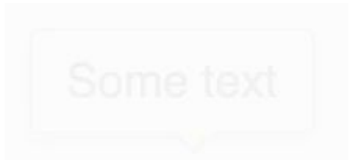
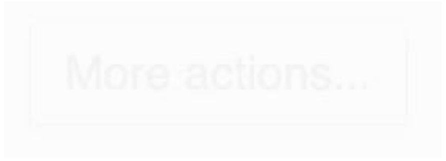


Tooltips, Hovercards, Menus, etc.



Mason Freed

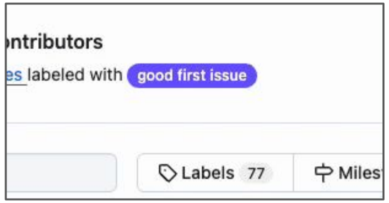
TPAC 2024
September 23-27, 2024

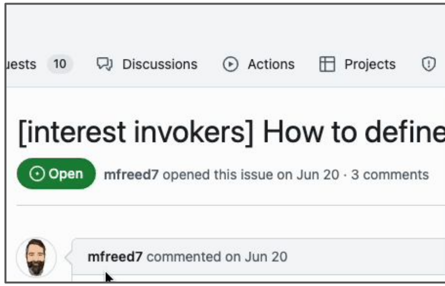
Outline

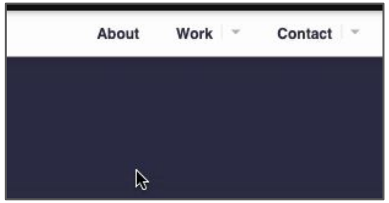
- Use cases we’re interested in solving
- Examples from production
- Required APIs
- Interest target API details

Use Cases To Solve

- "Tooltips" (or "plain hints")
    - Contain **auxiliary** information, not "required" for the user to see.
    - Does not contain interactive or semantically interesting (e.g. table) content.
    - Often used to remove non-critical information from information-dense pages.
- "Hovercards" (or "rich hints")
    - Also limited to **auxiliary** information.
    - Can contain more interesting content, including **interactive components**.
    - Often used to remove non-critical information from information-dense pages.
- "Hover menus"
    - A **menu** that is activated on hover.
    - Also always activated on *activation* (click, touch, Enter key, etc.)



