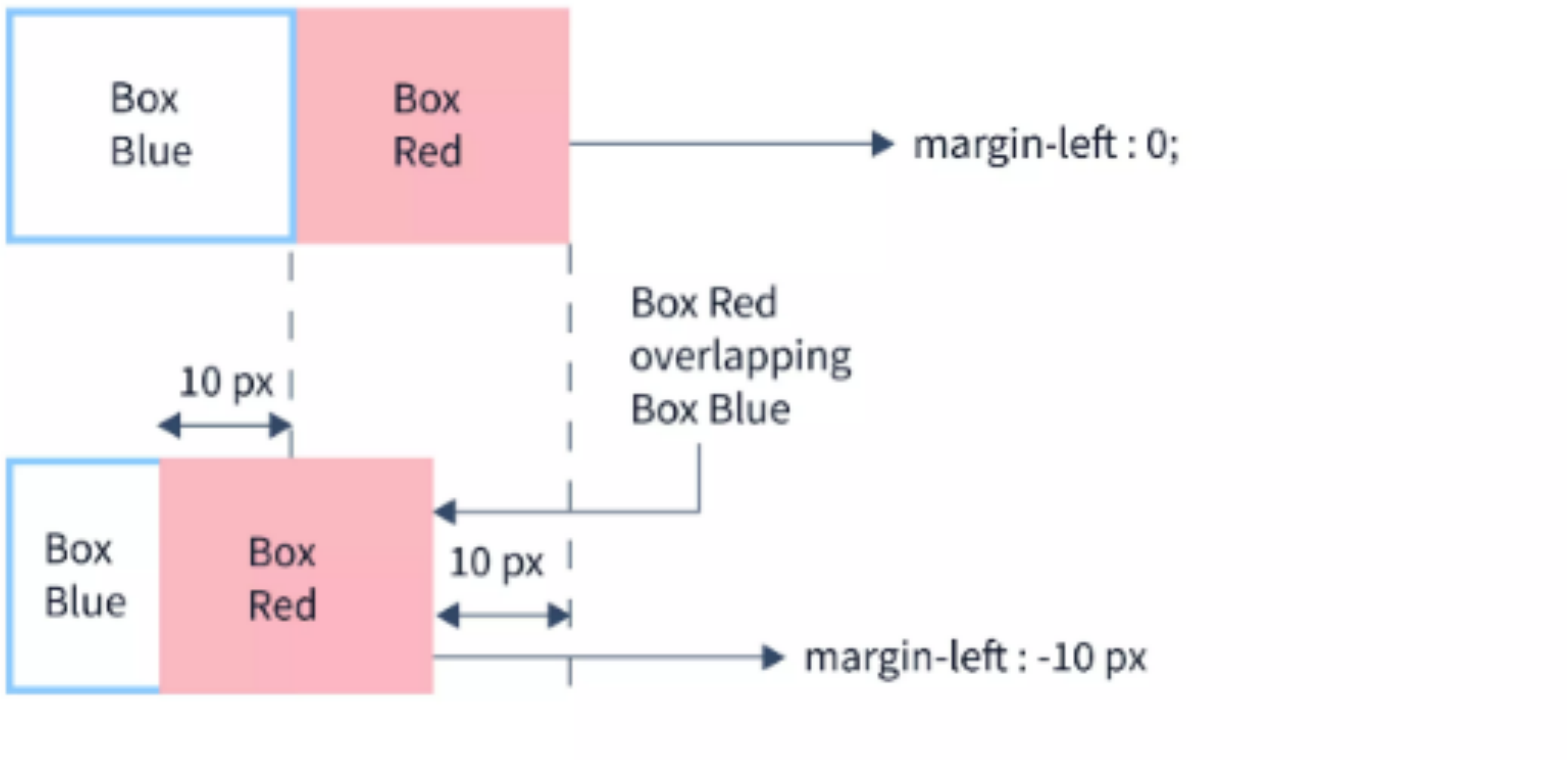


Negative Margins in CSS

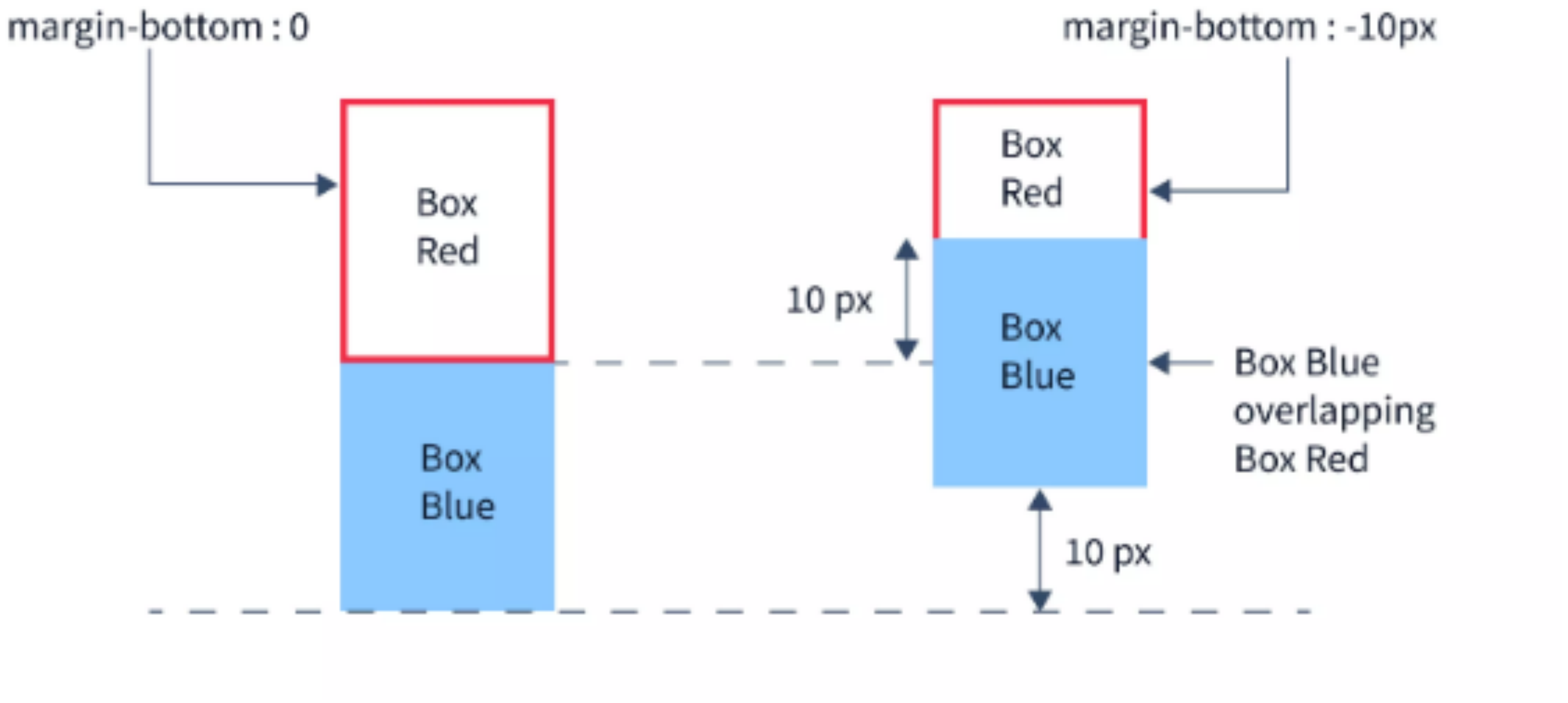
- Negative margins in CSS refers to the usage of negative values as margins. These are used to reduce the spacing between the elements and draw them closer together. Overlapping of HTML elements will occur if the negative margin value exceeds the existing spacing between the elements.
- If we use negative values with margin-top or margin-left properties, it draws our target element closer to its top or left neighbor respectively.
