What is the difference between :focus and :active?

Asked 9 years, 10 months ago   Active 6 months ago   Viewed 186k times

What is the difference between the `:focus` and `:active` pseudo-classes?

252

`css`   `css-selectors`   `pseudo-class`

★
59

edited Nov 8 '11 at 7:13   asked Nov 5 '09 at 2:40

BoltClock ♦   Jitendra Vyas
549k   133   1202   **61.7k**   205   514   799
1228

7 Answers

`:focus` and `:active` are two different states.

369

- `:focus` represents the state when the element is currently selected to receive input and
- `:active` represents the state when the element is currently being activated by the user.

✓

For example let's say we have a `<button>`. The `<button>` will not have any state to begin with. It just exists. If we use Tab to give "focus" to the `<button>`, it now enters its `:focus` state. If you then click (or press space), you then make the button enter its (`:active`) state.

On that note, when you click on an element, you give it focus, which also cultivates the illusion that `:focus` and `:active` are the same. **They are not the same.** When clicked the button is in `:focus:active` state.

An example:

```
<style type="text/css">
  button { font-weight: normal: color: black: }
```

```
<button>
  When clicked, my text turns red AND bold!<br />
  But not when focused, where my text just turns red
</button>
```

Run code snippet	Hide results		Full page

When clicked, my text turns red AND bold!
But not when focused, where my text just turns red

edit: jsfiddle

edited Feb 18 at 21:21   answered Nov 5 '09 at 2:51

Juan Castillo   Andrew Moore
3,716 3   30   38   **78.4k** 29 154 169

---

2   Also note that you can get the focused element by using the confusingly named `document.activeElement` property, although it needs to be in a try catch because IE8 can throw an exception. And be aware that older versions of browsers may treat :active and :focus differently. `function activeElement() { try { return document.activeElement; /* can get exeption in IE8 */ } catch(e) { } }` – robocat Jul 26 '12 at 1:51 ✎

---

5   I've created a JSFiddle of your example here: jsfiddle.net/NCwvj Testing in chrome (v24) I've noticed that clicking the button only invokes the `:active` state – Zaki Aziz Feb 19 '13 at 6:41 ✎

---

3   Nit: the order of focus and active affects the state of a HTML button if there is no :active:focus state - Putting :active state after :focus, I get the active changes when I hit tab and hit space. If :focus is last, I never see the active state. – Matt Gaunt Nov 16 '13 at 18:01

    @GauntFace, What browser are you using? On Chrome, the order of your `focus` and `active` declaration doesn't matter. It only matters when they contradict each other e.g. `color:red` and `color:blue` (then last one wins). – Pacerier May 4 '14 at 14:18 ✎

```
:active       Adds a style to an element that is activated
:focus        Adds a style to an element that has keyboard input focus
:hover        Adds a style to an element when you mouse over it
:lang         Adds a style to an element with a specific lang attribute
:link         Adds a style to an unvisited link
:visited      Adds a style to a visited link
```

53

Source: CSS Pseudo-classes

edited Jan 6 '16 at 16:02   answered Nov 5 '09 at 2:50

Stickers   Rubens Farias

50.3k 12   90   117   **50.4k** 5   116   153

6   While related, this doesn't answer the question – Gregor Weber Apr 19 '17 at 14:07

3   @GregorWeber it expose difference by showing definitions – Kamil Kiełczewski Feb 18 at 21:36

You probably know this, but if you define them in that order `visited` links will not change style when hovered or clicked because their "visitedness" will override the other pseudo-classes. LVHFA is the order most people will want to use in most cases. Not sure why you included `lang` ... – Mentalist Jun 20 at 5:46 ✎

There are four cases.

19

1. By default, active and focus are both off.

2. When you *tab* to cycle through *focusable elements*, they will enter `:focus` (without active).

3. When you *click* on a *non-focusable element*, it enters `:active` (without focus).

4. When you *click* on a *focusable element* it enters `:active:focus` (active and focus simultaneously).

Example:

```
<div>
  I cannot be focused.
```

```
    I am focusable.
</div>
```

```
div:focus {
    background: green;
}

div:active {
    color: orange;
}

div:focus:active {
    font-weight: bold;
}
```

When the page loads both are in case 1. When you press tab you will focus the second div and see it exhibit case 2. When you click on the first div you see case 3. When you click the second div, you see case 4.

