

# WHATNOT meeting notes

# WHATNOT meeting notes

**Web Hypertext Application Technology Navigating Open Topics**

<https://meet.google.com/hfm-ttah-hfh>

## Meeting times (alternating weekly)

*Be sure to update these when a Daylight Savings Time boundary is crossed!*

- [18:00 Berlin / 01:00 Tokyo / 09:00 San Francisco](#): AMER + EMEA friendly
- [01:00 Berlin / 08:00 Tokyo / 16:00 San Francisco](#): APAC + AMER friendly
- [10:00 Berlin / 17:00 Tokyo / 01:00 San Francisco](#): EMEA + APAC friendly

Please RSVP to your calendar invite, as instances with low attendance might get canceled!

## Open issues

- [Open "agenda+" issues across all WHATWG repositories](#)
- [Open "agenda+" PRs across all WHATWG repositories](#)

## Template

2025-01-01 ( AMER + EMEA ▾ )

Tracking issue: [https://github.com/whatwg/html/issues/ADD\\_ISSUE\\_NUMBER](https://github.com/whatwg/html/issues/ADD_ISSUE_NUMBER)

### Attendees

Person 1, Person 2, Person 3 (try @-ing people)

### Agenda

1. Review past action items
  - a. ...
  - b. ...
2. Carryovers
  - a. ...
  - b. ...
3. New topics
  - a. ...
  - b. ...

## Minutes

...

## Meetings

2025-10-02 ( EMEA + APAC ▾ )

Tracking issue: <https://github.com/whatwg/html/issues/11696>

### Attendees

Person 1, Person 2, Person 3 (try @-ing people)

### Agenda

1. Announcements
  - a. Scribing
2. Review past action items
  - a. Chris to make meeting time poll
  - b. Stephen Chenney will get started on writing up the spec PR for adding an attribute to the canvas rendering context for writing mode and figure out how many sites are making use of the CSS.
    - i. Carry over
3. Carryovers
  - a. [luke? bkardell] [Consider renaming HTMLOrSVGELEMENT to HTMLOrForeignElement #4702](#)
    - i. carryover
  - b. [zcorpan] [Redact ancestorOrigins using referrer policy](#)
    - i. carryover
4. New topics
  - a.

## Minutes

2025-09-25 ( AMER + EMEA ▾ )

Tracking issue: <https://github.com/whatwg/html/issues/11696>

### Attendees

Person 1, Person 2, Person 3 (try @-ing people)

## Agenda

5. Announcements
  - a. Scribing
  - b. [Chris] WHATUP at TPAC: <https://github.com/whatwg/html/issues/11711>
6. Review past action items
  - a. Fernando Fiori will file an issue to get feedback in the working group and continue with the existing behavior for focus without user activation if there is no feedback.
    - i. done
  - b. Stephen Chenney will get started on writing up the spec PR for adding an attribute to the canvas rendering context for writing mode and figure out how many sites are making use of the CSS.
    - i. Carry over
  - c. Kurt Catti-Schmidt will follow up separately regarding the adopted stylesheets attribute and put it back on the agenda if it needs to be fleshed out more.
    - i. done
  - d. Jacques Newman will add a backlink from the explainer to the HTML GitHub repo for the focus group proposal.
    - i. done
7. Carryovers
  - a. [Iwarlow] [Add missing IDREF reflections](#)
    - i. Follow ups by Luke
8. New topics
  - a. [Kurt] [#10673](#) (Declarative CSS Modules and Declarative Shadow DOM adoptedstylesheets attribute) to discuss getting it to Stage 2 now that a [PR](#) is ready.
    - i. AI: Olli to review, all to take a look, bring back next time.
  - b. [ffiori] [\[focus-without-user-activation\] Allow focus if a descendant has focus](#)
    - i. Needs some re-framing of problem.
  - c. [luke? bcardell] [Consider renaming HTMLOrSVGELEMENT to HTMLOrForeignElement #4702](#)
    - i. carryover
  - d. [smaug, all] meeting times?
    - i. AI Chris to make meeting time poll
  - e. [zcorpan] [Redact ancestorOrigins using referrer policy](#)
    - i. carryover

## Minutes

whatup:

simon: we need an agenda+ label for the f2f

chris: yes

anne: add the label through the meta repo, then run the script which will add it to all the repos

add missing idref reflections:

luke: we have for aria and for commandfor and popovertarget, the ability to set the targeted element from the equivalent of the idref attribute from js. that goes through element reflection. it was added for aria, and the new ones have them (commandfor and popovertarget), but we haven't added them for html content attributes, like label element for. input element's list, and various form elements which you can associate with a form. i raised an issue to say, should we add those missing ones? seemed like a good idea to add these. the question now is kind of on the shape of that and the exact behavior. i've initially proposed just adding new idl attributes that just do this specific thing, but it has been pointed out that there are existing properties for some or all of these that don't quite do what we want but we could make them do it. there were questions about validation that these properties do. if we take label.control, that will only take labelable element, which currently i think the type is HTMLFormControlelement, it would have to be raw htmlelement because of referencetarget, but we could make sure it only returns labelable elements if we wanted to. do people have thoughts? any objections?

mason: it sounds like a good idea, and i see the validation question, were there any other concerns raised?

luke: should we add a new form element or just keep control and make that writable and do the reflection? benefit of control is that we don't need to add a new property, but hard to feature detect if control is writable. we would need to add behavior to them so i don't know.

anne: has there been developer demand for these?

luke: for the label for element one, yes i've had people online mention it. for some of the others, i don't know. output's for attribute seems pointless.

anne: the demand is for the setter?

luke: yes, to be able to set it to an element via js to get the same behavior as popovertargetelement.

keith: for web compat, you can see that the setter is undefined, so it should be trivial for feature detection.

mason: on the validation question, i don't think any of the other element reflection attributes do any validation. popovertarget can point to anything. is that the right thing to follow?

keith: for shadowroots you can set it and get null back?

mason: i'm talking about popovertargetelement and the target is not a popover. i thought that was a question for labelable things

luke: popovertarget is the precedent for do nothing, but maybe that should not return an element. for commandfor there's not really validation you can do

mason: if you wanted you could check the command and see if it's the right one, but seems cleaner to just return the element

keith: if someone were to author the steps in the wrong order, that would fail. you could set it to a div and then add popover to it. idk if elements can become labelable

luke: custom elements can become registered and become form associated.

keith: that sounds like a real footgun if there's timing issues around CE registration for the validation

mason: i agree

simon: on the question of whether to introduce a new property, i think we should optimize for the long term ergonomics rather than feature detection. if feature detection is possible, then that's

good enough. idk which is actually better for web devs, but in general we try to avoid duplicating functionality.

