1/CSS Opacity:

The `opacity` property specifies the opacity/transparency of an element.

Transparent Image

The `opacity` property can take a value from 0.0 - 1.0. The lower the value

Example

```
img {
  opacity: 0.5;
}
```

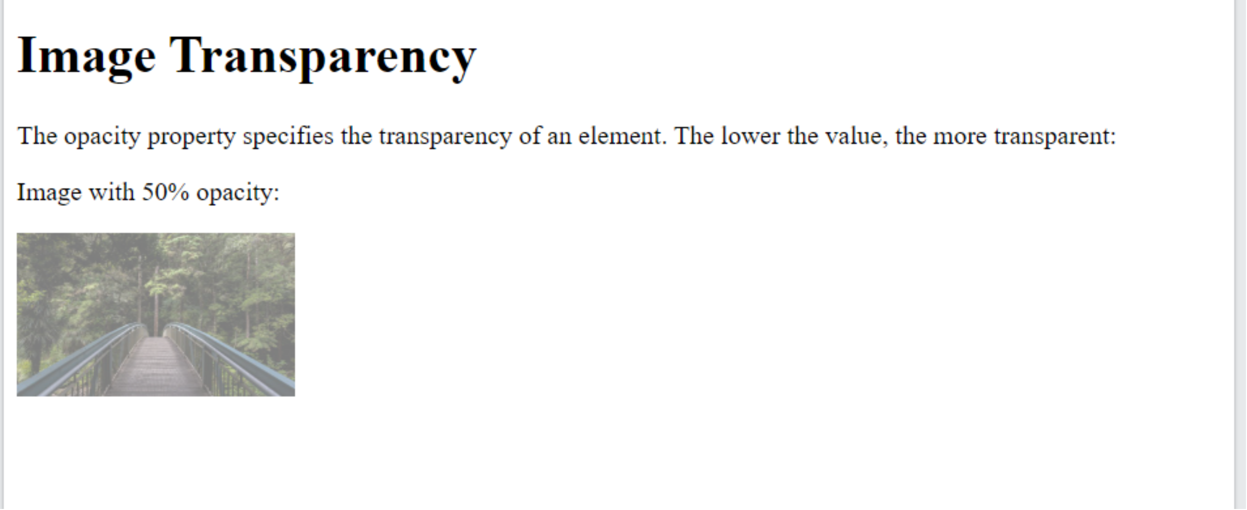


Image Transparency

The opacity property specifies the transparency of an element. The lower the value, the more transparent:

Image with 50% opacity:

Transparent Hover Effect

The `opacity` property is often used together with the `:hover` selector to change the opacity on mouse-over.

Example

<!DOCTYPE html>

<html>

```
<head>
<style>
img {
  opacity: 0.5;
}

img:hover {
  opacity: 1.0;
}
</style>
</head>
<body>

<h1>Image Transparency</h1>
<p>The opacity property is often used together with the :hover selector to change the opacity on mouse-over:</p>
<img src="img_forest.jpg" alt="Forest" width="170" height="100">
<img src="img_mountains.jpg" alt="Mountains" width="170" height="100">
<img src="img_5terre.jpg" alt="Italy" width="170" height="100">

</body>
```

```
</html>
```

