

# How to affect other elements when one element is hovered

Asked 10 years, 5 months ago    Active 10 months ago    Viewed 613k times

What I want to do is when a certain `div` is hovered, it'd affect the properties of another `div`.

551 For example, in [this JSFiddle demo](#), when you hover over `#cube` it changes the `background-color` but what I want is that when I hover over `#container`, `#cube` is affected.

★  
292  
↺

```
div {
  outline: 1px solid red;
}

#container {
  width: 200px;
  height: 30px;
}

#cube {
  width: 30px;
  height: 100%;
  background-color: red;
}

#cube:hover {
  width: 30px;
  height: 100%;
  background-color: blue;
}
```

```
<div id="container">
  <div id="cube">
  </div>
</div>
```

Run code snippet

Copy snippet to answer

[Expand snippet](#)

html    css    hover

Share   Edit   Follow   Flag

edited Jul 6 '20 at 9:21

**Penny Liu**  
 7,710   ● 5   ● 40   ● 66

asked Dec 21 '10 at 18:29

**Trufa**  
 35.7k   ● 41   ● 118   ● 180

6 Answers

Active   Oldest   Votes

▲ If the cube is directly inside the container:

1201

```
#container:hover > #cube { background-color: yellow; }
```

If cube is next to (after containers closing tag) the container:

```
#container:hover + #cube { background-color: yellow; }
```

If the cube is somewhere inside the container:

```
#container:hover #cube { background-color: yellow; }
```

If the cube is a sibling of the container:

```
#container:hover ~ #cube { background-color: yellow; }
```

Share Edit Follow Flag

edited Feb 1 '17 at 12:47



rockfeeler

3 ● 2

answered Dec 21 '10 at 18:36



Mike

13.5k ● 2 ● 15 ● 25

123 ▲ Don't forget the general sibling combinator `~` for 'cube is somewhere after container in the DOM and shares a parent' – [robertc](#) Dec 21 '10 at 18:50 ✎

4 ▲ That's pretty cool. Is there some source where I can find more information about that? Is it supported by all browser, is it CSS3? Would be great to have some more info about that. Thanks so much! – [Anonymous](#) Oct 19 '11 at 10:35

3 ▲ +1 Great answer @Mike. What if `#container` is next to `#cube`, i.e. `#container` follows `#cube`? – [PeterKA](#) Oct 2 '14 at 4:58

4 ▲ What to do if the hovered element is inside the container(that we want to be effected) ??? For example: `#cube:hover #container{Some CSS Effects}` – [Hanzallah Afgan](#) Mar 3 '15 at 15:19 ✎

2 ▲ Good answer !! What about if I want to change the parent when I hover the child. I think there is no selector for that. – [Mikel](#) Mar 8 '17 at 11:49

In this particular example, you can use:

```
#container:hover #cube {
  background-color: yellow;
}
```

This example only works since `cube` is a child of `container`. For more complicated scenarios, you'd need to use different CSS, or use JavaScript.

Share Edit Follow Flag

edited Oct 24 '19 at 17:30



TylerH

19.1k ● 49 ● 65 ● 86

answered Dec 21 '10 at 18:31



Emmett

13.1k ● 12 ● 52 ● 78

Using the sibling selector is the general solution for styling other elements when hovering over a given one, **but** it works *only if the other elements follow the given one in the DOM*. What can we do when the other

43

elements should actually be before the hovered one? Say we want to implement a signal bar rating widget like the one below:



This can actually be done easily using the CSS flexbox model, by setting `flex-direction` to `reverse`, so that the elements are displayed in the opposite order from the one they're in the DOM. The screenshot above is from such a widget, implemented with pure CSS.

[Flexbox is very well supported](#) by 95% of modern browsers.

► [Show code snippet](#)

Share Edit Follow Flag

edited Sep 9 '15 at 9:54

answered Sep 9 '15 at 4:27



Dan Dascalescu

111k ● 40 ● 276 ● 363

▲ Could this be edited so the highlight doesn't disappear when moving the mouse from 1 bar to the other? The flashing is a little distracting. – [Cerbrus](#) Sep 9 '15 at 6:23

▲ @Cerbrus: Added a solution that doesn't hide the hover when the mouse is between bars. The downside is the width of the bars is no longer equal. – [Dan Dascalescu](#) Sep 9 '15 at 9:22 ✎

1 ▲ Try this in your first snippet: on `.rating div`, remove the margin, and add `border-right: 4px solid white;` – [Cerbrus](#) Sep 9 '15 at 9:47 ✎

1 ▲ Flex direction (not well supported for IE) OR 1) black by default 2) all blue on mouse over the container 3) black for next sibling on bar hover :) – [Stephane Mathis](#) Oct 16 '15 at 10:30 ✎

1 ▲ I made this fiddle which (at least for me) made it a little more aparent on what was going on here. [jsfiddle.net/maxshuty/cj55y33p/3](http://jsfiddle.net/maxshuty/cj55y33p/3) – [maxshuty](#) Jun 15 '17 at 12:27

Only this worked for me:

10

```
#container:hover .cube { background-color: yellow; }
```

Where `.cube` is CssClass of the `#cube`.

Tested in **Firefox**, **Chrome** and **Edge**.

Share Edit Follow Flag

edited Feb 18 '16 at 10:13



Igor Ivancha

3,231 ● 4 ● 28 ● 39

answered Sep 15 '15 at 10:29



CyberHawk

848 ● 2 ● 11 ● 27

Big thanks to Mike and Robertc for their helpful posts!