keith: on one hand, if you already know controls exists, then it becomes settable, but it lacks the symmetry with the others that are suffixed element. for label its not forElement, its control. which is the worse tradeoff? i have my opinion

luke: if we decide that reflectors dont do the validation, if we decide that it should do the same validation, then yes i think the answer will be use the existing one. with .control how does that work with custom elements today?

keith: if you set a for element, you can figure out if it was applied by looking at label.control. those are useful properties which justify their existence.

simon: yep, seems fair

mason: seems like there are 3 questions: should we do this, should we validate, and what should the names be?

keith: is this label one the only controversial?

luke: input list and ?? all have the behavior of is this valid. output for i dont think does, but doesn't do anything. other one is the table header, something about table in the headers attribute, but that was added for completeness

keith: that begs the question, what are the use cases for this, maybe broken out into each one. we dont have to reflect every single one of these. currently theyre not all reflected. if we do some of them then that doesn't really change the state of play, but if we do all of them is that really beneficial to developers? maybe we should break them out into separate issues, and the less controversial ones can land more easily.

luke: im happy to separate label for.

anne: it doesn't seem like theres that much demand

luke: i could ask on social media but then people will just say yes

simon: upvotes is a property of has this been shared

luke: ill separate into separate issues and go from there

declarative css modules:

kurt: i have a pr ready, would appreciate some feedback, its my first PR in html. reading the stage 2 requirements, i think it meets that but is there more requirements that need to be met?

anne: i havent had a chance to review yet, but it does seem like the pr doesn't build? theres a red x at the bottom. its the conformance checker

kurt: other than that, if someone wants to take a look

chris wilson: to focus in on stage question, do we have consensus on stage 2? the difference is that there is a draft spec, and consensus that the rough api shape is the right thing to solve the problem.

olli: i think i want to take alook at how nested shadowdoms are handled. or do we end up leaking css modules from inner ones to others or vice versa?

kurt: that is how the module system works, so it would

chris wilson: olli, can you review next time and come back to the stage question?

anne: were domenics comments resolved?

kurt: yes, domenic left in support of this

anne: initially he had some issues with how module mappings work

kurt: he said that is resolved with this new approach

anne: reading his comment, he wanted also json in js modules to be solved with the same kind of system. are you sure that all of that is resolved by this? it seems like there were some larger asks

kurt: anything serializable to a data url or import map ?? includes those, that got brought up and got resolved as the bigger approach

anne: thanks

allow focus if descendant has focus:

fernando: during the last meeting domenec raised a point here, related to permissions policy user activation. i am trying to write the spec for the last resolution at tpac, allowing parent frame to set focus in iframe. he was worried that we were going with loose behavior and if there was dev support for this. i wrote a comment in the pr explaining this, i talked to our customers, and if we go with a strict behavior of not allowing the child iframe which has the policy disabled to use the focus apis, that would break many experiences. that supports the case that this pr is making, of allowing the child iframe even with the policy disabled, once its got focus it can keep moving it around inside itself. i put some screenshots from teams and outlook. \*shares screen to show examples\*

anne: in the case of a embeds b embeds c, if ? is focused then b is allowed to focus arbitrary things?

fernando: yes, anything inside b or iframe that b is hosting

simon: but b could set a policy to disallow c from stealing focus, and then isolation would be within c

olli: can b then steal stuff which was meant for c? and then move focus to c and then steal focus from c?

fernando: yes, because b had the power to move focus around, so it sounds like it should be able to. idk what the concern would be there

anne: as far as i understand from the pr, this change doesn't impact the permission at all. this sidesteps the permission. if we have a case of even if its just a embeds b embeds c, then if c has focus through the user, then if b has some loop where it tries to call element.focus on one of its things, it will now succeed where it would previously not. and maybe thats ok, but it seems a little weird

olli: i think we have had a security issue with this, but it might have been with the browser ui and the webpage, so you could steal input for the browser ui, but this is similarish issue

taylore: this would now let b succeed, but b was always able to steal focus. a can steal from b or c when its in focus. \*shares screen to show example\*

anne: is it true that with the current thing a can steal focus? it could block itself in which case it wouldn't be able to steal focus. that would be weird because then a couldn't steal it but then b could?

taylore: if a was disabled a b was allow self

anne: a could no longer disable itself because the initial check doesn't return true but the descendant check returns true, it ends up disabling part of the feature of giving people control over this.

taylore: if those were switched, then b - allow self applies to everything in the same origin. maybe that would cause some issues.

anne: if a sets it to none, i forgot if it propagates to b

fernando: yes, b cannot override that

anne: even though it gets disabled all the way through, it doesnt matter because the first check won't go through but the third will, so you cant actually disable it anymore with this proposed change

taylore: allow self lets you move focus on any descendant.

anne: the first step of the allow focus algorithms checks allowed to use, but that check no longer ends up mattering because of the inclusive descendant check

fernando: but the inclusive descendant check has focus

anne: it matters if one of them does have focus. previously it would return false in that case because the policy did not allow for it, and now it would return true

fernando: yes, thats intended. in that case, the reason is that if you have focus or your descendants have focus, then you have the power to move focus around. teams wants to give the app the freedom to move itself

anne: if teams want it, then cant they just delegate this permission?

fernando: they dont want to the frame to steal focus when they dont explicitly pass it to the frame. for example, in the case of a b c, they dont want b or c to steal focus, perhaps while the app is loading then the user focuses on the search bar for teams, maybe b wants to focus, they want to disallow that, so when the user wants to load the app, then they pass the focus and b can do whatever it wants with focus.

taylore: i think we need to think through that allow none example, but the intended behavior is that something beneath something that is allow self - anything at the allow self level is allowed to set focus, and anything is allow none then it has a conditional behavior where it can set focus but only if its been focused manually by clicking it, and thats intended. our main question is, is this behavior even though its odd is it suitable for a policy? or we should change the name? but we do believe that the intended behavior is what we want.

chris wilson: it seems like we need some more investigation here, and some convincing of the problem i guess?

anne: its not clear to me that if b has focus then c can always steal it. you could imagine that b could be an ad and c could be a point of sale, and you dont want b to be able to steal focus.

fernando: yes, you have to trust the inner app

taylore: this is how things currently work, right?

leo: maybe we can better illustrate how this works today and point out the delta?

anne: current behavior meaning current implementations that dont follow the spec?

taylore: whats possible today, and how they set programmatic focus

anne: that would vary across implementations

meeting times:

olli: now that domenico wont be joining, should we have different times? usually US-japan friendly zone its me and chromium folks, and its a bit late time for me, and theres the other one

chris wilson: i asked the question before about whether we want to keep the one thats later than this for everyone meeting, i think the general consensus is that we want to keep the one thats



earlier than this, eu-japan. maybe i should just set a poll and see if we want to move down to just doing the two different timezones.