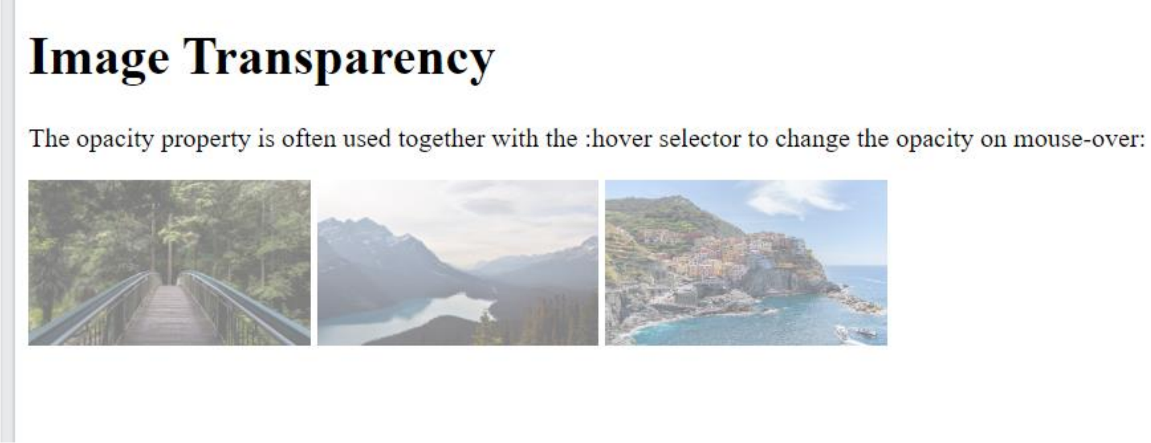


Image Transparency

The opacity property is often used together with the :hover selector to change the opacity on mouse-over:

Example explained

The first CSS block is similar to the code in Example 1. In addition, we have added what should happen when a user hovers over one of the images. In this case we want the image to NOT be transparent when the user hovers over it. The CSS for this is `opacity:1;`.

When the mouse pointer moves away from the image, the image will be transparent again.

Transparent Box

When using the `opacity` property to add transparency to the background of an element, all of its child elements inherit the same transparency. This can make the text inside a fully transparent element hard to read.

Transparency using RGBA

If you do not want to apply opacity to child elements, like in our example above, use **RGBA** color values. The following example sets the opacity for the background color and not the text.

You learned from our [CSS Colors Chapter](), that you can use RGB as a color value. In addition to RGB, you can use an RGB color value with an alpha channel (RGBA) - which specifies the opacity for a color.

An RGBA color value is specified with: rgba(red, green, blue, *alpha*).
The *alpha* parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

Tip: You will learn more about RGBA Colors in our CSS Colors Chapter.

*CSS visibility Property:

Definition and Usage

The `visibility` property specifies whether or not an element is visible.

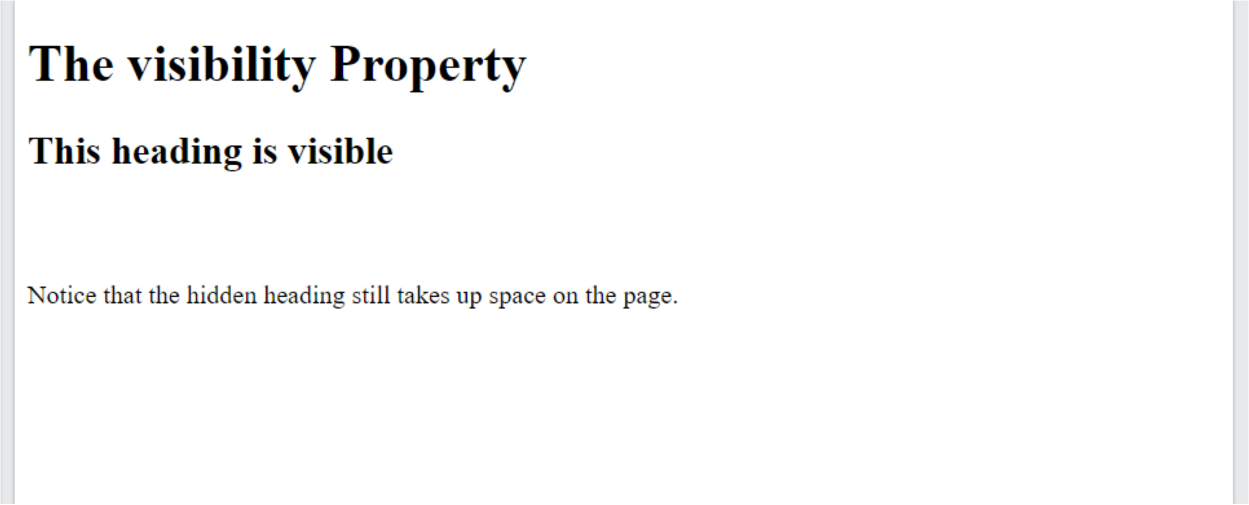
Tip: Hidden elements take up space on the page. Use the display property to both hide and remove an element from the document layout!