- If we use negative values with margin-bottom or margin-right properties, it pulls the neighboring elements closer to our target element.



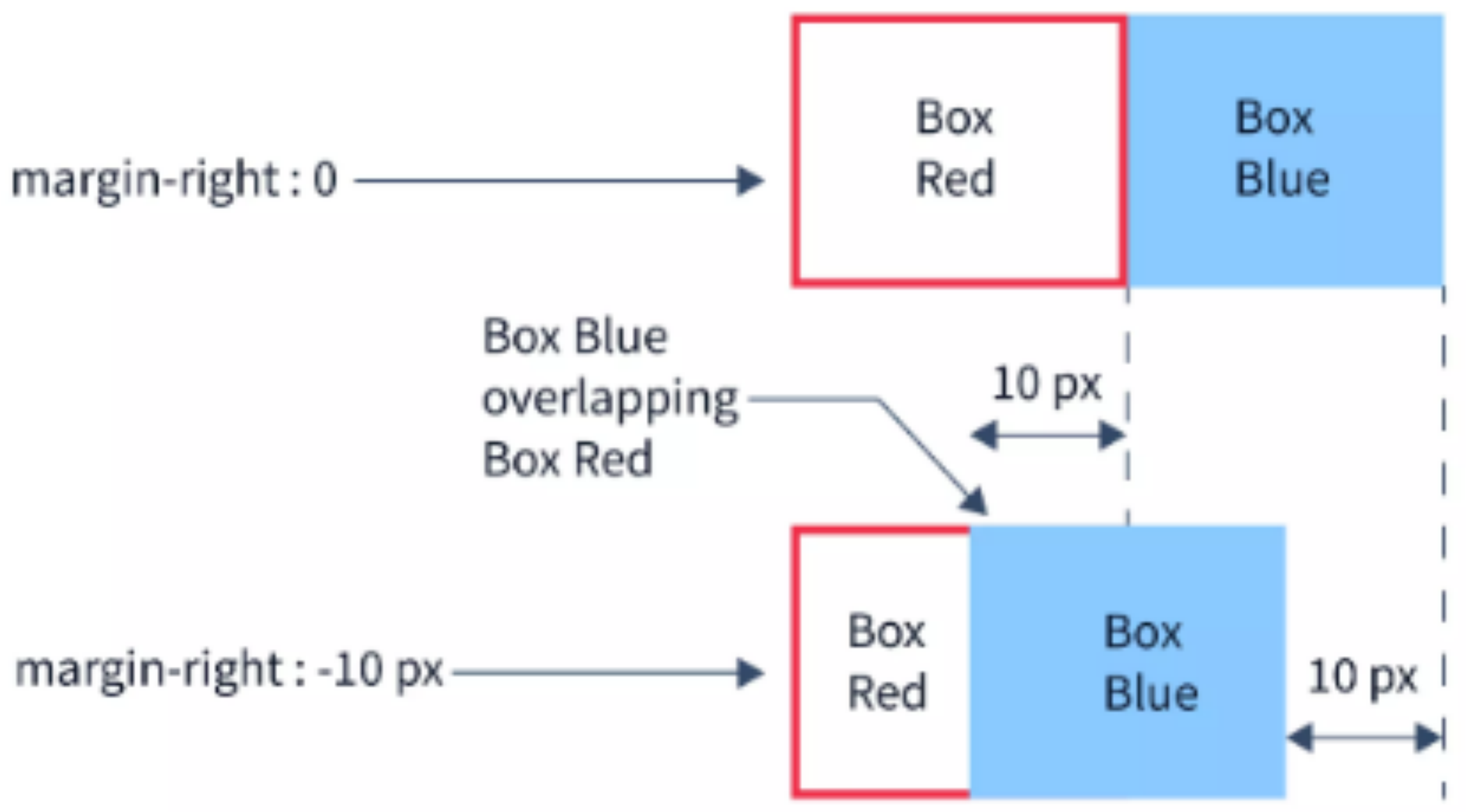
- Here while rendering boxRed, the browser calculates the baseline from where to start rendering boxRed and shifts that by 10px upwards due to the margin value on boxRed. This causes boxRed to shift upwards by 10px and overlap on boxBlue.