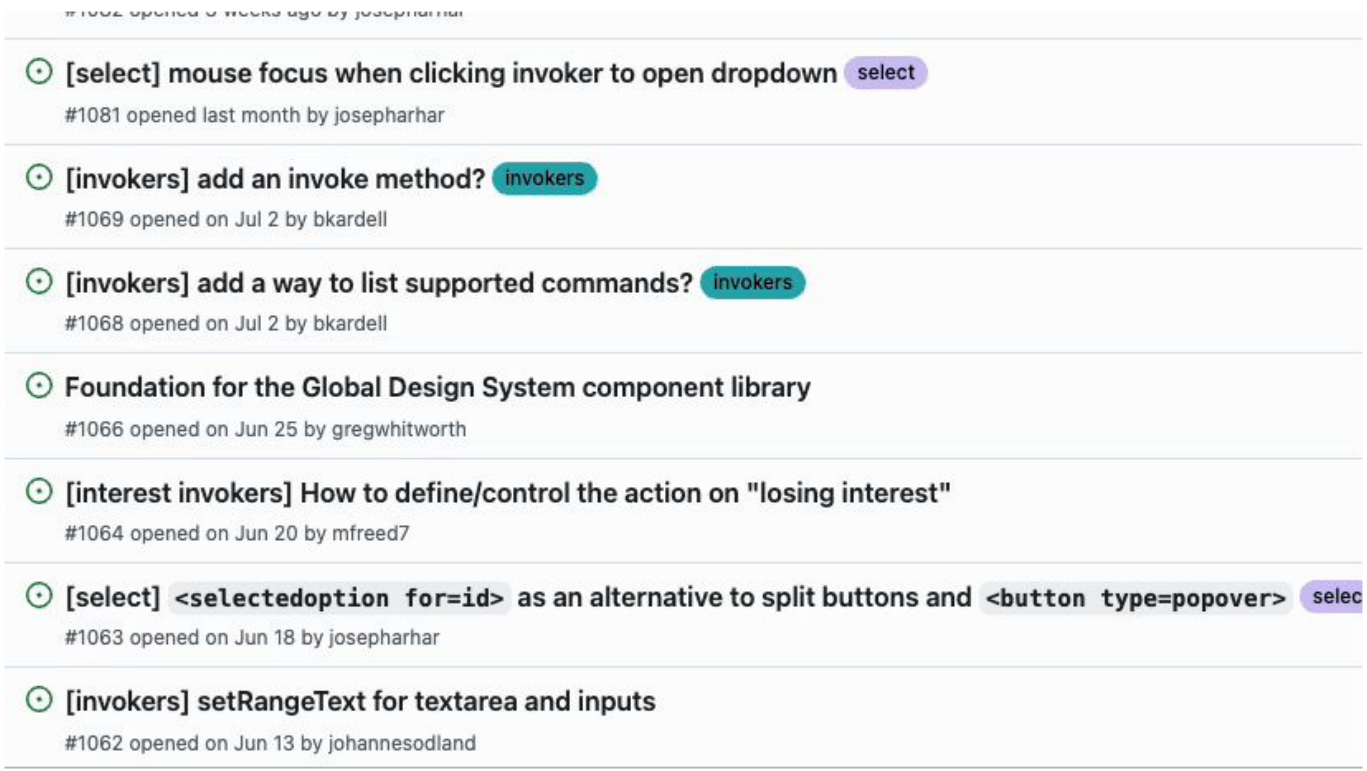

uests 10
Discussions
Actions
Projects
[interest invokers] How to define
Open
mfreed7 opened this issue on Jun 20 · 3 comments
mfreed7 commented on Jun 20



Key requirements

- The API solves the very common use cases on the prior slide.
- Declarative solution - no JS needed.
- Accessibility built-in - no ARIA needed.
- Works for non-desktop and non-mouse systems.

Examples in production - Github hovercards



[select] mouse focus when clicking invoker to open dropdown select
#1081 opened last month by josepharhar

[invokers] add an invoke method? invokers
#1069 opened on Jul 2 by bkardell

[invokers] add a way to list supported commands? invokers
#1068 opened on Jul 2 by bkardell

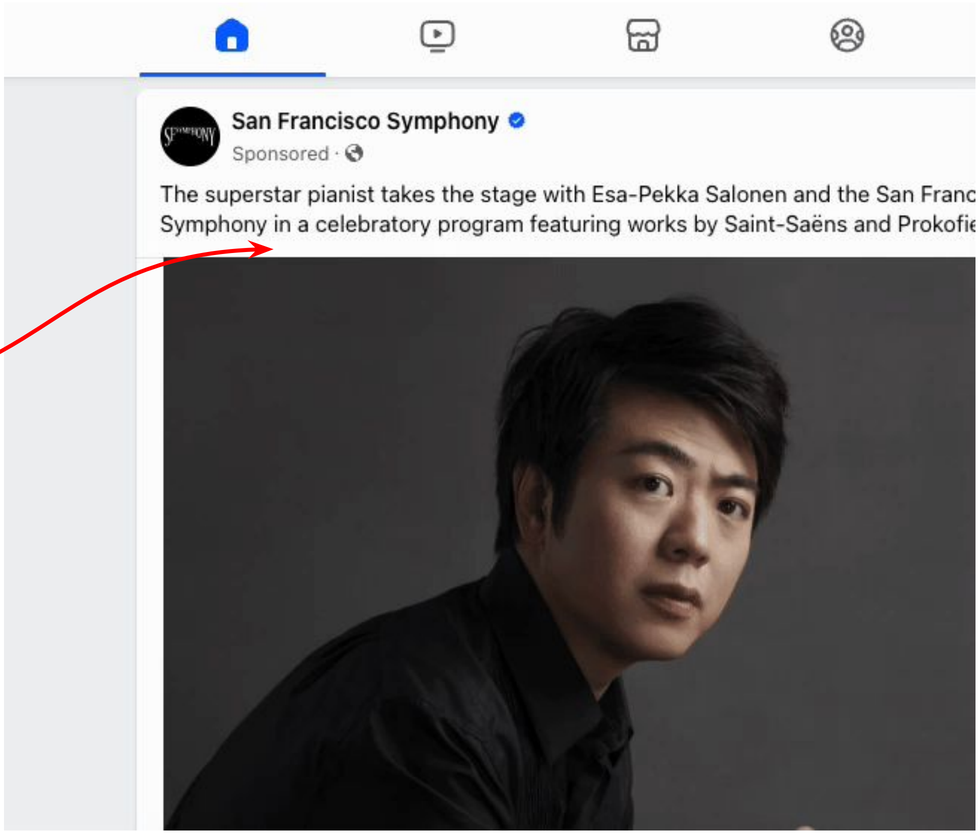
Foundation for the Global Design System component library
#1066 opened on Jun 25 by gregwhitworth