Example

Make <h2> elements visible and hidden:

```
<!DOCTYPE html>
<html>
<head>
<style>
h2.a {
  visibility: visible;
}


h2.b {
  visibility: hidden;
}
```

```
</style>
</head>
<body>

<h1>The visibility Property</h1>

<h2 class="a">This heading is visible</h2>

<h2 class="b">This heading is hidden</h2>

<p>Notice that the hidden heading still takes up space on the page.</p>

</body>
</html>
```



The visibility Property

This heading is visible

Notice that the hidden heading still takes up space on the page.

CSS resize Property:

Definition and Usage

The `resize` property defines if (and how) an element is resizable by the user.

Note: The `resize` property does not apply to inline elements or to block elements where overflow="visible". So, make sure that overflow is set to "scroll", "auto", or "hidden".

CSS Syntax

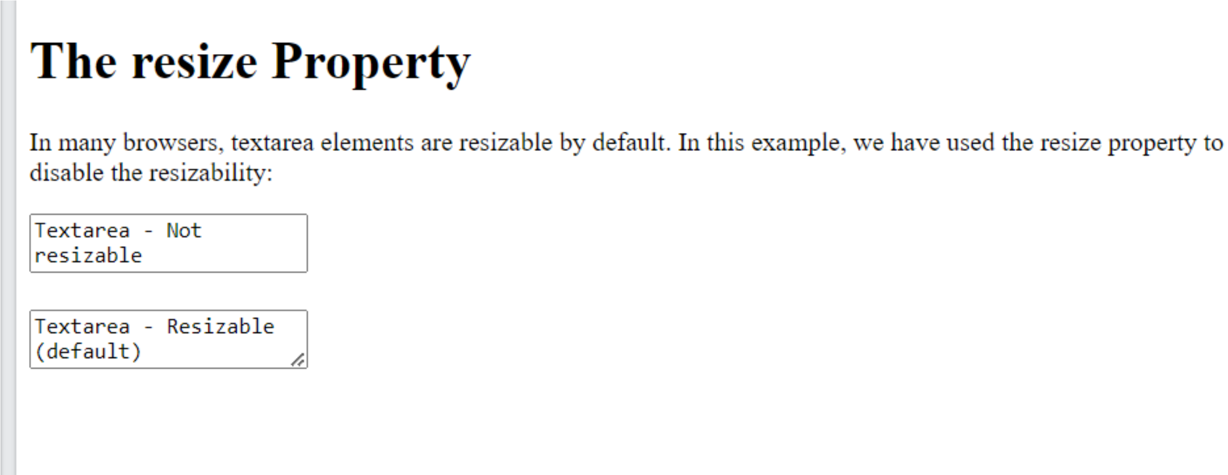
```
resize: none|both|horizontal|vertical|initial|inherit;
```

Example

In many browsers, <textarea> is resizable by default. Here, we have used the resize property to disable the resizability:

```
<!DOCTYPE html>

<html>

<head>

<style>

textarea#test {

  resize: none;

}

</style>

</head>

<body>



<h1>The resize Property</h1>
```

```
<p>In many browsers, textarea elements are resizable by default. In this example, we have used the resize property to disable the resizability:</p>

<textarea id="test">Textarea - Not resizable</textarea>

<br><br>

<textarea>Textarea - Resizable (default)</textarea>

</body>

</html>
```



The resize Property

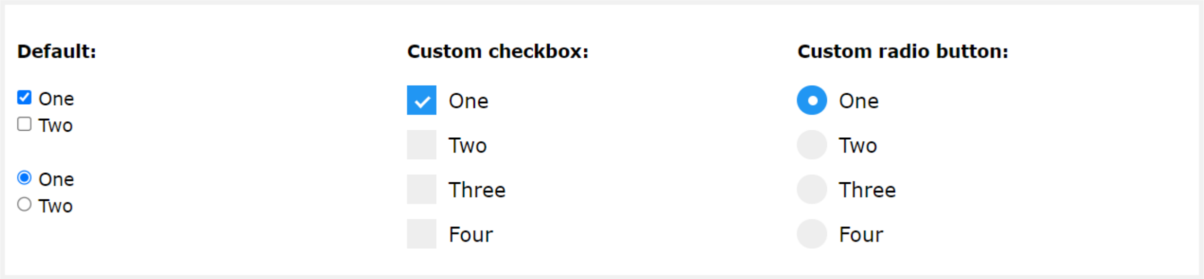
In many browsers, textarea elements are resizable by default. In this example, we have used the resize property to disable the resizability:

Textarea - Not resizable

Textarea - Resizable (default)

“You can hide an element in CSS using the CSS properties display: none or visibility: hidden . display: none removes the entire element from the page and mat affect the layout of the page. visibility: hidden hides the element while keeping the space the same.”