- Here the browser shifts the baseline from where to start rendering boxRed by 10px towards the left due to the margin value on boxRed. This causes boxRed to shift leftwards by 10px and overlap on boxBlue



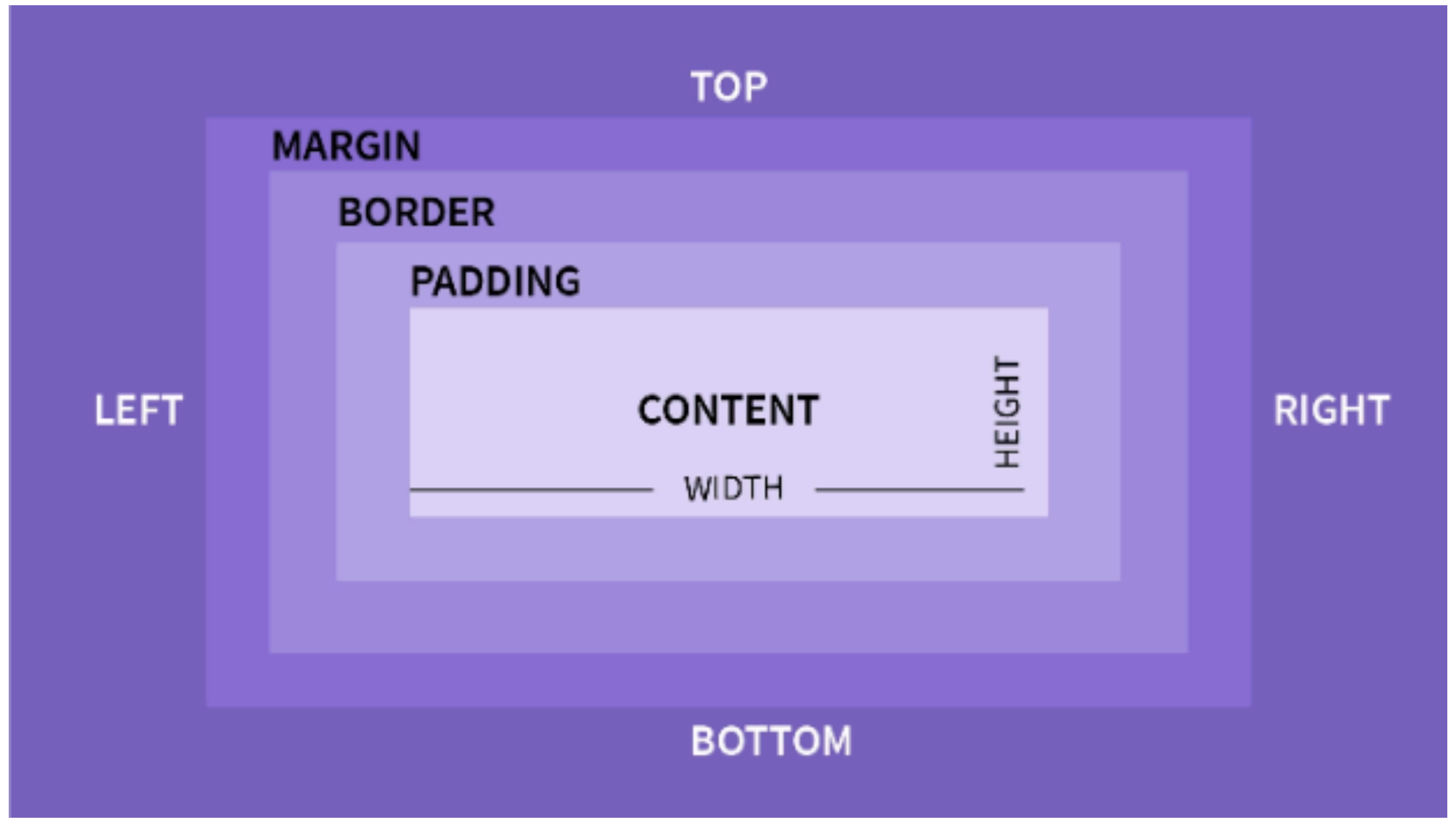
- Here the browser, while calculating the baseline for rendering boxBlue, shifts it upwards by 10px due to margin value on boxRed. This causes the boxBlue to shift upwards by 10px and overlap on boxRed.