2025-09-18 ( APAC + AMER ▾ )

Tracking issue: <https://github.com/whatwg/html/issues/11648>

## Attendees

Chris Wilson , Alex Keng, Alison Maher, Ana Sollano Kim, Dan Clark, Fernando Flori, Jacques Newman, Joey Arhar, Kurt Catt-Schmidt, Leo Lee, Olli Pettay, Stephen Chenney, Domenic Denicola

## Agenda

1. Review past action items
  - a. none
2. Carryovers
  - a. [dminor] [BaseURL is not specified correctly for CSS module scripts](#)
    - i. Recheck if needed on agenda
  - b. [ffiori] [\[focus-without-user-activation\] Allow focus if a descendant has focus](#)
    - i. Preserve behavior, see if users have feedback
  - c. [schenney] [Writing Mode for Canvas text #11449](#)
    - i. Came up during TAG review. Worthwhile to pursue a proposal.
  - d. [lwarlow] [Add missing IDREF reflections](#)
    - i. Carryover to next time
3. New topics
  - a. [kurt] [#10673 Declarative CSS Modules and Declarative Shadow DOM adoptedstylesheets attribute](#), specifically [jyasskin's "syntactic sugar" suggestion from w3ctag/design-reviews#1000 \(comment\)](#) as an approach for this feature.
    - i. Discussed, general support to continue. Kurt to follow up with Anne.
  - b. [jacques] [Focusgroup: Declarative keyboard focus navigation for a scoped set of composite widgets · Issue #11641 · whatwg/html](#)
    - i. General approval of exploration, moved to stage 1
  - c. [alex] [Proposal: Customized built-in elements via `elementInternals.type` · Issue #11061 · whatwg/html](#)
    - i.

## Minutes

- **Transcription and Agenda Management** Chris Wilson initiated the meeting by confirming transcription was acceptable to attendees and noting the need to inform late attendees about automated transcription. We removed the baseURL agenda item as it had been carried over from a previous meeting without action.

- Focus Without User Activation** Fernando Fiori sought clarification on the "focus without user activation" feature, specifically regarding a debug resolution allowing parent frames to programmatically set focus into child iframes. He explained the feature aims to prevent focus stealing but is encountering feedback on whether to allow subframes to use focus APIs once user activation is granted, an edge case that could break some websites. Domenic Denicola questioned the proposal's intent, asking if it aims to prevent malicious frames or if it trusts frames once user activation is given, noting concerns about a wrapper frame bypassing the policy. Fernando clarified that, based on his understanding of its 2019 inception, the main idea of the "focus without user activation" feature was to prevent focus stealing. He acknowledged Domenic's perspective on simplicity and the potential for a wrapper frame to bypass the policy, indicating he was not strongly opposed to that viewpoint. Domenic suggested re-evaluating the policy's purpose based on user and web developer needs rather than just a bug fix. Fernando identified two main use cases for the "focus without user activation" feature: improving accessibility by preventing unexpected focus changes during page loading and preventing malicious intent, such as stealing password input. Domenic inquired about feedback from sites that use these cases regarding the proposed changes and whether the existing behavior breaks important use cases. Fernando mentioned an Outlook example that could break depending on the chosen approach, and Domenic expressed concern about opening up a loophole if the behavior becomes less restrictive. Fernando agreed to raise an issue in the working group to gather feedback on the proposed changes for "focus without user activation," indicating that if no strong support for new behavior emerges, they will maintain the existing, more restrictive behavior. Olli Pettay also noted a separate potential issue with "user translate user activation" staying around too long, which could allow malicious iframes to continue using focus even after navigation.
- Canvas Writing Mode for Text Rendering** Stephen Chenney introduced the issue of inconsistent canvas text rendering across browsers regarding CSS writing mode properties, proposing to add a `writingMode` attribute to the canvas 2D rendering context. He clarified that Firefox's behavior, which uses CSS, is insufficient because it lacks per-drawing call control and does not handle vertical text sensibly with existing attributes. Chris Wilson and Domenic Denicola supported pursuing this fix to achieve interoperability, and Stephen Chenney planned to start drafting a spec PR.
- Interactions Between Writing-mode CSS Properties and Canvas Attributes** Stephen Chenney discussed the interaction between CSS properties and new

canvas attributes for text rendering, aiming for the attribute to take precedence over CSS, similar to direction and language attributes. He acknowledged that existing browser behavior, which often gives CSS precedence for properties like ``direction``, presents a backward compatibility challenge. Stephen Chenney committed to investigating these compatibility issues to ensure a sensible solution.

- **Adopted Stylesheets Attribute for Declarative Shadow DOM** Kurt Catti-Schmidt presented a proposal for an ``adoptedStylesheets`` attribute for declarative Shadow DOM, which would allow stylesheets to be adopted declaratively. He explained that his revised approach uses ``style type="module"`` with a specifier as syntactic sugar for an import map, where the specifier becomes the import map key and the style tag's content becomes a data URI of type ``text/css``. Domenic Denicola found this approach to be a clean fit with existing infrastructure, and Kurt Catti-Schmidt noted that it was simple to prototype in Chromium with good performance. Olli Pettay questioned how the ``adoptedStylesheets`` proposal would work with nested Shadow DOMs or templates, particularly if nested elements wished to use specific stylesheets rather than global ones. Kurt Catti-Schmidt explained that import maps are global, so the first definition of a given key is locked in, meaning nested elements would need to use different names if they desired a specific stylesheet. Dan Clark addressed the concern about name collisions if the same specifier were used for multiple types (e.g., script and style modules), deeming it an unlikely issue and not a blocker.
- **Focus Group for Keyboard Navigation** Jacques Newman introduced "focus group," a proposal to simplify keyboard navigation within components like toolbars and menus, which currently requires extensive JavaScript. He proposed a new HTML attribute, ``focusgroup``, that makes focusable children navigable with arrow keys and requires specifying a behavior, which can map to an ARIA role for accessibility. Olli Pettay found the attribute's content, such as "tab list inline wrap no memory," to be unclear initially. Jacques Newman clarified that the ``focusgroup`` attribute's values like "inline versus block" control keyboard management, "wrapping" indicates wrap-around behavior, and "memory" dictates whether focus resumes where it left off. He emphasized that a focus group functions as a single tab stop, simplifying common web patterns for developers. Joey Arhar confirmed the existence of an associated GitHub issue, and Domenic Denicola praised the proposal for fitting well with HTML's design. The group agreed to move the proposal to stage one, with Chris Wilson marking the issue accordingly.