Whether an element is focusable or not is [another question](#). Most are not by default. But, it's safe to assume `<a>` , `<input>` , `<textarea>` are focusable by default.

edited May 23 '17 at 12:26

Community ♦
1   1

answered Oct 13 '15 at 10:31

James Lawson
5,179   24   34

Thanks for pointing out how elements can have `:active` but not `:focus` . That's one thing I was confused about that the other answers did not address. – B-Stewart Oct 26 '17 at 23:48

5

:focus is when an element is able to accept input - the cursor in a input box or a link that has been tabbed to.

:active is when an element is being activated by a user - the time between when a user presses a mouse button and then releases it.

answered Nov 5 '09 at 2:47

Emily

2   Hi! Is there a state for "currently clicked"? Lets say there are three links in the nav menu, how do you get it to keep a certain color when currently "on"
the link? to show the user where he currently is. as :active only works as long as the link (in this case) is being clicked, but changes back on mouse
button-release. – Kizer Yakuza Aug 11 '13 at 13:55 ✎

0
Active is when the user activating that point (Like mouse clicking, if we use tab from field-to-field there is no sign from active style.
Maybe clicking need more time, just try hold click on that point), focus is happened after the point is activated. Try this :

```
<style type="text/css">
  input { font-weight: normal; color: black; }
  input:focus { color: green; outline: 1px solid green; }
  input:active { color: red; outline: 1px solid red; }
</style>

<input type="text"/>
<input type="text"/>
```

answered Apr 10 '14 at 4:35

Anggie Aziz
98 1 10

-2
Focus can only be given by keyboard input, but an Element can be activated by both, a mouse or a keyboard.

If one would use :focus on a link, the style rule would only apply with pressing a botton on the keyboard.

answered Aug 19 '13 at 18:10

Volker Ackermann
41 2

1   `:focus` doesn't work like that. The textarea I'm typing in currently has focus because I clicked a button. It also can lose and restore focus by clicking
out of it, and into it again. In just a second, I'm going to give focus to the Add Comment button to the right by clicking on it. All of this without keyboard
input causing focus. – Joel Mellon Jul 24 '14 at 21:24

If you can't have the focus on a link, it's simply because a link cannot be focused unless you put it on an anchor, or you add a tabindex attribute. But you
can focus DOM elements like divs or inputs. – Alex Feb 2 '15 at 14:43 ✎

-5

Using "focus" will give keyboard users the same effect that mouse users get when they hover with a mouse. "Active" is needed to get the same effect in Internet Explorer.

The reality is, these states do not work as they should for all users. Not using all three selectors creates accessibility issues for many keyboard-only users who are physically unable to use a mouse. I invite you to take the #nomouse challenge (nomouse dot org).

answered Jun 18 '15 at 23:22

AMG
11   3

1   :hover and :focus are not the same thing. :hover is a specific state and :focus is a specific state. It's a little confusing and misleading to equate them. –
     Garry Polley Aug 27 '15 at 20:31 ✎

1   I'm not doubting your expertise. I'm trying to point out that :hover and :active are not the same thing. And it's confusing for people newer to web and
     accessibility when it's put as simply as :hover is approximately equal to :active for keyboard usage. I appreciate the answer, but maybe a little more
     depth would help? – Garry Polley Aug 31 '15 at 13:36

     Of course they aren't the same thing! I didn't say they were approximately the same. (Please read my posts again.) I'm talking about the difference
     between using a mouse and using a keyboard to accomplish the same task. I'm saying you have to use all of them to give both sets of users the same
     experience. Otherwise keyboard only users who are physically unable to use a mouse will have difficulty in seeing where they have tabbed to on a
     page. This creates an accessibility issue for these folks in navigating the page. – AMG Sep 1 '15 at 18:34 ✎