- Here while calculating the baseline for rendering boxBlue, the browser shifts it by 10px towards the left due to the margin value on boxRed. This causes boxBlue to shift leftwards by 10px and overlap on boxRed.

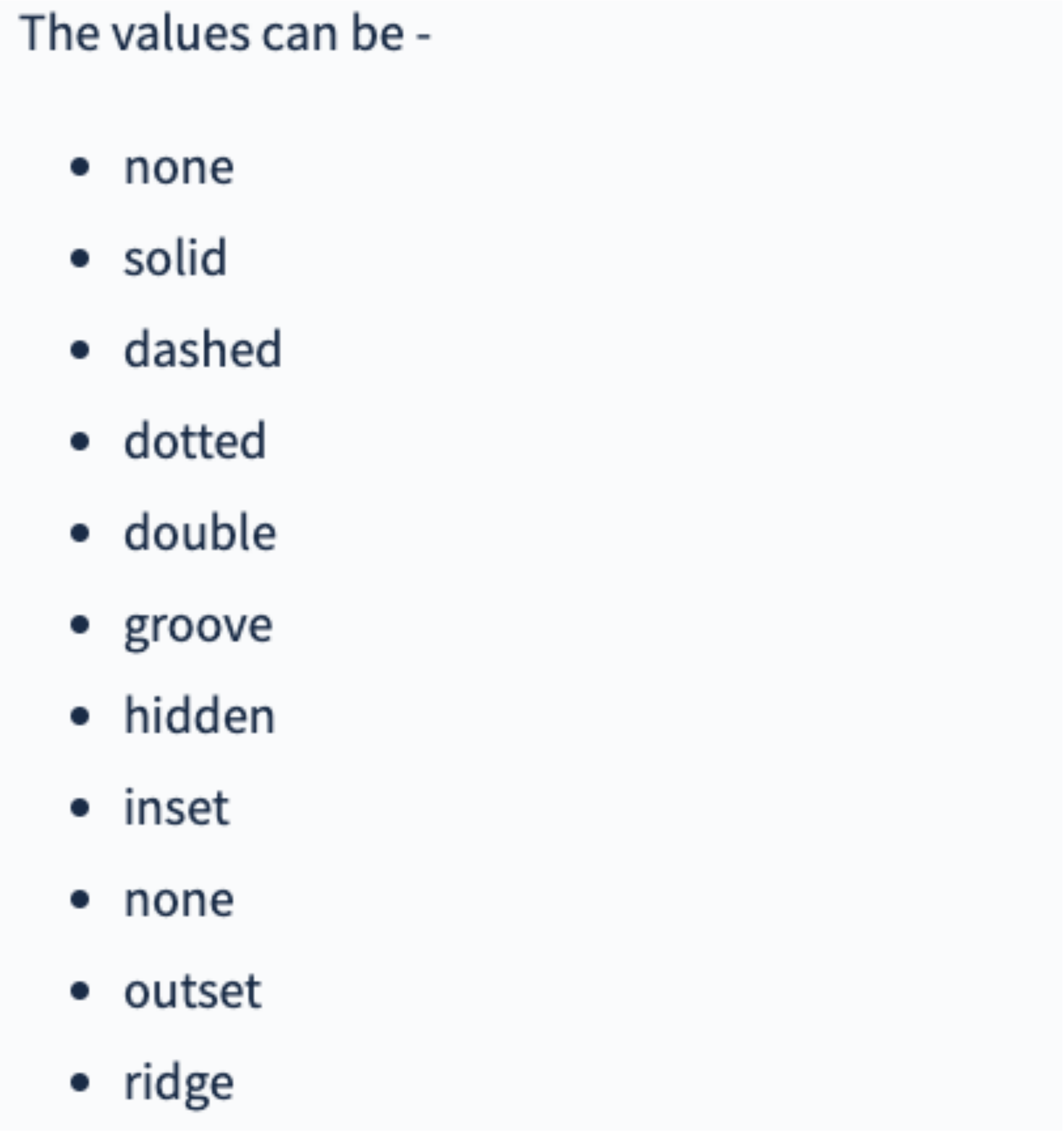
CSS Borders

- CSS is all about emphasising and decorating HTML elements. And one of the way to make your element noticable is to add **BORDERS** to it
- B**order** is a property in CSS which is used to give a border to your elements, which we will see later in this article. Let's understand how you can define a border to an element using some properties or border



CSS Border Properties

- Border-style - The property border-style is used to define the border around your element. It is the first step to add your border. It takes various values as per your needs in how you want your border to look.
- Syntax: border-style : [style-value];



The values can be -

- none
- solid
- dashed
- dotted
- double
- groove
- hidden
- inset
- none
- outset
- ridge

- Border-color - The property border-color will define the color of your border. Here you can give the color in any format like Hex(#2f2f2f), RGB[rgb(255,0,0)] or even the color-name(red). This will set the color of all the 4 sides of your border.
- Border-width - The property border-width will define the width of your border. It is used to give your border a particular thickness. You can use any size unit you want like px, em, rem etc
- Border-radius - you might have noticed that all the borders above are in rectangular shape. What if I want the vertices to be round? You can do that using border-radius property. Just like border-width, you can use any unit for size here as well like px, em, percentages, etc.