[interest invokers] How to define/control the action on "losing interest"
#1064 opened on Jun 20 by mfreed7

[select] `<selectedoption for=id>` as an alternative to split buttons and `<button type=popover>` selec
#1063 opened on Jun 18 by josepharhar

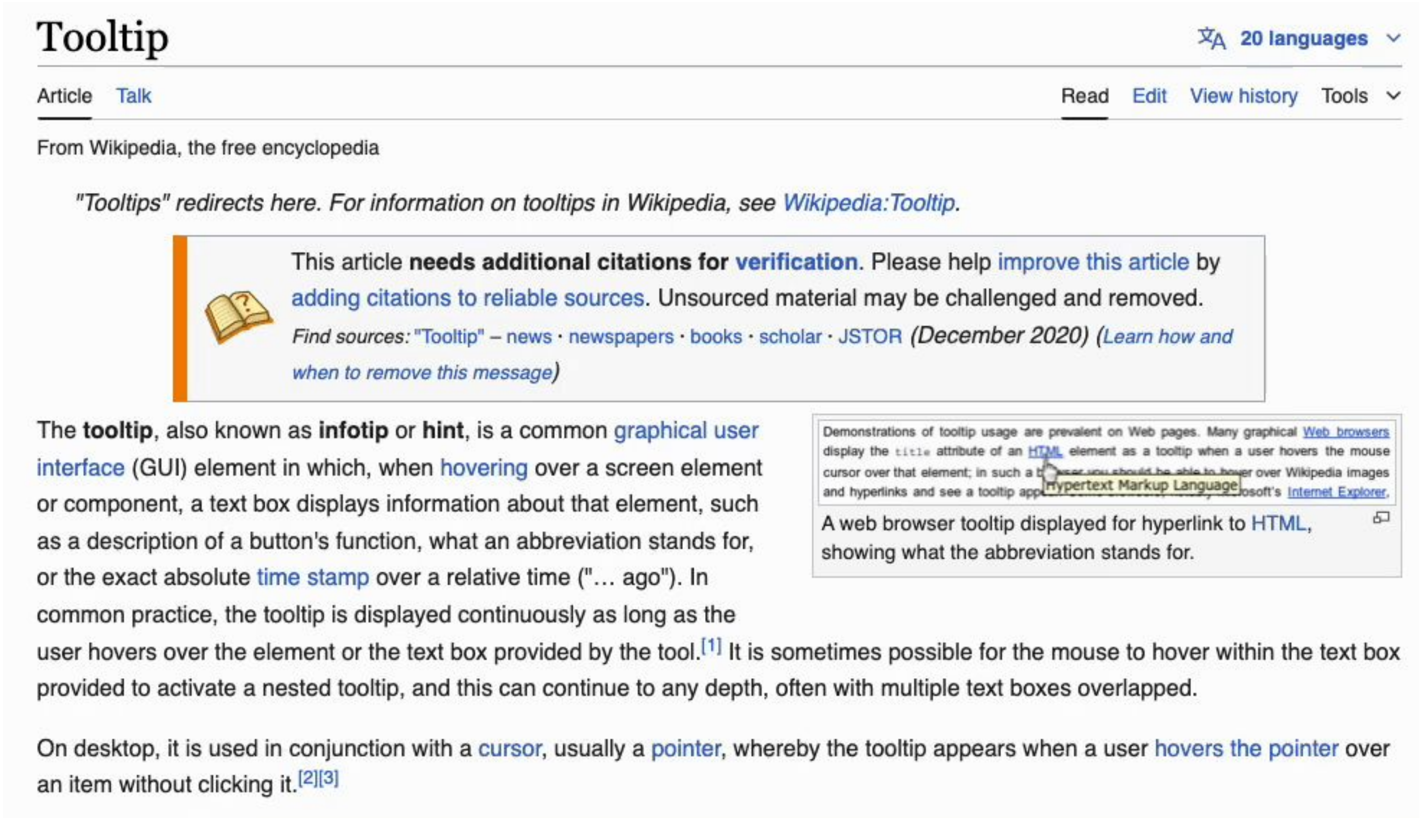
[invokers] setRangeText for textarea and inputs
#1062 opened on Jun 13 by johannesodland

