Size basis for focus appearance

This is a discussion document regarding how the size of a component should be calculated for focus-appearance.

The two options being discussed are:

- Based on the concept of the ‘component’ (current text);
- Based on the target size of the component.

Relevant normative text

Focus appearance: When a component receives keyboard focus, an area of the focus indicator meets the following:

- **Minimum area**: The area is either:
    - at least as large as the area of a 1 CSS pixel thick perimeter of the unfocused component, or
    - at least as large as the area of a 4 CSS pixel thick line along the shortest side of a minimum bounding box of the unfocused component, and no part of the area is thinner than 2 CSS pixels.
- [...]

Note: In HTML the size of a component is measured up to and including the CSS border.

Target: region of the display that will accept a pointer action, such as the interactive area of a user interface component

Argument for basing it on component

The SC text currently bases its scope on the size of the component, which means:

- For the vast majority of components it is obvious, it just works. Open up the browser inspector, check the size, done.
- We can provide tech-specific guidance in the understanding document to enable consistent testing, e.g.
    - In HTML the size of a component is measured up to and including the CSS border.
    - In PDF and ePub the focus indicator is not under author control.
    - …
- Basing it on a measure for mouse-use would create a disconnect. For example, it could incentivise people to reduce the hit area so the focus indicator is not as large. (See card and checkbox examples below.)

The confusion seems to have been mostly caused by the current note which talks about the visual size. We would like to update the note to something like:

"*In technologies where it is not possible to interrogate the component size the minimum area can be taken from a bounding box around the visible content of the component.*"

"

Difficult examples

https://codepen.io/wilcofiers/full/OJjrddm (NB: AlastairC & MichaelG didn't consider those difficult, starting with browser tools found the size, it would help to add some explanations here, maybe copying from Michael's doc)

Argument for basing it on target size

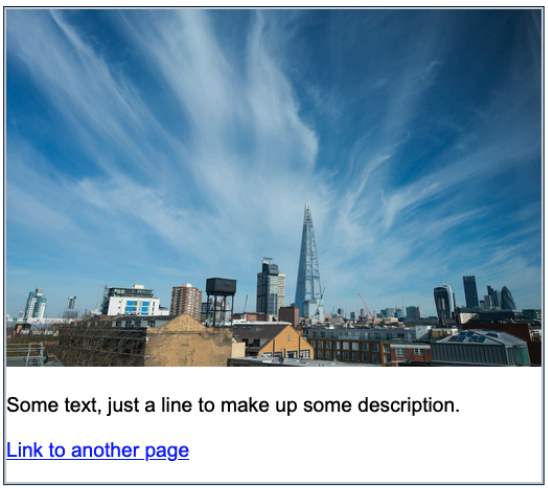
The SC does not state how to decide the shape of the component. Yes, it could be the border-box. It could also be the visual appearance, it could be the smallest of the two, it could be the unclipped area of the border-box, it could be the target. It can be many, many things.

Basing it on target size would mean:

- It's technology agnostic
- for most technologies it can be programmatically determined
- it matches the border-box / browser indicated size in HTML and SVG (unless clipped, which is very rare)
- It doesn't require this problematic "visible area" thing as a fallback

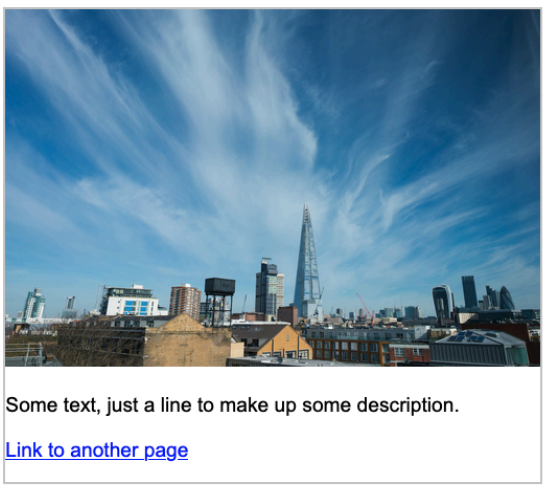
Difficult examples

A "card" such as this on a test page:



Some text, just a line to make up some description.

Link to another page

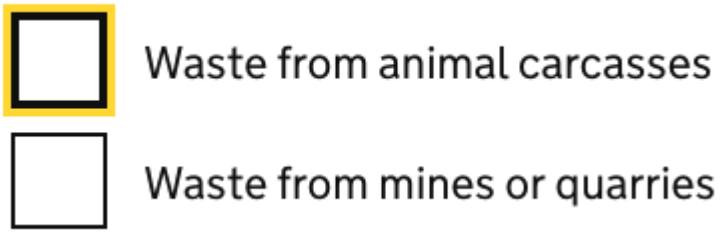


Some text, just a line to make up some description.

Link to another page

It can be implemented with the blue text being a link and javascript expanding the hit area, or the whole thing could be wrapped in a link. Basing it on the focusable component is easier to understand.

A small form control with a large label, e.g. checkbox



Properly associating the label makes the hit-area for the checkbox several times the size of the checkbox. Requiring a very large indicator would be unexpected (and unnecessary) in this scenario.