If you have two elements in your HTML and you want to `:hover` over one and target a style change in the other the two elements must be directly related--parents, children or siblings. This means that the two elements either must be one inside the other or must both be contained within the same larger element.

I wanted to display definitions in a box on the right side of the browser as my users read through my site and `:hover` over highlighted terms; therefore, I did not want the 'definition' element to be displayed inside the 'text' element.

I almost gave up and just added javascript to my page, but this is the future dang it! We should not have to put up with back sass from CSS and HTML telling us where we have to place our elements to achieve the effects we want! In the end we compromised.

**While the actual HTML elements in the file must be either nested or contained in a single element to be valid `:hover` targets to each other, the css `position` attribute can be used to display any element where ever you want. I used `position:fixed` to place the target of my `:hover` action where I wanted it on the user's screen regardless to its location in the HTML document.**

The html:

```
<div id="explainBox" class="explainBox"> /*Common parent*/

  <a class="defP" id="light" href="http://en.wikipedia.or/wiki/Light">Light
/*highlighted term in text*/
</a> is as ubiquitous as it is mysterious. /*plain text*/

  <div id="definitions"> /*Container for :hover-displayed definitions*/
    <p class="def" id="light"> /*example definition entry*/ Light:
      <br/>Short Answer: The type of energy you see
    </p>
  </div>

</div>
```

The css:

```
/*read: "when user hovers over #light somewhere inside #explainBox
  set display to inline-block for #light directly inside of #definitions.*/

#explainBox #light:hover~#definitions>#light {
  display: inline-block;
}

.def {
  display: none;
}

#definitions {
  background-color: black;
  position: fixed;
  /*position attribute*/
  top: 5em;
  /*position attribute*/
  right: 2em;
  /*position attribute*/
  width: 20em;
  height: 30em;
  border: 1px solid orange;
  border-radius: 12px;
```

```
border-radius: 12px;
padding: 10px;
}
```

In this example the target of a `:hover` command from an element within `#explainBox` must either be `#explainBox` or also within `#explainBox`. The position attributes assigned to `#definitions` force it to appear in the desired location (outside `#explainBox`) even though it is technically located in an unwanted position within the HTML document.

I understand it is considered bad form to use the same `#id` for more than one HTML element; however, in this case the instances of `#light` can be described independently due to their respective positions in uniquely `#id`'d elements. Is there any reason not to repeat the `id #light` in this case?

Share Edit Follow Flag

edited Feb 18 '16 at 10:13



Igor Ivancha

3,231 ● 4 ● 28 ● 39

answered May 10 '13 at 0:37



rick

81 ● 1 ● 2



Quite the long answer for such a short point, but here's a jsfiddle of it [jsfiddle.net/ubershmekel/bWgg6/1](http://jsfiddle.net/ubershmekel/bWgg6/1) – ubershmekel Jan 4 '14 at 1:15

1 some browsers will freak out when you use the same `ID` multiple times. Just use a `class`. – Serj Sagan Jul 21 '16 at 17:51



Here is another idea that allow you to affect other elements without considering any specific selector and by only using the `:hover` state of the main element.

7

For this, I will rely on the use of custom properties (CSS variables). As we can read in the [specification](#):



Custom properties are **ordinary properties**, so they can be declared on any element, are resolved **with the normal inheritance** and **cascade rules** ...

The idea is to define custom properties within the main element and use them to style child elements and since these properties are inherited we simply need to change them within the main element on hover.

Here is an example:

```
#container {
  width: 200px;
  height: 30px;
  border: 1px solid var(--c);
  --c:red;
}
#container:hover {
  --c:blue;
}
#container > div {
  width: 30px;
  height: 100%;
  background-color: var(--c);
}
```

```
<div id="container">
  <div>
    </div>
  </div>
```

[Run code snippet](#)[Copy snippet to answer](#)[Expand snippet](#)

## Why this can be better than using specific selector combined with hover?

I can provide at least 2 reasons that make this method a good one to consider:

1. If we have many nested elements that share the same styles, this will avoid us complex selector to target all of them on hover. Using Custom properties, we simply change the value when hovering on the parent element.
2. A custom property can be used to replace a value of any property and also a partial value of it. For example we can define a custom property for a color and we use it within a `border`, `linear-gradient`, `background-color`, `box-shadow` etc. This will avoid us resetting all these properties on hover.

Here is a more complex example:

```
.container {
  --c:red;
  width:400px;
  display:flex;
  border:1px solid var(--c);
  justify-content:space-between;
  padding:5px;
  background:linear-gradient(var(--c),var(--c)) 0 50%/100% 3px no-repeat;
}
.box {
  width:30%;
  background:var(--c);
  box-shadow:0px 0px 5px var(--c);
  position:relative;
}
.box:before {
  content:"A";
  display:block;
  width:15px;
  margin:0 auto;
  height:100%;
  color:var(--c);
  background:#fff;
}

/*Hover*/
.container:hover {
  --c:blue;
}
```

```
<div class="container">
  <div class="box"></div>
  <div class="box"></div>
</div>
```

Run code snippet

Copy snippet to answer

[Expand snippet](#)

As we can see above, we only need **one CSS declaration** in order to change many properties of different elements.

Share Edit Follow Flag

edited Jun 17 '18 at 22:17

answered Jun 3 '18 at 14:01



**Temani Afif**

181k ● 14 ● 166 ● 216