**Weekly Nerd #1**

**Stop using JS for that**(Door Kilian valkhof)

* 20 jaar ervaring met websites bouwen
* Deel van Electron Goverance Team
* Maakt een browser voor Polypane

(<https://polypane.app/>)

* HTML/CSS is **bewerkbaar**
* The rule of least power (basisprincipe voor hoe web technologieën gebouwd zijn).

*Choose the least powerful language suitable for a given purpose.*

**(Schakel eerst van zwak naar sterkst) Javascript is heel gevoelig en breekbaar.**

Als je duidelijke HTML schrijft kan men het ook gebruiken.

**Browser makers are listening.**

And are implementing the stuff we built by hand.

De browser kan alles wat jij kan in javascript maar sneller en toegankelijker.

**Once you learn something**, you don’t learn it again.

Alles wat je bouwt in de browser, blijft werken in de browser (met name moderne browsers).

• Als Web Developer heb je minder tijd onderzoek om te leren wat je graag wilt.

In de komende jaren, als je iets bouwt wat je eerder hebt gebouwd, kijk eerst of er misschien een betere manier is om hetzelfde te bereiken.

**Let’s get started**

* Custom Toggles (oftewel switches). Een slider bijvoorbeeld die in principe werkt als een checkbox. Je kan in principe hetzelfde bereiken met een checkbox die de browser zelf aanbiedt. De browser doet in principe al heel veel voor je.

<label>

<input type=” checkbox”/>My Awesome feature

</label>

Op basis van jouw operator weergeeft hij de checkbox. Jouw input elementen zijn daarom ook minder stijlbaar. Met **appearance: none**, kun je de styling checkbox van de operating system negeren. Doormiddel van pseudo selectors “::before” en “::after”. Je hebt ook een checked pseudo class waardoor je duidelijk een onderscheid kunt weergeven van **aan** en **uit.**  Focus visible geldt enkel als je niet met een muis interactie hebt met een element. Nu kun je de focus stijlen! Met outline color transparent wordt jouw transparent kleur vervangen met de kleur van de gebruiker (voor outlines, voor tekst kleur). Je maakt de gebruiker zo makkelijk. Safari herkent de **switch** binnen het input element. attribuut en maakt er ook een switch van.

**Datalist**

De datalist is een element binnen HTML waar je verschillende opties mee aan kan geven die je binnen een input kunt gebruiken. De browser heeft meer toegang dan wat jij wilt kunnen implementeren via Javascript. Met bijvoorbeeld een color-picker. De browser geeft jou meer opties waar je veel minder voor hoef te doen.

**In-page transitions**

Het is veel fijner als je scrolt naar het deel waar je gebruiker naar toe wilt. Stel je klikt weg op een lange wikipedia pagina, dan weet de gebruiker niet. Door middel van scroll-behavior scrolt de pagina zelf naar positie die je aan biedt. Veel mensen kunnen duizelig worden door het scrollen (soort motion sickness).

**Accessibility**

Stel je hebt een fixed header, en je scrolt naar een locatie die de navbar bedekt. Daarvoor kun je heel handig deze CSS gebruiken:

**CSS:**

Scroll-margin-top: 100px.

**Target**-psuedo class kun je duidelijk highlighten waar je als gebruiker bent.

Met **position sticky** kun je intentioneel zijn met wat je blijven tonen door waarheen de gebruiker scrolt.

**Carousels**

Je hebt nu scroll-snap API’s. Met scrol snap kun je sliders maken die snappen aan de rand van hun omliggende element (Parent Element). Je hebt scroll-snap type, die zet je op de parent. Dan geef je aan in welke richting je scrolt (dat kan X of Y zijn). En dan geef je aan op wat voor manier je wilt dat hij snapt. Dat kan via *mandatory* of *proximity* zijn. In het child-element moet je aangeven **waar** je element begint. Dat kan start, center of end zijn.

**Accordians and Modals**

Stel je hebt bakken aan data, dan kun je details en summary gebruiken. Handig bij FAQ, secties binnen een website. Wil je dat driehoekje veranderen? Gebruik de marker selector.

summary::marker {

}

[open] summary::marker {

}

**Dialog**

Dialog is een nieuw element in HTML wat je kan gebruiken om je gebruiker keuzes te laten maken binnen een modal. In JavaScript kon je alert, prompt, etc. gebruiken. Je kunt niet de top-layer zelf iets stijlen maar je kan wel er iets tussen zetten. Het laatste wat je hebt toegevoegd aan de top layer komt vooraan. Je kunt ook meerdere submit buttons hebben binnen formulier. Met starting-style kun je animeren (Dit is **nog** in zijn werking). Met backdrop selector kun je de dialog styling.

**Container queries**

Ze werken als mediaqueries, maar in plaats van hoe breed is mijn browser kun je kijken naar het omliggende element. In plaats dat je steeds rekening moet houden met waar het compenent zich bevindt kun je doormiddel van countainer queries aan de hand van de afmeting werken. Met CQ-units kun je kijken hoeveel, hoe breed iets specifiek groot moet zijn binnen de container.

**The Parent Selector :has()**Stijl op basis van een ander element. Je kunt een bepaalde styling toepassen op een element die de gebruiker zelf kiest. Polypane, where is has website.

**Stop using JS in the future…**

Field-sizing selector is het HEEL handig om de breedte van een input en textarea zelf te bepalen.

**Selectlist**

Seleclist heeft alle semantics en interacties voor een lijst. Elke seleclist heeft verschillende parts. Dropdown, selection, option.

**Scroll-driven animations**

Scroll-driven animations vervangen elk bestaand libraries die nodig had moeten hebben om bepaalde animaties te laten afspelen.