Examples in production - Facebook hovercard and tooltip


San Francisco Symphony
Sponsored ·
The superstar pianist takes the stage with Esa-Pekka Salonen and the San Franc
Symphony in a celebratory program featuring works by Saint-Saëns and Prokofie

Note the nested tooltip, which also has a `title` double-tooltip

Examples in production - Wikipedia hovercard



Tooltip

20 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

"Tooltips" redirects here. For information on tooltips in Wikipedia, see Wikipedia:Tooltip.

This article **needs additional citations for verification**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.
Find sources: "Tooltip" – news · newspapers · books · scholar · JSTOR *(December 2020) (Learn how and when to remove this message)*

The **tooltip**, also known as **infotip** or **hint**, is a common graphical user interface (GUI) element in which, when hovering over a screen element or component, a text box displays information about that element, such as a description of a button's function, what an abbreviation stands for, or the exact absolute time stamp over a relative time ("… ago"). In common practice, the tooltip is displayed continuously as long as the user hovers over the element or the text box provided by the tool.[1] It is sometimes possible for the mouse to hover within the text box provided to activate a nested tooltip, and this can continue to any depth, often with multiple text boxes overlapped.

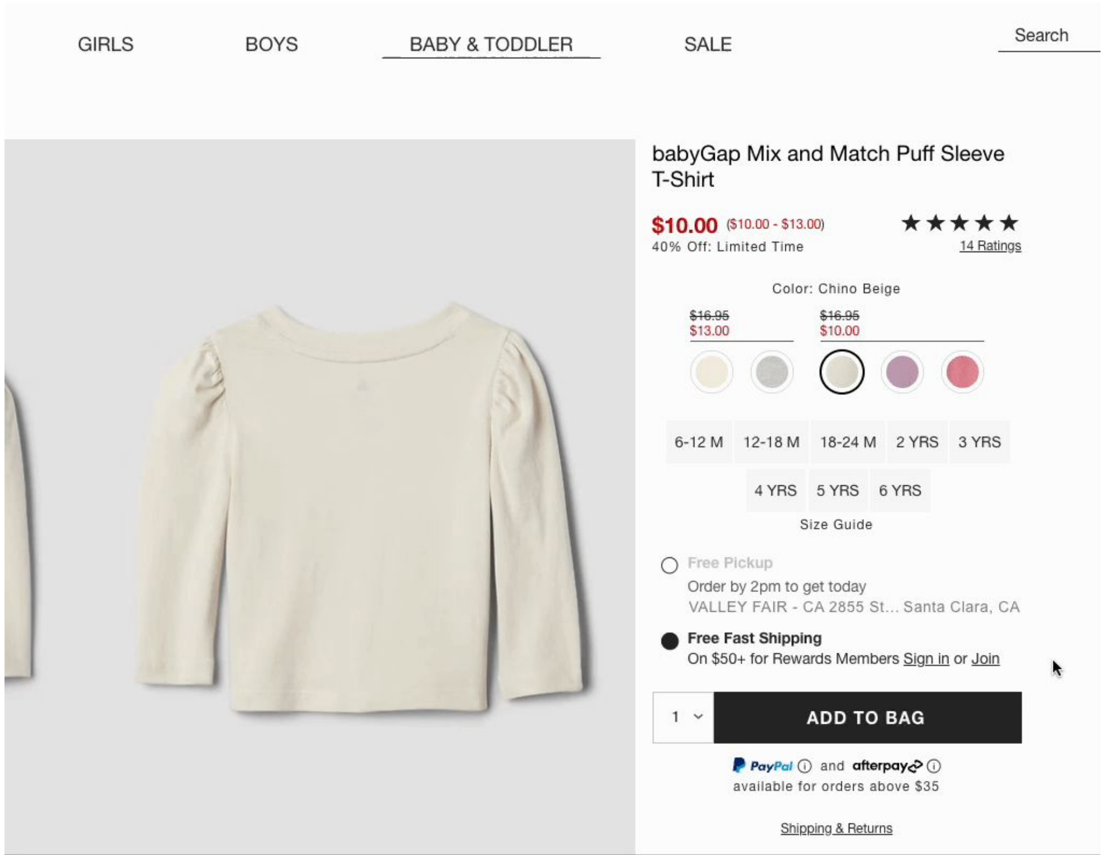
Demonstrations of tooltip usage are prevalent on Web pages. Many graphical Web browsers display the `title` attribute of an HTML element as a tooltip when a user hovers the mouse cursor over that element; in such a b[illegible]er over Wikipedia images and hyperlinks and see a tooltip app[illegible]osoft's Internet Explorer,

Hypertext Markup Language

A web browser tooltip displayed for hyperlink to HTML, showing what the abbreviation stands for.

On desktop, it is used in conjunction with a cursor, usually a pointer, whereby the tooltip appears when a user hovers the pointer over an item without clicking it.[2][3]

Examples in production - Gap hover menu



babyGap Mix and Match Puff Sleeve T-Shirt

$10.00 ($10.00 - $13.00)
40% Off: Limited Time

★★★★★ 14 Ratings

Color: Chino Beige

~~$16.95~~ $13.00 ~~$16.95~~ $10.00

Free Pickup
Order by 2pm to get today
VALLEY FAIR - CA 2855 St... Santa Clara, CA

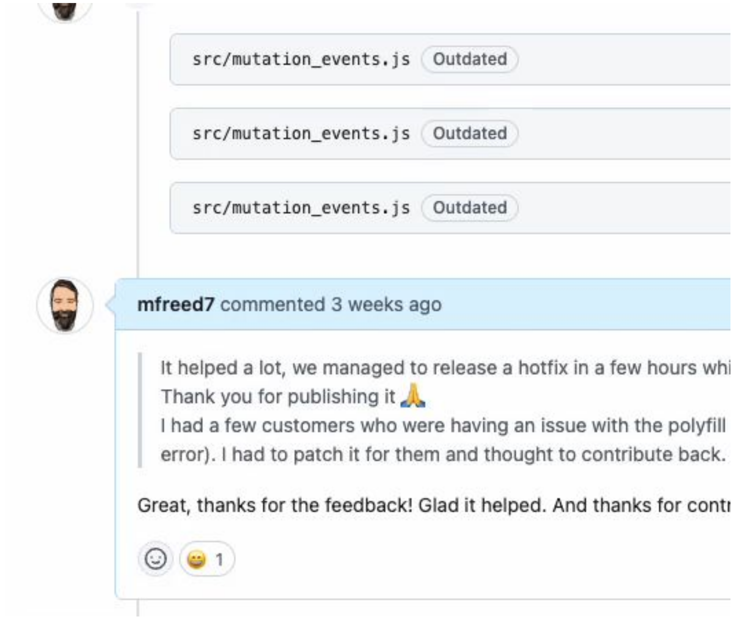
Free Fast Shipping
On $50+ for Rewards Members Sign in or Join