How TO - Custom Checkbox:



Default:

☑ One
☐ Two

● One
○ Two

Custom checkbox:

☑ One
☐ Two
☐ Three
☐ Four

Custom radio button:

● One
○ Two
○ Three
○ Four

How To Create a Custom Checkbox

Step 1) Add HTML:

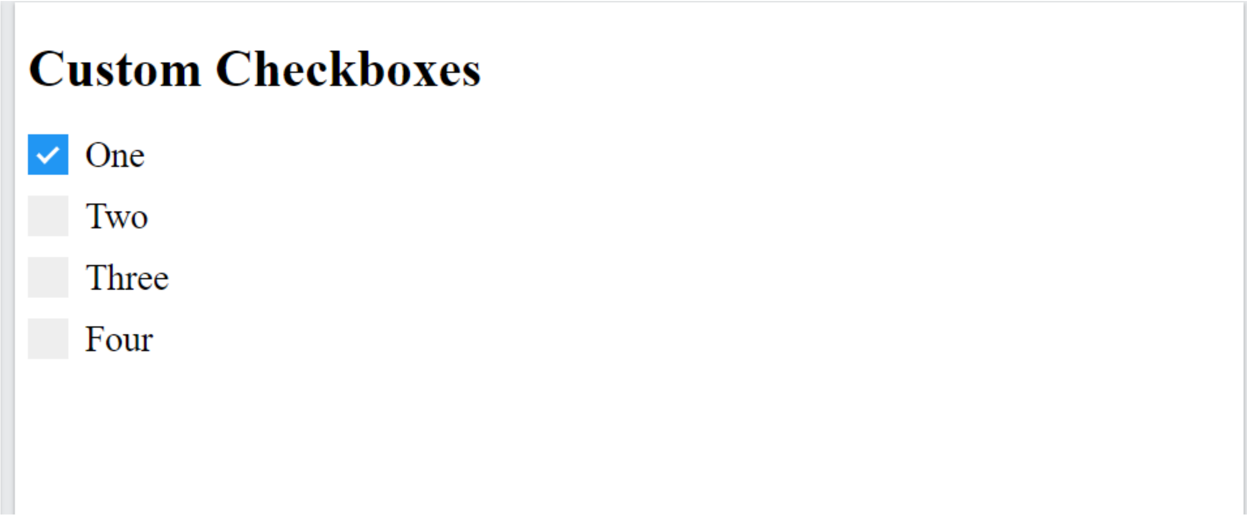
Example

```html
<label class="container">One
  <input type="checkbox" checked="checked">
  <span class="checkmark"></span>
</label>

<label class="container">Two
  <input type="checkbox">
  <span class="checkmark"></span>
</label>

<label class="container">Three
  <input type="checkbox">
  <span class="checkmark"></span>
</label>

<label class="container">Four
  <input type="checkbox">
  <span class="checkmark"></span>
</label>
```

Step 2) Add CSS:

Example

```
/* Customize the label (the container) */
.container {
  display: block;
  position: relative;
  padding-left: 35px;
  margin-bottom: 12px;
  cursor: pointer;
  font-size: 22px;
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
}

/* Hide the browser's default checkbox */
.container input {
  position: absolute;
  opacity: 0;
  cursor: pointer;
  height: 0;
  width: 0;
}

/* Create a custom checkbox */
.checkmark {
  position: absolute;
  top: 0;
  left: 0;
  height: 25px;
  width: 25px;
  background-color: #eee;
}

/* On mouse-over, add a grey background color */
.container:hover input ~ .checkmark {
  background-color: #ccc;
}

/* When the checkbox is checked, add a blue background */
.container input:checked ~ .checkmark {
  background-color: #2196F3;
}

/* Create the checkmark/indicator (hidden when not checked) */
.checkmark:after {
  content: "";
  position: absolute;
  display: none;
}

/* Show the checkmark when checked */
.container input:checked ~ .checkmark:after {
  display: block;
}

/* Style the checkmark/indicator */
.container .checkmark:after {
  left: 9px;
  top: 5px;
  width: 5px;
  height: 10px;
  border: solid white;
  border-width: 0 3px 3px 0;
  -webkit-transform: rotate(45deg);
  -ms-transform: rotate(45deg);
  transform: rotate(45deg);
}
```