- **Element Internals Type Proposal Refinement** Alex Keng presented an updated proposal for "element internals type," following last week's consensus to pursue a decomposition approach rather than a comprehensive solution. They introduced a ``buttonActivationBehaviors`` static property, which enables ``commandFor`` and ``command`` attributes and properties, and a new ``command`` event. This property also provides implicit behaviors like form association, click event activation, and default focusability, along with a default ARIA role of ``button``. Domenic Denicola expressed strong concerns about the implicit behaviors bundled with ``buttonActivationBehaviors``, arguing they are generally undesirable and that authors should manually implement them. They specifically opposed a third way to set ARIA semantics, automatic form association for all command-related elements, and bundled click event activation and focusability, advocating for more granular decomposition. Domenic suggested that custom elements and element internals are power tools for developers who should be expected to manage these behaviors explicitly, rather than relying on bundled presets. Alex Keng explained that form association was included to address form submission and reset issues related to button clicking behaviors. Domenic Denicola questioned the necessity of adding convenience methods for already achievable actions and expressed concern that implicit form submission, often disliked by authors, would be bundled with access to form attributes. Dan Clark acknowledged the push and pull between granular decomposition and bundling, citing accessibility concerns that necessitate certain "button stuff" to be included to prevent broken user experiences, particularly with features like popover targets. Domenic Denicola further elaborated on their stance, questioning how much bundling is truly essential for accessibility, distinguishing between essential accessibility and convenience features. They argued that using custom elements and element internals inherently requires more effort, making it reasonable for developers to manually piece together components for accessibility rather than relying on automatic bundling that creates "weird hierarchies". Dan Clark agreed to discuss these points with accessibility experts to determine how much decomposition is feasible without negative feedback. The proposal was then moved to stage one, with no objections.

#### Action items:

- ☐ Fernando Fiori will file an issue to get feedback in the working group and continue with the existing behavior for focus without user activation if there is no feedback.

- ☐ Stephen Chenney will get started on writing up the spec PR for adding an attribute to the canvas rendering context for writing mode and figure out how many sites are making use of the CSS.
- ☐ Kurt Catti-Schmidt will follow up separately regarding the adopted stylesheets attribute and put it back on the agenda if it needs to be fleshed out more.
- ☐ Jacques Newman will add a backlink from the explainer to the HTML GitHub repo for the focus group proposal.

2025-09-11 ( AMER + EMEA ▾ )

Tracking issue: <https://github.com/whatwg/html/issues/11648>

## Attendees

Chris Wilson , Ana Sollano Kim, Brecht De Ruyte, Luke, Leo Lee, Olli Pettay, Stephen Chenney , Joey Arhar, Alex Keng, David Baron, Dominic Farolino , Kagami Rosylight , stephanie zhang, Jan-Niklas Jäschke , Lea Verou , Alice Boxhall, Keith Cirkel, Anne van Kesteren

## Agenda

4. Review past action items
  - a. none
5. Carryovers
  - a. [zcorpan] [Make <embed> and <object> behave like <iframe> for image loading](#)
    - i. Carry over to next meeting
6. New topics
  - a. [dandclark] [Proposal: Customized built-in elements via `elementInternals.type` #11061](#)
    - i. Alex Keng presented proposal, Lea + Keith + Luke + Dan expressed concern over behaves-like approach.
  - b. [dminor] [BaseURL is not specified correctly for CSS module scripts](#)
  - c. [ffiori] [\[focus-without-user-activation\] Allow focus if a descendant has focus](#)
  - d. [stephaniey Zhang] [FormControlRange Interface](#)
    - i. Raised , approved to move to stage 1.
  - e. [danclark,alice] [Reference Target](#)
    - i. discussed
  - f. [schenney] [Writing Mode for Canvas text #11449](#)
  - g. [lwarlow] [Add missing IDREF reflections](#)

## Minutes

**Customized Built-in Elements Discussion:** Alex Keng from Microsoft Edge presented [Proposal: Customized built-in elements via `elementInternals.type` #11061](#), outlining two approaches: a comprehensive solution and a more scoped-down, composability-oriented approach.

Alex: the type property is on ElementInternals. (Example contained a mistake in setting type on this rather than an ElementInternals instance.)

Luke: What does this do that enables not having the wrapping element?

Alex: syntax is similar to radiogroup, using aria role to enable radio behavior. No parser stuff going on.

Luke: Does it make sense for the design to be following HTML as closely as possible? I don't know that input and then various types is particularly good design. Maybe just behave like radio, and mixins can be a bit more scoped?

Alex: possible, but we have to say what behavior is compatible with what, and that might be a lot of work.

Luke: not sure we need the internal stock type at all.

Lea: I agree this is a problem worth solving, but have reservations about framing around specific types of controls (e.g. behaves like button when we might add behavior to button in the future). We should follow a composition approach to frame it around the type of behavior and command invokers and popovers. JS-based inheritance is a well-defined way; static prop that looks kind of like a mix in and kinda insert is a bit concerning. Some of these things seem like general form control attributes, not specific to buttons. I hope eventually we'd get actual proper higher level inheritance. Composition approach would be a lot more clear.

Taylor: I work with Alex on this feature. Main critique against decomp idea is having things too low level will be too confusing for web component authors. They just want something that's button-like. Leaning toward higher level behaves-like first then maybe figuring out how to do lower level.

Keith: I appreciate the problem this is trying to solve. I think there's a missing piece of the conversation: custom element devs aren't aware of what they're asking for. If we implement all of button behaviors for a custom element, it will do things that aren't desired. "We just want something to behave like a button" is somewhat reductive - I think composition is the better choice here.

Dan: hearing a lot of reasonable feedback pushing against a higher level approach here. Some of these behaviors need to be grouped, like enabling pop over target.

Alex: these approaches aren't necessarily in conflict with one another.

Luke: I don't think we should be exposing popover target to custom elements. Just do comm

Chris: let's talk about [Reference Target](#)

Alice: couple of issues: [1093](#), should it be nullable or not?

Keith: +1 to making nullable

Anne: not sure where we ended up after Spain? Seems like one of the less important things to decide?

Luke: can we make progress on any more important decisions?

Anne: if you all prefer returning null, that's consistent with some attrs and inconsistent with others, seems ok.

Alice: there's no removeattribute on shadowroot.

Olli: I think I might have commented in the spec PR?

Alice: seems like we can move forward with that, and have work to do on WPTs. Bigger topic: phase two of puzzle <https://github.com/WICG/webcomponents/issues/1111>. I think we're all pretty much on board with reference target component. The referencetargetmap is less clear what problem it's solving. I'm very confident about ARIA active descendant inside of a list element; if your list item is inside of your shadow root it's not as obv.