PayPal and afterpay available for orders above $35

Shipping & Returns

Quick note on “auxiliary”

- Most design systems say that hovercard/tooltip content must be “auxiliary”, meaning not required to be seen to accomplish a task.
- However, “auxiliary” is a grey area.
- E.g. for me, to fully understand a Github comment, I **need** to know who the commenter is.
- Several production sites said that their **engagement metrics were negatively impacted** by reducing access to hovercard content.



src/mutation_events.js Outdated

src/mutation_events.js Outdated

src/mutation_events.js Outdated

mfreed7 commented 3 weeks ago

> It helped a lot, we managed to release a hotfix in a few hours whi
> Thank you for publishing it 🙏
> I had a few customers who were having an issue with the polyfill
> error). I had to patch it for them and thought to contribute back.

Great, thanks for the feedback! Glad it helped. And thanks for contri

😄 1

Features needed to build a hovercard

Popover API

1. The hovercard displays on top of page content.

Anchor Positioning

2. The hovercard is positioned next to the element it explains.

popover=hint

3. The hovercard should light-dismiss, but should not dismiss other normal popovers like select pickers.
4. The user triggers the hovercard by hovering with a mouse, focusing with the keyboard, or long-pressing on a touchscreen.
5. Proper a11y connections need to be made, e.g. aria-expanded, aria-details, potentially aria-describedby, etc.

interest invokers

Interesttarget declarative API proposal

```
<a href=foo interesttarget=card>interesting link</a>

<div popover id=card>Hovercard</div>
```

- The popover is shown when “interest is shown” in the link.
- The popover is closed when “interest is lost” in the link.
- ⇒ The same pattern can work for hovercards, tooltips, and menus.

Input modalities

- By far the most difficult part of this API (both for web standards **and** for design system developers) is handling all of the input modalities:
    - **Nearly all** design systems handle **mouse**-activation via hover.
    - **Most** design systems try to provide good **screen reader** support, with varying success.
    - **Some** design systems build affordances for **keyboard**, again with varying success.
    - **Almost no** design system builds **touch screen** support.
    - **Almost no** design system builds support for “**other**” input types.

Mouse

- **Hover** is the standard way to do this, universally implemented in design systems.
- Some questions around the edges:
  - Losing interest happens when the element or the target element are de-hovered.
  - Controlling delays for show and hide
  - “Safe triangles”
  - Second-popover zero-delay
- But overall, this is “solved” and roughly standard.

Keyboard

- Methods used by design systems:
  - Show hovercard as soon as the element is focused.
  - Show hovercard when focused, but after a delay.
  - Add a focusable icon (ℹ️) next to the element, which can be keyboard-activated. This is typically only used in very special circumstances, such as the “CVV” field of credit card forms.
  - Use a special hotkey, such as ALT-Up Arrow, to activate the hovercard.
- Issues:
  - Users tend to find the **pure-focus** based activation annoying and distracting, since it interferes with normal keyboard navigation of the site.
  - The **focusable icon** (ℹ️) approach similarly adds both visual clutter and extra tab stops that users and developers do not like.
  - The **hotkey** approach works, and is used in some design systems, but it lacks **discoverability**.

Keyboard - ideas (add yours!)

- **UA provides a special hotkey** to “show interest” via the keyboard.
- Discoverability: focusing the element shows a UA-provided tooltip that informs the user about the hotkey.
- (Alternative idea next slide.)
- Losing interest happens via ESC.