Custom Checkboxes

☑ One

□ Two

□ Three

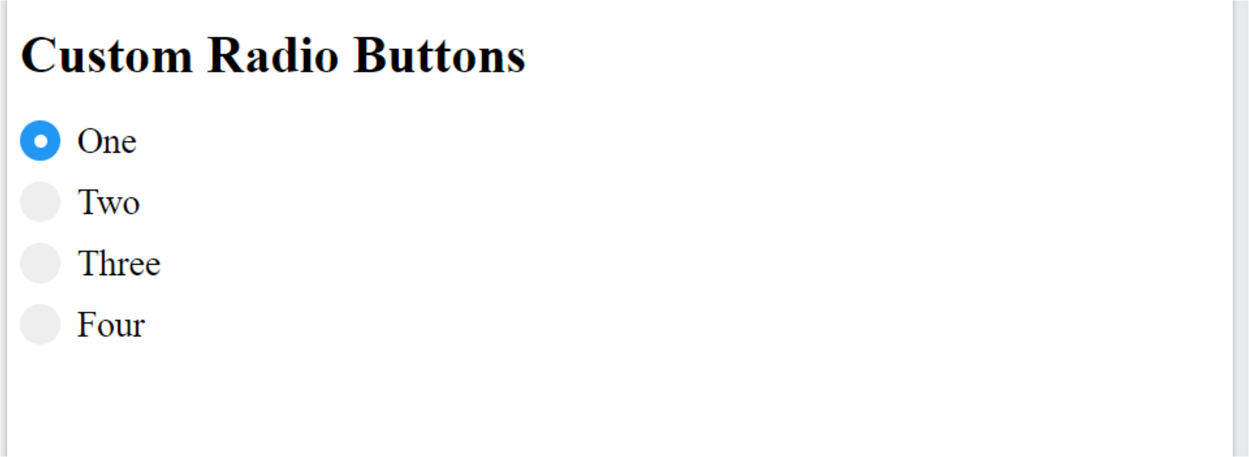
□ Four

How To Create a Custom Radio Button:

Example

```
/* Customize the label (the container) */
.container {
  display: block;
  position: relative;
  padding-left: 35px;
  margin-bottom: 12px;
  cursor: pointer;
  font-size: 22px;
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
}

/* Hide the browser's default radio button */
.container input {
  position: absolute;
  opacity: 0;
  cursor: pointer;
  height: 0;
  width: 0;
}
```

```
/* Create a custom radio button */
.checkmark {
  position: absolute;
  top: 0;
  left: 0;
  height: 25px;
  width: 25px;
  background-color: #eee;
  border-radius: 50%;
}

/* On mouse-over, add a grey background color */
.container:hover input ~ .checkmark {
  background-color: #ccc;
}

/* When the radio button is checked, add a blue background */
.container input:checked ~ .checkmark {
  background-color: #2196F3;
}

/* Create the indicator (the dot/circle - hidden when not checked) */
.checkmark:after {
  content: "";
  position: absolute;
  display: none;
}

/* Show the indicator (dot/circle) when checked */
.container input:checked ~ .checkmark:after {
  display: block;
}

/* Style the indicator (dot/circle) */
.container .checkmark:after {
  top: 9px;
  left: 9px;
  width: 8px;
  height: 8px;
  border-radius: 50%;
  background: white;
}
```



Custom Radio Buttons

● One

○ Two

○ Three

○ Four

CSS calc() Function:

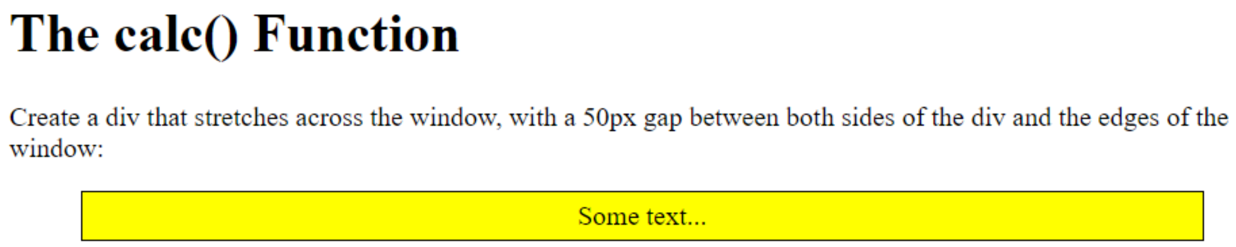
Definition and Usage

The calc() function performs a calculation to be used as the property value.

Example

Use calc() to calculate the width of a <div> element:

```
#div1 {
  position: absolute;
  left: 50px;
  width: calc(100% - 100px);
  border: 1px solid black;
  background-color: yellow;
  padding: 5px;
  text-align: center;
}
```



The calc() Function

Create a div that stretches across the window, with a 50px gap between both sides of the div and the edges of the window:

Some text...

CSS Shadow Effects:

CSS Shadow Effects

With CSS you can add shadow to text and to elements.

In these chapters you will learn about the following properties:

- `text-shadow`
- `box-shadow`

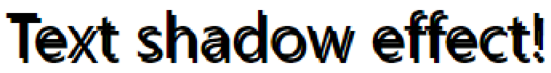
CSS Text Shadow

The CSS `text-shadow` property applies shadow to text.

Example

In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px):

```
<!DOCTYPE html>

<html>

<head>

<style>

h1 {

  text-shadow: 2px 2px;

}

</style>

</head>

<body>


<h1>Text-shadow effect!</h1>


</body>

</html>
```

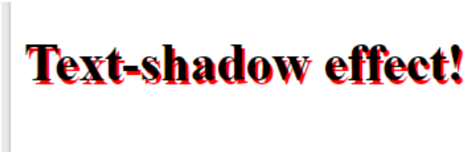


Text shadow effect!

Next, add a color to the shadow:

```
<!DOCTYPE html>

<html>

<head>
```

```
<style>

h1 {

  text-shadow: 2px 2px red;

}

</style>

</head>

<body>


<h1>Text-shadow effect!</h1>


</body>

</html>
```

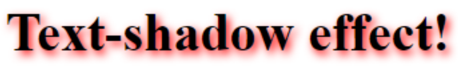


Text-shadow effect!

Then, add a blur effect to the shadow:

```
<!DOCTYPE html>

<html>

<head>

<style>

h1 {

  text-shadow: 2px 2px 5px red;

}

</style>
```

```
</head>

<body>


<h1>Text-shadow effect!</h1>


</body>

</html>
```



Text-shadow effect!

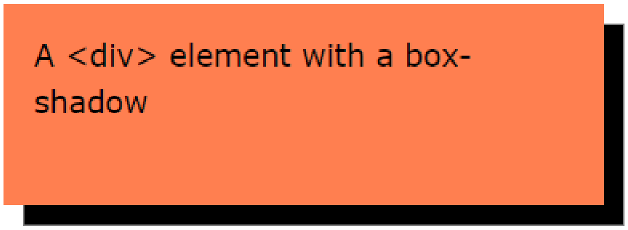
CSS Box Shadow:

CSS box-shadow Property

The CSS `box-shadow` property is used to apply one or more shadows to an element.

Specify a Horizontal and a Vertical Shadow

In its simplest use, you only specify a horizontal and a vertical shadow. The default color of the shadow is the current text-color.



