

Forum:[Hack Forums](#)

Thread: [css card flip on click instead of hover](#)

Mank - October 12, 2020, 3:31 pm

I have this <https://paste.ofcode.org/pazWmmkQURzyhtc2rNnfjx> the hover works fine on my computer but I realized you cannot hover on mobile. when I change it to :active instead it makes me hold it to keep the back part of the card displayed. Is there a way to make it so I just have to click it for it to rotate and click again for it to rotate back to the front?

Disease - October 12, 2020, 6:06 pm

(Oct 12, 2020 03:31 PM)Mank Wrote: [\[CITE\]](#) I have this <https://paste.ofcode.org/pazWmmkQURzyhtc2rNnfjx> the hover works fine on my computer but I realized you cannot hover on mobile. when I change it to :active instead it makes me hold it to keep the back part of the card displayed. Is there a way to make it so I just have to click it for it to rotate and click again for it to rotate back to the front?

You may consider using JavaScript to apply the hover transformation on click with mobile devices. For example:

When a user clicks the card, it'll check to see if the mobile-click class exists. If it doesn't, it adds it. Otherwise it removes it. Thus creating a toggle effect on that class.

You could then apply that with CSS via the likes of:

```
Code .card:hover > .front, .card.mobile-click > .front {}  
.card:hover > .back, .card.mobile-click > .back {}
```

Which effectively just makes the rule applicable only when the mobile-click class exists on a card.

Mank - October 12, 2020, 6:40 pm

(Oct 12, 2020 06:06 PM)Disease Wrote: [\[CITE\]](#) You may consider using JavaScript to apply the hover transformation on click with mobile devices. For example:

When a user clicks the card, it'll check to see if the mobile-click class exists. If it doesn't, it adds it. Otherwise it removes it. Thus creating a toggle effect on that class.

You could then apply that with CSS via the likes of:

```
Code .card:hover > .front, .card.mobile-click > .front {}  
.card:hover > .back, .card.mobile-click > .back {}
```

Which effectively just makes the rule applicable only when the mobile-click class exists on a card.

Thanks, I appreciate it but it did not work

Charger - October 12, 2020, 6:57 pm

(Oct 12, 2020 06:40 PM)Mank Wrote: [\[CITE\]](#) Thanks, I appreciate it but it did not work

Did you put stuff in Code .card:hover > .front, .card.mobile-click > .front {}

```
.card:hover > .back, .card.mobile-click > .back {}
```

Mister Slyther - October 13, 2020, 6:49 am

On mobile, you can just use `:focus`, and `:hover` on phone acts as a click btw.

I would not resort to using Javascript un-necessarily, you don't need to complicate things.

Mank - October 13, 2020, 9:44 am

(Oct 13, 2020 06:49 AM)Mister Slyther Wrote: [\[CITE\]](#) On mobile, you can just use `:focus`, and `:hover` on phone acts as a click btw.

I would not resort to using Javascript un-necessarily, you don't need to complicate things.
Could I pm you please

Disease - October 13, 2020, 2:19 pm

(Oct 13, 2020 06:49 AM)Mister Slyther Wrote: [\[CITE\]](#) On mobile, you can just use `:focus`, and `:hover` on phone acts as a click btw.

I would not resort to using Javascript un-necessarily, you don't need to complicate things.
Using the `:focus` pseudo-class isn't the way to go, as it isn't supported by Safari. It'll work fine in other browsers on mobile, though. But you'll alienate a significant percentage of your mobile users going that route. Same as the `:active` psuedo-class, which in non-Safari browsers would active on tapping an element.

In theory you should be fine with the original code, using just `:hover` pseudo-classes. But the problem is it's unreliable. Often times devices will consider the first tap on an element to be a hover event, and subsequent taps to be clicking. You need to effectively click off the element to reset the event. Hence the JS solution- it eliminates this quirk, and is more reliable and consistent across devices.

Mister Slyther - October 13, 2020, 2:25 pm

(Oct 13, 2020 02:19 PM)Disease Wrote: [\[CITE\]](#) Using the `:focus` pseudo-class isn't the way to go, as it isn't supported by Safari. It'll work fine in other browsers on mobile, though. But you'll alienate a significant percentage of your mobile users going that route. Same as the `:active` psuedo-class, which in non-Safari browsers would active on tapping an element.

In theory you should be fine with the original code, using just `:hover` pseudo-classes. But the problem is it's unreliable. Often times devices will consider the first tap on an element to be a hover event, and subsequent taps to be clicking. You need to effectively click off the element to reset the event. Hence the JS solution- it eliminates this quirk, and is more reliable and consistent across devices.

2 years ago when I took an SCSS course, the instructor actually did a fully functional card flip using CSS only.

I'll be happy to link the source tomorrow.

However, I can't remember if he took what you said into consideration. Will check.

Mank - October 13, 2020, 5:01 pm

Fuck it I got pissed off and just created a button that flips it with javascript, much easier