This is a link with interesttarget

Side-bar: hotkey “cheat sheet”

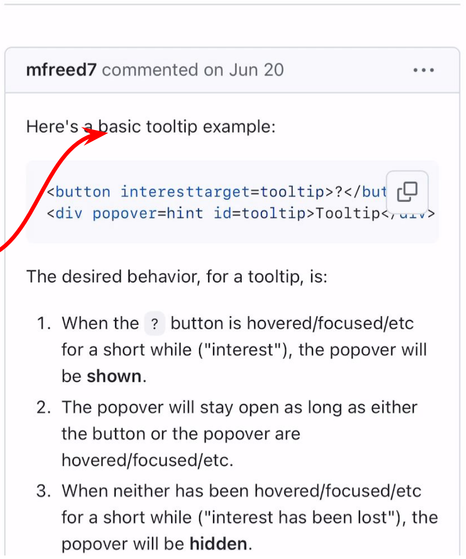
- The most common keyboard pattern for hovercards seems to be a hot-key.
- There are a few issues with hot-keys:
    - They are not easily discoverable
    - It is hard to ensure uniqueness and avoid “collisions”.
- Idea: what if there was a browser-standardized way to…
    - Give users an easy way to see all hot-keys for a **page** and for any **element**? I.e. a “cheat sheet” for hot-keys. This would include browser-provided hot-keys (such as the Tab key) and developer-provided hot-keys (e.g. from `accesskey`).
    - Give developers a way to declare all of their hotkeys for a page/element, which ties in to the above “cheat sheet”? More than what’s available from `accesskey`.
    - Perhaps provide a way to let the browser select from alternative hot-keys when collisions occur.
- This requires more work, but perhaps it alleviates many problems in addition to the hovercard activation problem.

Touch screen (biggest open question)

- Potential methods (no actual implementations that we could find):
  - Fake long-press via `touchstart`, `touchend`, and tricks like `-webkit-touch-callout: none`.
  - Add a focusable icon (ℹ️) next to the element, which can be keyboard-activated.
- Issues:
  - “True” long-press support is commonly requested. It is **not easily implementable via existing web APIs**. It is available and commonly-used on **native apps**.
  - Almost no design system supports touch screen activation of hovercards at all, due to the lack of an API.

Touch screen - ideas (add yours!)

- add an item to the UA-provided long-press menu that provides hovercard activation (or the **hovercard itself?**), here:
- If no context menu would have been shown by long-press, simply directly trigger the popover.
- Another option: **first** show the developer-provided hovercard, and **then** provide the user an extra-tap way to get back to the context menu.
- Losing interest happens via tapping outside the popover.



mfreed7 commented on Jun 20

Here's a basic tooltip example:

```
<button interesttarget=tooltip>?</but
<div popover=hint id=tooltip>Tooltip</div>
```

The desired behavior, for a tooltip, is:

1. When the `?` button is hovered/focused/etc for a short while ("interest"), the popover will be **shown**.
2. The popover will stay open as long as either the button or the popover are hovered/focused/etc.
3. When neither has been hovered/focused/etc for a short while ("interest has been lost"), the popover will be **hidden**.

Other Input Modalities

- Examples:
  - Playstation
  - Vision Pro
  - Watch face (touch?)
- No design system we could find supports these explicitly.
- By virtue of their uniqueness and novelty, standardizing exact solutions for these interfaces seems tricky.
- Proposal: leave these up to the UA. If there’s a way to add an affordance for activating the hovercard, do it. If it doesn’t make sense, rely on hovercards being “auxiliary”.

Conclusion

- This API (interesttarget) promises to solve several very common use cases on the web, including tooltips, hovercards, and hover-menus.
- The mechanics for developers should be very simple, and should remove the need to re-invent hovercard activation for each design system, and on each platform.
- The primary open questions are around the specifics for keyboard and touchscreen activation.