Border-color Properties

- The border-color property can be subdivided into 4 CSS properties. Here also you can specify color in any format you want like Hex, RGB or Name.
- border-bottom-color - This property is used to add color only to the bottom border
- border-top-color - This property is used to add color only to the top border
- border-left-color - This property is used to add color only to the left side border
- border-right-color - This property is used to add color only to the right side border

Border Properties Using Shorthand

- For adding border to your element, border-style property is a compulsory one without which border will not occur. border-width and border-color can be used as per the programmers need.
- But defining all these 3 using separate properties is not considerable. There is a shortcut to write all the property in one place. You can use the shorthand **border** property and include all the 3 values of properties mentioned above.
- Syntax : border: 'border-width' 'border-style' 'border-color';

Box sizing in CSS

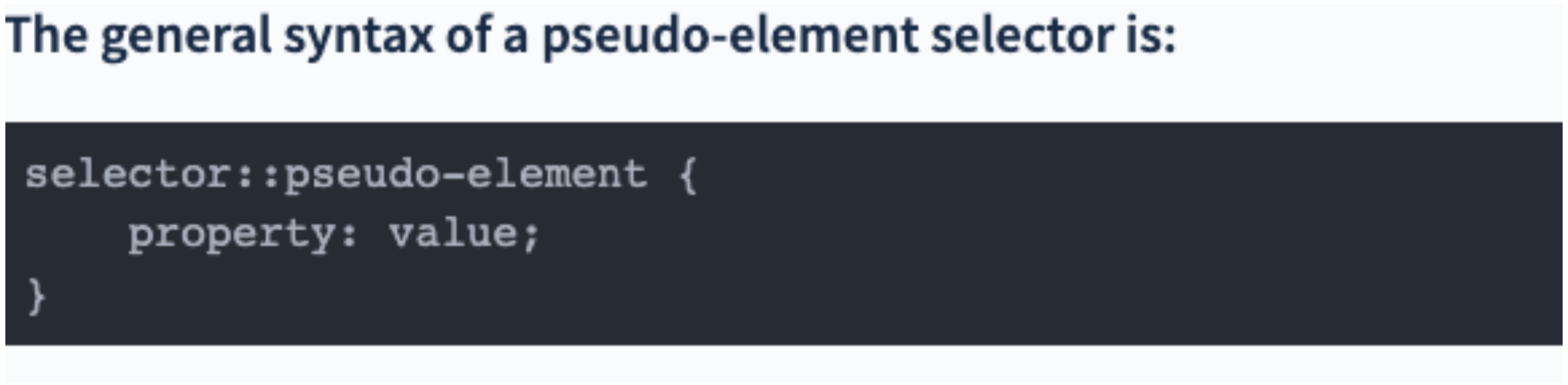
- Every property like margin, padding, height, width, font-size, etc. in CSS that has some dimensions or length needs a unit.
- CSS provides us with lots of units, some of whose values are fixed and are called **absolute units**, while there are others whose values are relative to other values like that of the parent element or to the default value for that particular HTML element, these are called **relative units**.
- Among the absolute units in CSS, we have centimeter, millimeter, pixel, etc., While among the relative units in CSS, we have percentage, em, vh, rem, etc

Pseudo Element in CSS

- In CSS, we use selectors to style specific HTML elements like styling all p elements, styling all elements with class title, etc. Pseudo-elements are keywords added to CSS selectors that apply styles to a specific part of an HTML element.
- Ex:
- ::first-letter - Style first letter of the content of all p elements.
- ::first-line - Style first line of the content of all p elements
- Only one pseudo-element in CSS can be used in a selector. It must appear after the double-color (::) of the CSS selector.

Pseudo-elements Syntax

- A pseudo-element in CSS is denoted using a double-colon (::). Though, most of the browsers support single-colon (:) for pseudo-elements.