Keith: isn't computation recursive?

Alice: not in the way you're thinking.

stephaniey Zhang: [FormControlRange Interface](#) want to see if we can bring this to stage one incubation.

Anne: seems reasonable as stage 1

Stephen: similar bump for canvas-text writing mode feature.

2025-09-04 ( EMEA + APAC ▾ )

Tracking issue: <https://github.com/whatwg/html/issues/???>

## Attendees

Philip Jägenstedt   Jake Archibald   [spieters@mozilla.com](mailto:spieters@mozilla.com)   Keith Cirkel   Domenic Denicola

## Agenda

7. Review past action items
  - a. None
8. Carryovers
  - a. [zcorpan] [Rendering <select> as a listbox is a one-line widget that opens a popup on iOS and Android · Issue #8189 · whatwg/html · GitHub](#)
  - b. [foolip] [Add the `interestfor` attribute #11006](#) suggesting to advance to stage 2
9. New topics
  - a. [zcorpan] [Make <embed> and <object> behave like <iframe> for image loading](#)
  - b. [dandclark] [Proposal: Customized built-in elements via `elementInternals.type` #11061](#)
  - c. [keithamus] Images in <video>

## Minutes

(Discussion of tracking issues) <https://github.com/whatwg/html/issues/11358> example tracking issue

Philip to create one after the fact

## Rendering <select>

Simon: Don't recall what needs doing. The request was for feedback, and that's happened.

## Interestfor

Philip: Next up interestfor. Most checkboxes are checked- except implementer interest. CSS side has landed, :interest-source, :interest-target, and some properties to control delay. So when do we bounce things to stage 2? We do that by consensus of the companies of the steering group. How do we do this without Apple present?

Domenic: \*describes process document\*. We can consult apple async.

Philip: \*describes process document\*

Simon: We should try to get feedback from Apple in case they want to object.

Philip: For sure, but do we have Mozilla support for stage 2?

Simon: <https://github.com/mozilla/standards-positions/issues/1181>

Keith: We have some concerns but no hard objections. Need to review the PR.

Philip: On device independence, we discussed this in OpenUI, provided a ton of options, spoke to UX folks. The model we landed on was an added option to the long press context menu. If you prevent the context menu from showing we immediately show the popover without additional delay. What's covered in spec is keyboard, pointer devices (mostly mice & touch), and for any other modality it says the UA "should". Not "must" because for example it might not make sense on a wearable.

Keith: How successful has the origin trial been?

Philip: I'll get back with details.

Domenic: Looking at the WebKit standards position, if we're happy with Mozilla actively affirming stage 2, then we should leave a note on the SP and ensure that if they strongly object then it is crystal clear.

Philip: Should we say stage 2?

Keith: I think we can provide positive affirmation of stage 2.



Philip: When should we make the comments?

Domenic: Tuesday

Philip: We can give it one week. I'll also comment on the tracking issues, and on WebKit SP.

## **Embed and object**

Simon: We made some changes, did some testing but it turns out there's some file extension shenanigans. We need to assess the webcompat issues.

Domenic: Everyone wants to move to the autosizing model

Simon: SVG needs it, for webcompat.

Domenic: What if everyone just puts width/height on their embeds?

Simon: I need to look into it, I ran it on 1% or 1000 pages or so. Many object/embeds are flash. Running on full dataset is expensive but maybe we can bite that.

```
SELECT
  page,
  url,
  rank,
  REGEXP_EXTRACT_ALL(response_body, r"(?i)(<(?:object|embed)\s+(?:[^\>]+)>)" ) AS
elements
FROM
  `httparchive.crawl.requests` TABLESAMPLE SYSTEM (1 PERCENT)
WHERE
  date = "2025-08-01"
  AND client = "desktop"
  AND is_main_document IS TRUE
  AND REGEXP_CONTAINS(response_body, r"(?i)(<(?:object|embed)\s+(?:[^\>]+)>)" )
ORDER BY rank, page
```

Domenic: What we really need to find is objects & embeds with no width and height

Simon: I plan to filter it further.

Jake: Would it be width \_or\_ height. Autosizing kicks in if it needs to calculate one or the other?

Domenic: Maybe. I don't think Chrome likes the extension behavior, I don't think Mozilla does the same thing?

Simon: I think we figure it out after we've received something.

Domenic: You don't receive width or height until after it loads and then you have the content type.

Simon: Yes also the browsing context.

Domenic: So you always create the browsing context and add autosizing on response?

Simon: Except images don't have a browsing context. Maybe we create and destroy later.

Philip: Aren't there magical documents that exist for this? Images and iframes. Is it same for embed and object?

Simon: Yes it is now in Gecko but we have special code to hide the browsing context from the page.

Domenic: That still seems cleaner - hiding it from document.frames and such

Simon: If we can get rid of that too, that would be nice but we're not sure on compat.

Philip: Can we interrogate this through http archive, or usecounter?

Simon: Usecounters seem nice, but I'm not sure exactly what to count.

Domenic: Upper bound could be page with iframe that access the index property, like frames[0] - you might get a different response.

Simon: It could break pages that fetch some other browsing contexts if you change the indexing.

Domenic: Adding a flag like "is hidden from index" isn't the worst in the spec. We should spend some time on the minimum thing we can do, e.g. autosizing based on extension or content type. With spec editor hat on I'd be fine making everything hidden browsing context.

Simon: Not clear on handling of plugins, if a page tries to use flash, the page isn't supported but it doesn't have a browsing context.

Domenic: Enthusiastic about this though.

Philip: What's different between iframes and embed/object with browsing contexts?

Jake: They're the one thing on the platform where it `_may_` create a browsing context, based on the response, but in Chrome it's based on extension?

Domenic: At spec level it's much messier. Sometimes a browsing context, sometimes fallback content, sometimes image. A totally different path where it fetches the response and feeds it into the create a browsing context/fallback/image algorithms. Very different from anything else, an image for example goes through all the image request/response machinery, whereas embed does not.

Simon: Also serviceworkers don't support object/embed because they're too messy.

Jake: One reason behind that was browsing context, but the other reason is lack of trust of the security model of plugins. So even if we could know if there was a browsing context we'd still skip it because of plugins.

Domenic: There are no plugins on the web!

Jake: We can add a point saying if any UA is still doing plugins they **MUST** not or something.

Simon: Okay, I have some work to do.

Philip: Simon I shared a table with you - 118k rows. Is that right?

Simon: It's not a lot of pages using embed/object.

Philip: Of 16 million in http archive?

Simon: 11 million for desktop.

Philip: Next up

### **ElementInternals type**

Philip: Domenic wrote many words on the issue, can we do anything here?

