Accessible tap targets

- Testing your touch targets
- Using media queries to detect touchscreen use

When your design is displayed on a mobile device, you should ensure that interactive elements like buttons or links are large enough, and have enough space around them, to make them easy to press without accidentally overlapping onto other elements. This benefits all users, but is especially helpful for anyone with a motor impairment.

A minimum recommended touch target size is around 48 device independent pixels on a site with a properly set mobile viewport. For example, while an icon may only have a width and height of 24px, you can use additional padding to bring the tap target size up to 48px. The 48x48 pixel area corresponds to around 9mm, which is about the size of a person's finger pad area.

In the demo, I have added padding to all of the links in order to make sure they meet the minimum size.

{% Glitch { id: 'tap-targets', path: 'index.html' } %}

Touch targets should also be spaced about 8 pixels apart, both horizontally and vertically, so that a user's finger pressing on one tap target does not inadvertently touch another tap target.

Testing your touch targets

If your target is text and you have used relative values such as `em` or `rem` to size the text and any padding, you can use DevTools to check that the computed value of that area is large enough. In the example below I am using `em` for my text and padding.

{% Glitch { id: 'tap-targets-2', path: 'style.css' } %}

Inspect the `a` of the link, and in Chrome DevTools switch to the Computed pane where you can inspect the various parts of the box and see what pixel size they resolve to. In Firefox DevTools there is a Layout Panel. In that Panel you get the actual size of the inspected element.

{% Img src="image/admin/vmFzREveRttHVDfLqqCx.jpg", alt="The Layout Panel in Firefox DevTools showing the size of the a element", width="800", height="565" %}

Using media queries to detect touchscreen use

Instead of simply testing for viewport dimensions, and then guessing that small dimensions are likely to be phones or tablets, which in turn use a touchscreen, there are now more robust ways to adapt your design based on actual device capabilities.

One of the media features we can now test for with media queries is whether the user's *primary input* is a touchscreen (`pointer`) and whether *any* of the currently detected inputs is a touchscreen (`any-pointer`). The `pointer` and `any-pointer` features will return `fine` or `coarse`. A fine pointer will be someone using a mouse or trackpad, even if that mouse is connected via Bluetooth to a phone. A `coarse` pointer indicates a touchscreen, which could be a mobile device but may also be a laptop screen or large tablet.

If you are adjusting your CSS within a media query to increase the touch target, testing for a coarse pointer allows you to increase the tap targets for all touchscreen users. This gives the larger tap area whether the device is a phone or larger device.

```
.container a {
  padding: .2em;
}

@media (any-pointer: coarse) {
  .container a {
    padding: .8em;
  }
}
```

You can find out more about interaction media features such as pointer in the Responsive web design basics article.

□□□ #2
□ □□□ □□□ 2026-10-31 21:32:50 UTC
□ □□□ □□□ 2026-10-31 21:32:52 UTC