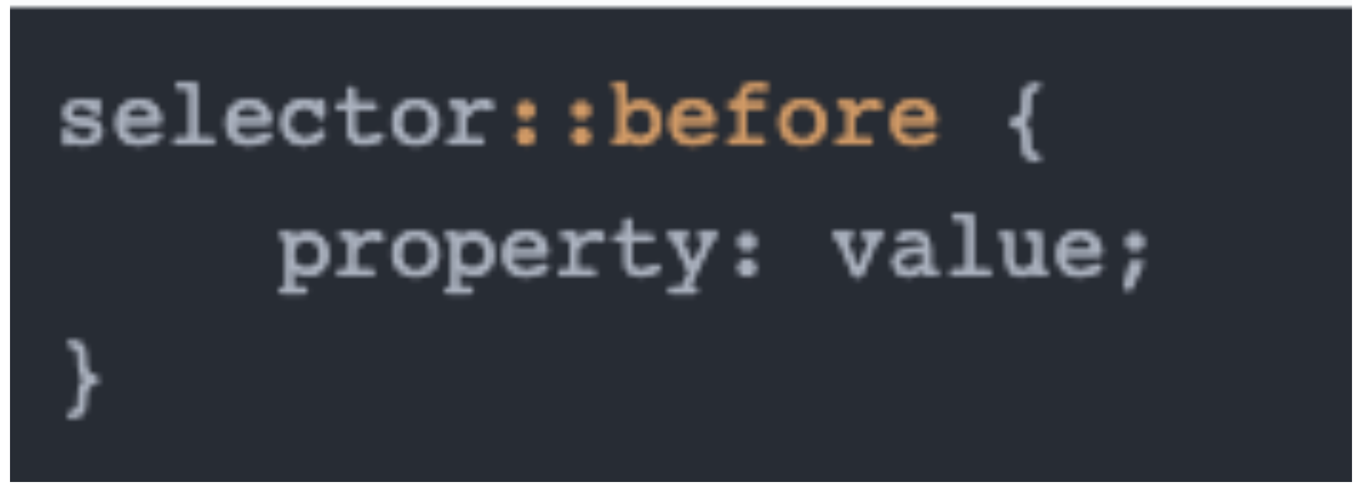
The general syntax of a pseudo-element selector is:

```
selector::pseudo-element {
    property: value;
}
```

- selector : A pattern to select the HTML elements that we want to style.
- pseudo-element : A keyword added to a CSS selector that lets us to style specific parts of the selected HTML elements

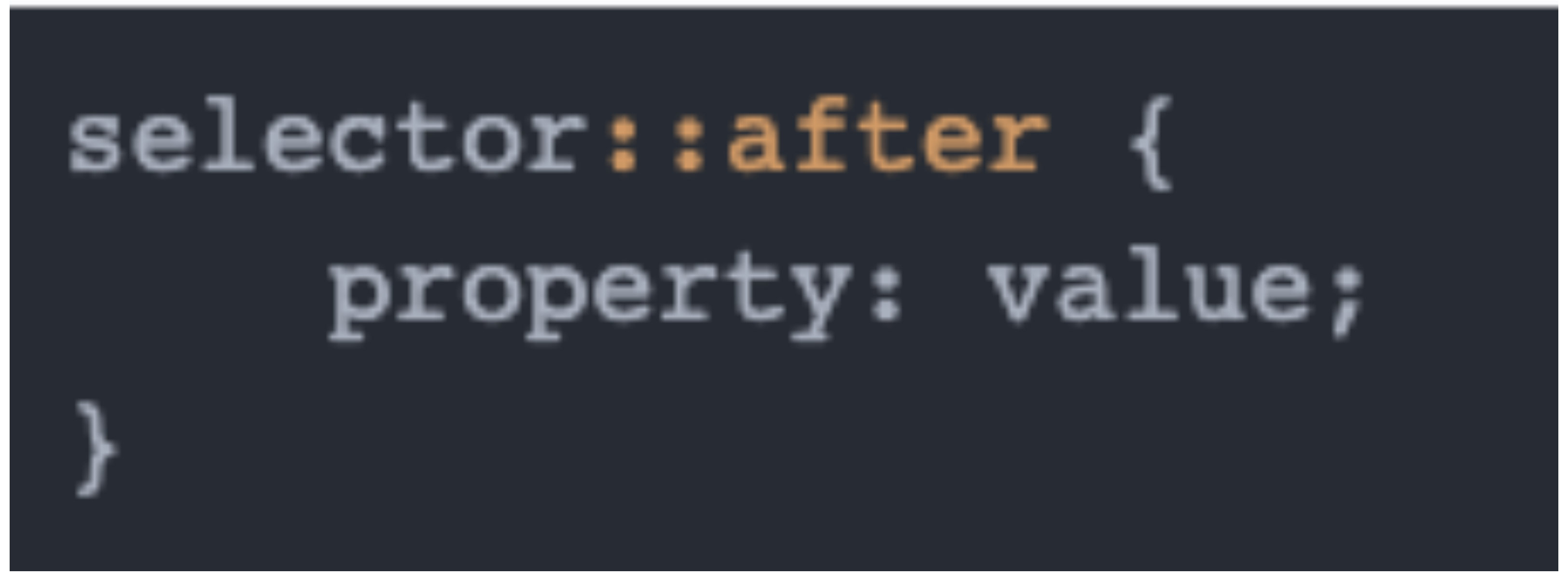
List of Pseudo-elements in CSS

- CSS provides a list of pseudo-elements used to style specific parts of an HTML element
- ::before - The ::before creates a pseudo-element used to **insert some content before the selected HTML element**. The content property is used to insert the desired content before the selected HTML element. The inserted content is inline by default
- The content added using the content property can be styled using other CSS properties like background-color, color, font-weight, etc



```
selector::before {
    property: value;
}
```

- ::after - The ::after creates a pseudo-element used to **insert some content after the selected HTML element**. The content property is used to insert the desired content after the selected HTML element. The inserted content is inline by default.
- Syntax:

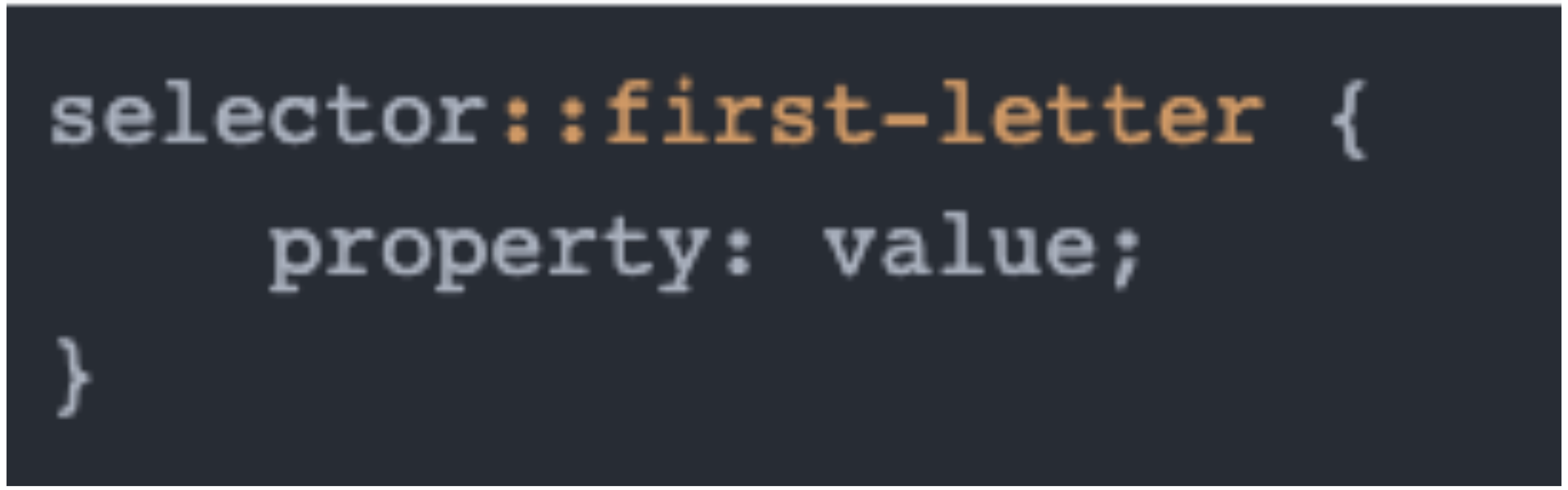


```
selector::after {
    property: value;
}
```

-

::first-letter

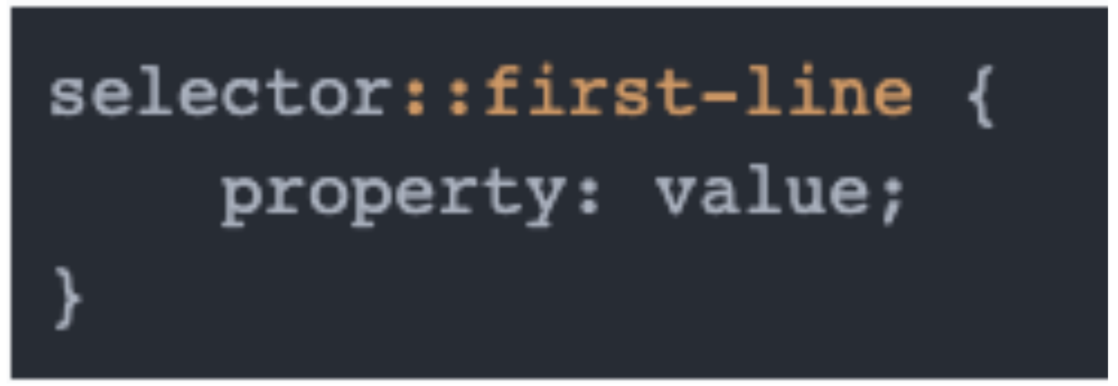
- The ::first-letter pseudo element in CSS is used to apply styles to the first letter of the first line of the selected HTML block-level element.
- A **block-level** element is an HTML element that begins a new line on a web page and extends the full width of the available horizontal space of its parent element.
- The ::first-letter pseudo-element works only when the selected block-level element is not preceded by other content like images, inline tables, etc
- Syntax:



```
selector::first-letter {
    property: value;
}
```