Domenic: Does mozilla have words.

Jake: A frustration that custom element cannot be a form.

Domenic: This target submit buttons, generic buttons, labels.

Keith: Good motivations but lack of use cases.

Domenic: Normal buttons with popovertarget and so, but submit buttons, I've heard use cases...

Philip: As we don't have much I'll treat as new topic for next time. Any other late additions to the agenda?

## Images in video

[Proposal: Support image formats in `<video>` · Issue #11554 · whatwg/html](#)

Keith: ex-colleagues at GitHub aren't sure image-in-video is useful for them. People don't want their markdown to be translated to `<video>`. They want a specific solution to if prefers-reduced-motion, gifs become click-to-play; maybe image-in-video is generalizing too far? Maybe UAs can just do that? Or we can put it in CSS? Any thoughts?

Domenic: Browsers don't do stuff automatically to prefers-reduced-motion so they might not want to do it here.

Jake: Changing the default maybe not but an opt in.

Philip: Is it widely used enough?

Jake: prefers-reduced-motion comes from user but opt in needed by developer.

Simon: Some users have it without knowing right?

Jake: If someone is creating a css animation & video and we stop animating part of that it's a tricky situation.

Domenic: Popular windows tip to improve performance!

Jake: CSS has animation-play-state, so making their own button if they want.

Philip: There's a whole slew of issues around this, autoplay is one, but seeking on gifs is another?

Jake: Lea's proposal is making images work in a video tag, and an animated image format would get controls. I would like it if images/video were just raw media formats and an image tag could play any supported format.

Simon: Video formats in img, should we synchronise those animations. Gifs are synchronised.

Jake: Particular to gifs? Animated PNG? WebP?

Simon: All of them. I havent checked what Safari does though.

Jake: Animated AVIF in an img performs terrible compared to AV1 in a video. So something about that pipeline is difficult to optimise, I wonder if that's part of that.

Philip: Image decoders I think are CPU only.

Jake: Gut feeling is videos in image tag are more interesting than images in video tag.

Simon: Videos are more efficient for network. Can transform to another format on the server.

Jake: CSS could be interesting, but how does SVG work? Maybe it doesn't.

Philip: Do videos respect prefers-reduced-motion?

Jake: No, we discussed autoplay=maybe or something that would detect/respect it.

Philip: Is any user pref?

Jake: Yes but separate from prefers-reduced-motion.

2025-08-28 ( AMER + EMEA ▾ )

Tracking issue: (missing)

## Attendees

Mason Freed, Noam Rosenthal, Stephen Chenney, Luke Warlow, Lea Verou, Olli Pettay, Jake Archibald, Benjamin VanderSloot, Kagami Roslylight, Domenic Farolino, Joey Arhar, Jan-Niklas Jäschke, Keith Cirkel, Simon Pieters

## Agenda

1. Review past action items
  - a. None
2. Carryovers
  - a. [Rendering <select> as a listbox is a one-line widget that opens a popup on iOS and Android · Issue #8189 · whatwg/html · GitHub](#)
  - b. [Add the interestfor attribute #11006](#) - needs review
  - c. [Should we remove XSLT from the web platform? #11523](#)
    - i. <https://github.com/mozilla/standards-positions/issues/1287>
    - ii. [https://github.com/mfreed7/xslt\\_polyfill](https://github.com/mfreed7/xslt_polyfill)
    - iii. [https://github.com/mfreed7/xslt\\_extension](https://github.com/mfreed7/xslt_extension)
3. New topics

- a. [Proposal: Support image formats in ``<video>`` · Issue #11554 · whatwg/html](#)
- b. Listen to and (hopefully) resolve between:
  - i. [How to handle `addEventListener` on ``CSSPseudoElement`` · Issue #12163 · w3c/csswg-drafts](#)
  - ii. [\[css-pseudo-4\] Proxying pseudo-elements as "real" \(`<slot>`\) elements · Issue #12575 · w3c/csswg-drafts](#)
- c. [Proposal: Add Writing Mode to `CanvasRenderingContext2D` and `offscreen` Issue #11449](#)

## Minutes

Noam: Let's go through.. No past action items. Few carry overs.

Joey: First one (rendering `<select>` as listbox). Would like to hear back on mozilla standards position. Spoke to Olli about this last week.

Olli: Didn't get a chance to get to that.... XSLT took some time. Try to get to this this week.

Joey: Thank you.

Mason: Interestfor, mentioned last meeting, I need to check all boxes. I would love a review.

Current spec PR - thanks to Phillip - aligns to chromium

Keith: On my list

Mason: Corresponding changes to CSS spec are linked too. Check those out.

Dominic: Don't see CSS PRs in OP of spec PR, are they further down?

Mason: I'll get those added. They're in spec text but we should add pointers directly.

Dominic: Couple from random comments but OP would be easier.

Mason: I'll get Phillip to do that.

Mason: XSLT is here as a carry over. Thanks for standards positions. Is anyone from WebKit able to provide a standards position? Would be much appreciated. Without anything else changing my plan is pushing forward - exploring usecases, sites found to use XSLT, plus the list Simon put together. Making sure there's a story to tell for those, so we don't break them. There's a polyfill I've been working on which is pretty functional at this point. Works in line in an RSS feed which gets it working for developers who add it. There is a browser extension which injects the polyfill on any site, meaning those still work.

Jake: If I wrote something about XSTL is it worth linking to the polyfill/extension?

Mason: For sure, the links are:::

Mason: Several people are PRing to polyfill which is great.

Luke: Is there a situation where that extension could pre-ship with the browser? Some ship by default (maybe not publicly facing). If it's found it's not webcompat, too many routes reliant on this, is that doable?

Mason: It has come up, but we're loathed to do it. Anything published by the browser is owned by the browser forever. The thing I did do which we're happy doing, if you navigate to a site that uses XSLT, it'll come up with a banner pointing to a search for XSTL extensions. So it's close to what you asked for.

Luke: The other side, if it's not webcompat, what would be the plan? A document policy added? I as a site owner can say I don't want it running. I don't know the nature of the vulns. Giving site authors an opt out seems useful. If deprecation is going to take a while.

Mason: Good idea. If we're moving ahead with deprecation there'll be an enterprise policy to opt back in but I hadn't thought of early opting out.

Keith: the search - are we worried about phishing? idk what the security model is for chrome extension store, but i know there are extensions that report website usage statistics, so someone could make an extension ocalled xslt thing and then exfiltrate users site data. is that a concern? also, is the extension compatible with all browsers or just chromium?

mason: im not an extension expert, but its a simple extension thats 20 lines long. idk if its directly compatible or just needs modifications. im now afraid of the phishing thing

keith: theres presumably a security team looking at that, so its just a case of throwing the commentary to that team? telling them thats a vector and that theres nothing mentioning xslt that could climb that

mason: youre right that makes it more vulnerable. ill talk to extensions about this.