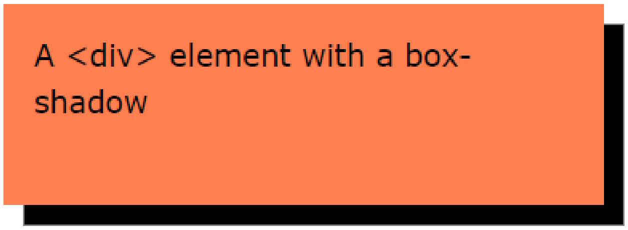
A <div> element with a box-shadow

Example

Specify a horizontal and a vertical shadow:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  height: 100px;
  padding: 15px;
  background-color: coral;
  box-shadow: 10px 10px;
}
</style>
</head>
<body>

<h1>The box-shadow Property</h1>
```

```
<div>This is a div element with a box-shadow</div>


</body>

</html>
```



A <div> element with a box-shadow

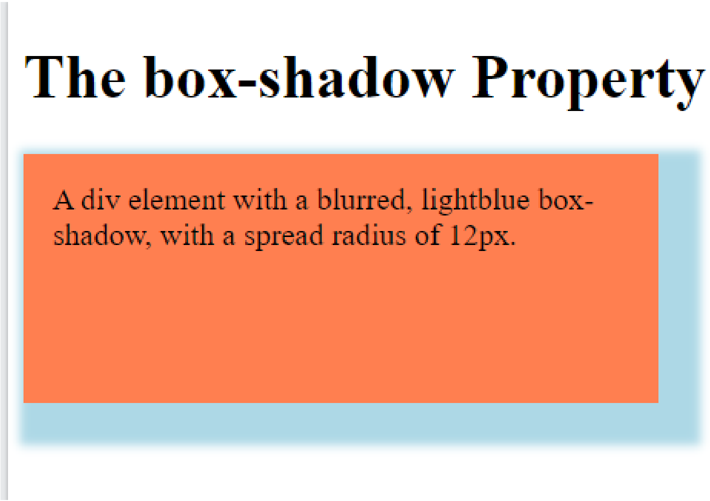
Set the Spread Radius of the Shadow

The `spread` parameter defines the spread radius. A positive value increases the size of the shadow, a negative value decreases the size of the shadow.

Example

Set the spread radius of the shadow:

```
<!DOCTYPE html>

<html>

<head>

<style>

div {

  width: 300px;

  height: 100px;

  padding: 15px;

  background-color: coral;
  box-shadow: 10px 10px 5px 12px lightblue;

}

</style>

</head>

<body>


<h1>The box-shadow Property</h1>


<div>A div element with a blurred, lightblue box-shadow, with a spread radius of 12px.</div>


</body>

</html>
```



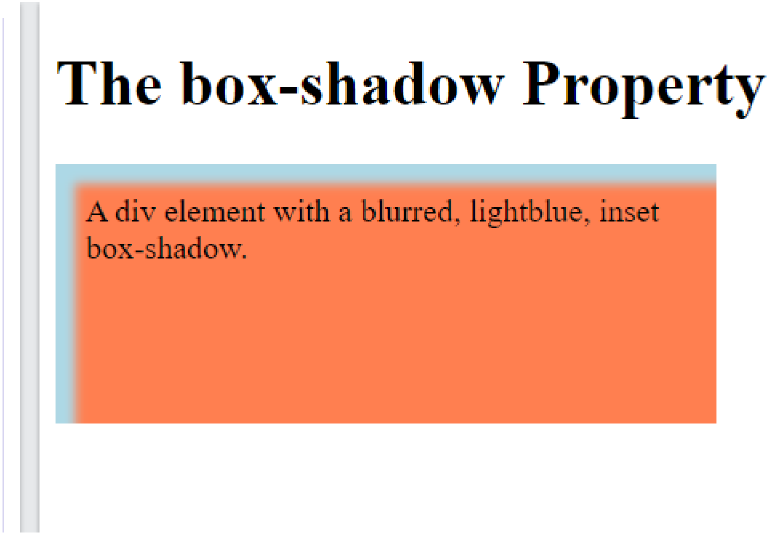
The box-shadow Property

A div element with a blurred, lightblue box-shadow, with a spread radius of 12px.

Set the inset Parameter

The `inset` parameter changes the shadow from an outer shadow (outset) to an inner shadow.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  height: 100px;
  padding: 15px;
  background-color: coral;
  box-shadow: 10px 10px 5px lightblue inset;
}
</style>
</head>
<body>

<h1>The box-shadow Property</h1>

<div>A div element with a blurred, lightblue, inset box-shadow.</div>

</body>
</html>
```



The box-shadow Property

A div element with a blurred, lightblue, inset box-shadow.