:first-line

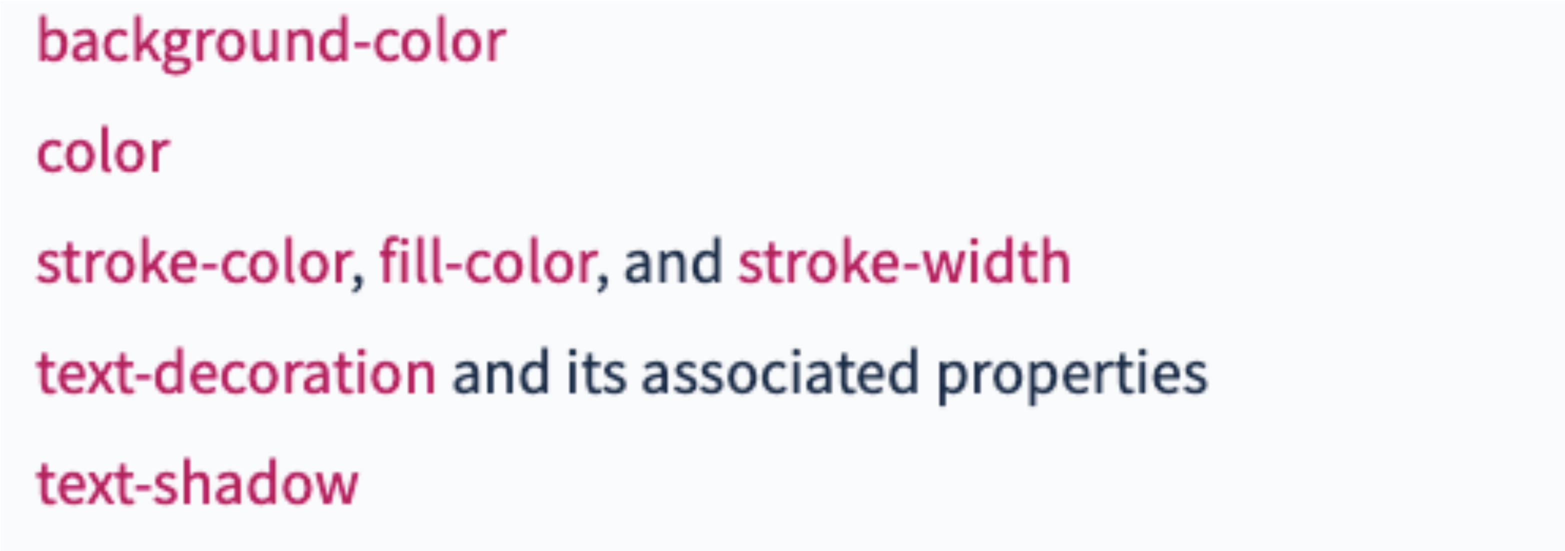
- The ::first-line pseudo element in CSS is used to **apply style to the first line of the selected HTML block-level element**. The length of the first line depends on many factors like the width of the element, the width of the document, and the font size.
- Only a subset of CSS properties can be used with the ::first-line pseudo-element.
- All **background** properties such as background, background-color, etc
- All **font** properties such as font-style, font-weight, etc
- Other properties such as color, letter-spacing, text-decoration, etc



```
selector::first-line {
    property: value;
}
```

::selection

- The ::selection pseudo element in CSS is used to apply styles to the part of the HTML document highlighted by the user by clicking and dragging across the text
- Only the below CSS properties can be used along with ::selection.



background-color

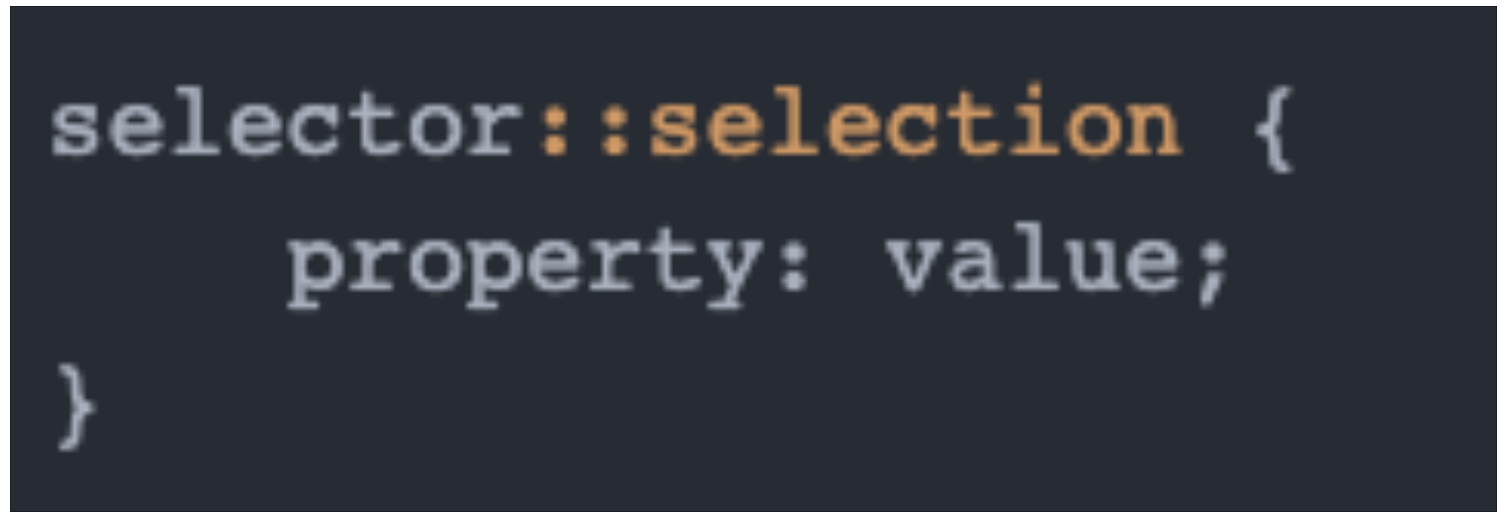
color

stroke-color, fill-color, and stroke-width

text-decoration and its associated properties

text-shadow

- Syntax :



```
selector::selection {
    property: value;
}
```

Thank You

Abhi Chakraborty (27th June 2022)