Noam: Anything else on this?

Mason: Anyone else in the room not okay with this? Or are we all okay moving forward?

Simon: Okay with moving forward. One question: How do we find devices that are affected. Seems tricky. Try to ship to a number of users to see who complains but are there other strategies?

Mason: Rooting out cases... (missed)

Simon: We don't have data for logins on specific devices that are in users local network, for e.g.

Mason: Same for every deprecation, to varying degrees. Embedded could be more difficult. Lots of loud warnings that might get noticed, and in Chrome turned off completely way early in pre-stable channels so it breaks early in pre-stable. When the time comes to turn off, be careful and slow... turn it back on as we find new use cases. Suggestions appreciated.

keith: do you have a usecounter to see when people use the extension?

mason: there is a counter on the extension being installed, i dont have a use counter for the other thing now. ill take that as an action item

Luke: Other idea - is it plausible that it can be allowed on local IPs only? Stuff like routers can work but open web doesn't. Not the greatest but better than nothing.

Mason: A world where all of this is possible, but it means that population is still vulnerable. It's not as good but it's an option if we need to take it.

Mason: Mutation events got removed successfully with little fanfare and had 200x the usage.

Noam: I don't see problem with embedded. They don't have to upgrade to new chromium, if they do they can put their own extension on their device. It's almost a lost problem wrt deprecation, they chose what version.

Mason: Data served from a device with a modern browser. So a server, serving XSLT. You would need something like an extension.

Stephen: The biggest invisible population is enterprise. No use counter will ever tell you what's going on there. Government service sites which rarely get updated, but still important to the population are the biggest risk.

Mason: +1, what i'm most worried about. Been reaching out to enterprise folks. If everyone could do this fact finding mission - find a way to contact this population that would be great.

Lea: Embedded devices - having as browser flag might help? But wouldn't help with enterprise.

Noam: Cautiously moving forward then?

Mason: That's the conclusion I'm going to draw unless someone says otherwise.

Noam: Supporting image formats in video. Did we discuss this?

Lea: We discussed CSS parts in CSSWG.

Lea: These slides are for a different problem but overlap some of the scope.

Lea: (presents some slides: <https://webplatform.design/talks/image-animation/#cover>)

Lea: Doesn't look like extensive changes required. It is arguable animated images are just video with no audio, just a different codec, essentially. One hairy thing: how do you deal with static images in video, what duration do you show? Do you still show controls? How feasible? A few details but overall seems like a small change, the gains are significant though.

Jake: With my web developer hat on I agree with all of this. Another fun thing: video with transparency on web you're either using AV9 or AVIF, AV1 doesn't support it. The video equivalent doesn't have transparency. If you put a jpeg in a video - the duration? Just do the same thing as a 1 frame video, is that a duration of 0?

Lea: Duration 0 makes the most sense. It's about feasibility.

Jake: Talked to some implementers here, our image/video pipelines are very different so it would be implementation pain but aside from that, I like it.

Stephen: +1 to treating as single frame video. Worst case that's viable. I think in Chrome the pipelines are quite different, so would require... it's not very easy to implement... video always go through compositor I think. Not 100% sure. Currently if you have duplicate images in many image tags they all animate at the same rate but videos don't, which is probably good but there's a memory cost associated with this.

Lea: One thing someone might ask: what about the opposite, making `<img>` and media element, it might be great independently, it's a lot harder, but maybe the same. In terms of API - because defaults of image are different to media elements, e.g. autoplay on by default, looping by default, webcompat of changing the prototype chain. Advantages of doing with img is a tailored UI, usually simpler UI for animated imaged

keith: why do you need to make changes to the prototype chain?

lea: Oh simply allowing videos in `<img>` is totally fine, though it doesn't solve the issue with playback. But it would be another nice step in bringing them together and WebKit already implements it.

keith: we could propose those additional apis/attributes on img elements without any web compat risk of altering prototype chains

jake: i agree with keith, i dont think you mean that as a counterproposal, its something we could do in the future

keith: yeah if my tone wasn't supportive i will explicitly say that wasn't me saying its a bad idea, im just saying we can avoid compat issues this way

lea: Not a counterproposal, though i was hoping to gauge whether this is a way that we may go down the line, as some of the proposals we discussed in the CSS WG would actually get in the way of that ever happening.

Noam: I think this needs implementer interest, not just "this looks like a good idea" but "we are going to do it", in a roadmap. From idea to execution. That's my take on this. Prototype change



is somewhere between never and probably never gonna happen. Not the kind of work any browser implementer is going to take on.

Jake: Could create a mixin in specland, if there's commonality.

Simon: I wanted to say similar. It doesn't make sense to have image support everything video supports. Images have distinct feature sets. When we overload elements in HTML it causes problems, e.g. `<input>`, `<object>`. For web developers if they want all of what `<video>` does, they can use `<video>`. I'm not sure why I'd add all the features to `<img>`. It doesn't enable anything in particular.

Emilio: Agree, e.g. decoding attribute, I don't think we want to synchronously decode video, so +1 to just supporting images in media but the other way around seems more complicated.

Luke: Of use cases here, how important is lazy loading?

Keith: We have open issue for this.

Simon: We can add lazy loading to video, just needs more work

Noam: We can add this to the mixin. Problem solved-ish.

Lea: Also things like fetchpriority, but high isn't desired and use cases for low can be sort of dealt with lazy loading

Noam: Videos are specced to load with 206 range requests. Images are not, specifically. We'd need to figure that out. I don't know if you can 206 images in the same way. Say an image with 5 minutes, or a few seconds, you just want the middle part - you probably need all of it, so can't use ranges. A detail that once implemented can probably be figured out.

Lea: Other implementation detail is can we show controls if only controls are needed.

Simon: Range might also be webcompat issue - sending range requests from images.

Expectations from server side might be to get range requests from videos but not images.

Jake: You might get a range request initially, but as soon as it realises it's a format it can't range request on it'll stop.

Simon: Right but videos from `<img>` - having `<img>` always send range request might be a problem.

Lea: Yes, lets not focus on video from img. Videos already send range requests for an image, I suppose? Though right now maybe it's not very used. Presumably video still needs to send range request to find out mime type, so initial requests are already being sent?

Noam: We can handle these details more during execution.

Lea: A lot of implementers here. Any sense from implementers, about interest?

Stephen: Igalia is involved with this, we have a client who is interested. We can start pushing this forward.

Keith: I guess you want procedural confirmation of stage 0 or stage 1?

Stephen: I agree it's probably at the point where people want to move forward with it but we need to get a proper proposal up.

Noam: Yes if there's no spec at all it doesn't matter but we need a champion.

Lea: I am championing it. It's funded.

Keith: If you assume stage 1 then the next step is to write a draft spec.

Noam: Stage 0 criteria is met.

Lea: Does anyone have requirements for stages?

Keith: <https://whatwg.org/stages>

Noam: Stage 0 is the group is interested.

Dominic: Stage 1 is more formal, we would need consensus that we would accept a draft spec.

Lea: We have consensus that this is a problem worth solving? There is at least one implementer interested.

Noam: Internally, I'd like to speak with experts to ensure there are no strong objections to this.

Stephen: We'd have to do an I2P and talk with video people.

Dominic: Also consensus the problem is worth solving. Did Apple weigh in? If not we should reach out to them first, rather than surprising Apple. So we can say Stage 0 for now and reach consensus for Apple.

Lea: So I should hold off on a draft spec PR?

Keith: Raise a standards position on webkit/standards-positions for Apple.

Simon: I should clarify the staging process is opt in. You can start with writing a PR then a proposal if that's how you want to do it. Maybe you wouldn't reach a stage until the requirements of that stage are met but it doesn't prevent you from doing things that aren't in a particular stage.

Lea: Some chicken and egg, easier if there's a draft spec but draft spec is easier with consensus. But I get the general idea.

Noam: Anything else? ... Let's move forward

Noam: About pseudo elements. There's a lot more pseudo elements than there were before. It's a lot more expressive than it used to be. Desire to give them JS apis that relate to layout & UI events etc. The main thing we discussed in last meeting was to listen to clicks and other events, that go on specific pseudos - rather than defaulting to the element, without knowing where it came from. Also getBoundingClientRect, ResizeObserver, connecting a pseudo to a popover target. 2 proposals for this: 1 specific to event listeners, already a spec for IDL interface for CSSPseudoElement, that IDL interface is not the same as Element, but it's a proxy to an element plus the name of a pseudo. So you can say "this's before" or "this's backdrop". They always exist even if the pseudo element doesn't. Might or might not happen. Style & Layout create these on the fly, not connected to JS, not in real DOM. 2 directions: current spec to have this proxy element, have that be an EventTarget. If you want to listen to clicks on a before element you call `element.pseudo('::before').addEventListener()`. Another direction proposed: proxy elements with a real element instead, the idea is that when you want a pseudo element to behave like real DOM, you add a real DOM element to proxy them. It's not displayed by itself but represents the pseudo, but style & layout is dealt with by the pseudo element, the DOM element is just a proxy. Entirely encapsulated inside CSS world, no changes to HTML, but part of web platform. Strong opinions from this group about the other direction. We want to know what people feel about this comparison.

Lea: This is not for all pseudos? Just element backed pseudos? What is the scope?

Noam: Any pseudo.

Lea: Like first-letter?

Noam: No just tree abiding.

Lea: <slot> allow shadow dom to compose light DOM but this means slots in light DOM? The reverse of what it's doing? If you see those pseudo elements as that. Maybe a broader thing there. I like how you can use the existing DOM and everything. I'm quite in favour of this

solution. There are adjacent problems, but maybe it can solve those. A lot of interesting design space. One of the best ideas to come up with this problem.

Noam: Proposal right now is you can use any element. The cross reference between element & pseudo is in CSS, it's a `display` value property. Having real element with semantic names is also interesting.

Luke: How does this work with AT? A button exposed as a button that is suddenly something else? Scroll marker provides its own A11y stuff, does HTML win? CSS win? Are there weird situations, stuff like if you've got aria-hidden on the pseudo element element. E.g. spin buttons in number are hidden, do you need to manually do that in HTML, or does it automatically do that? How does it work if it changes styles dynamically? If I hover a button and style changes to `display: pseudo(backdrop)` does that change it? Gut reaction on `display` is that it doesn't feel like the right property here. Display is layout oriented, this seems to be entirely new.

Noam: A11y by default is the element will be display:none, plus whatever bespoke things to help with a11y.

Emilio: This comment about CSS - it feels like ... if you expose it via CSS it breaks things you could otherwise depend on, stuff like you'd need to resolve style before you can dispatch an event, so it probably needs to be in the DOM. I'd rather make something like ``<layout-target>`` a real element rather than this proxy thing. I haven't thought out how this would work.

Noam: The pseudo element is not known in DOM time though, only in layout, e.g. you don't know if there's a backdrop until you've computed style, so making the association in DOM doesn't change that.

Emilio: Imagine there's an element named `<pseudo>`, you can make that element magic and not have a layout box, and layout could make it when needed, everything kind of just works, I think? But if it's not a DOM attribute then you need to figure out how DOM apis work. DOM apis don't need to know their style in order to work.

Luke: The other thing - it does still come back to, if you have this thing that you want to be a real element, why is not a real element? Scroll markers are in some ways links, so should we have a way to make scroll markers links? File input button for example would not be a thing today, we'd add an element. Some things can't be elements, e.g. view-transitions. I worry this is over indexing on one problem space - a small subset rather than all pseudos. Maybe we address the small subset.

Lea: One component of this is encapsulation. Arguably if you want to expose them like this encapsulation is not a concern. E.g. shadow dom backdrop might have wrappers you don't want to expose. Other than that maybe you don't want to write a bunch of HTML.... There are cases of elements that are generated when not supplied, e.g. tbody, etc. We don't have a good pattern for layout generated elements right?

Noam: Right but like `::column` is a pseudo element, but you don't know how many columns you have, you might have N based on scroll, layout, etc. This is the only proposal right now that deals with this. There's no current... Probably whatever is created in this space needs to be compatible with how pseudo elements work.

Lea: What are use cases beside backdrop? I know there are many but where is the list?

Noam: I'll send it. Scroll markers are one. People want to know which one is clicked. Likewise view transitions, which are generated on the fly. DOM creates layout creates style creates DOM. A pseudo DOM.

Lea: What % of usecases are covered for shadow dom cases? Backdrop is solved but view transitions are not.

Noam: Shadowdom already has an element, event propagation, shadowRoot, getBoundingClientRect etc.

Lea: Not for closed shadow roots.

Keith: Time check

Noam: let's timeout this call and continue in thread. Good start!

Older meeting notes are here:

[https://docs.google.com/document/d/1iaarr4Ho715CUULrvi\\_LD3TwshAcN2odDLBBEK0FjH0/edit?tab=t.0](https://docs.google.com/document/d/1iaarr4Ho715CUULrvi_LD3TwshAcN2odDLBBEK0FjH0/edit?tab=t.0)