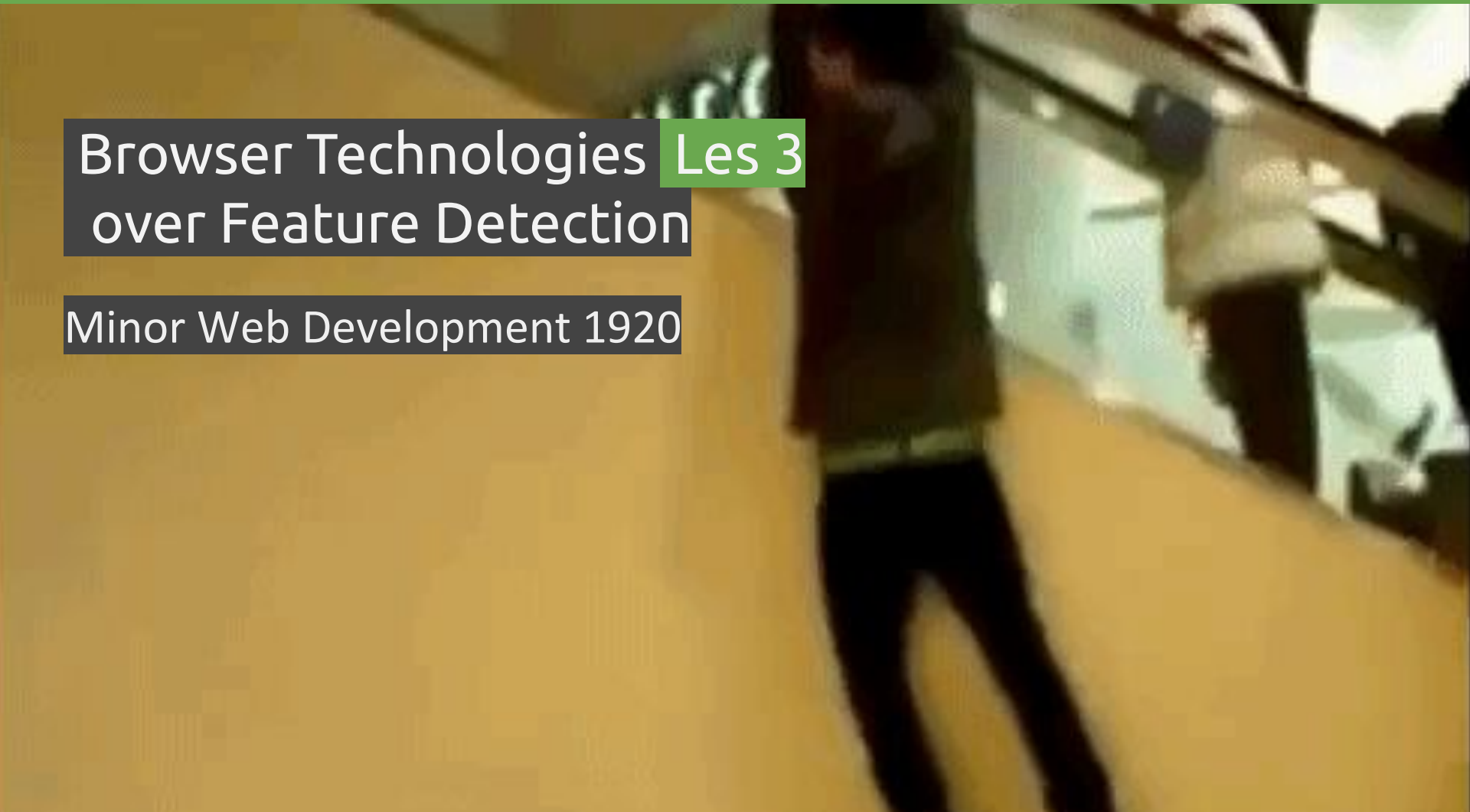
Browser Technologies Les 3
over Feature Detection

Minor Web Development 1920



Browser Technologies Les 3

Vandaag

1. Over de planning en gang van zaken
2. Terugkijken naar de opdracht van vorige week
3. BEspreken van het artikel *HTML: The Inaccessible Parts*
4. College over Feature Detectie en Browsers
5. Opdracht 2: Progressive Enhanced Browser Technologie

Browser technologies 1920								
Week 2								
	Woensdag 18/3			Donderdag 19/3			Vrijdag 20/3	
	College Feature detection en Browsers			College Vragen over Browsers, PE en Feature detectie				
	2. Progressive Enhanced Browser Technologie	Bespreken in groepjes			Bespreken in groepjes		Code review	
	HTML: The Inaccessible Parts			The Role of Enhancement in Web Design				
	Accessibility Through Semantic HTML			The Accessibility Mindset				

Browser technologies 1920							
Week 3							
	Woensdag 25/3			Donderdag 26/3			Vrijdag 27/3
	College over Notificaties						
	Studenten presentaties	Bespreken in groepjes			Bespreken in groepjes		Beoordeling
	Progressive Enhancement and Data Visualizations						
	Make the Web Work For Everyone						

Browser Technologies Les 3

Terugkijken naar de opdracht van vorige week

Breek het Web - Features

1. Afbeeldingen => Inhoud => Video?
2. Custom fonts => Presentatie => Font rendering?
3. Kleur => Fysieke factoren => Blind?
4. Muis / trackpad / toetsenbord => User input => VR?
5. Breedband internet => Omgevingsfactoren => Context?
6. Cookies => User settings & preferences => Do Not Track?

7. Javascript volledig => blockers, CDN’s, fouten door developers
8. Javascript deels, bepaalde features => browser & device verschillen

   Vaak (ook) geen keuze voor de gebruiker

The Web is ubiquitous & messy

- Afbeeldingen: optioneel
- (Delen van) JavaScript: optioneel
- Snelheid: optioneel
- Muis: optioneel
- Maar ook: localStorage, keyframe animations, kleur, fonts, `<canvas>`, `<video>`, transforms, `position: fixed`, touch events: optioneel.
- En nog veel meer technieken die we nu over het hoofd zien: *optioneel*.

Browser Technologies Les 3

College over Feature Detectie en Browsers

Instead of thinking about the specifics of how a finished website might look, progressive enhancement encourages you to think about the fundamental meaning of what the website is providing.

— *Jeremy Keith*

Progressive Enhancement - Cutting the mustard

"HTML5" browsers:
IE9+
Firefox 3.5+
Opera 9+ (and probably further back)
Safari 4+
Chrome 1+ (I think)
iPhone and iPad iOS1+
Android phone and tablets
Blackberry OS6+
Windows 7.5+ (new Mango
Mobile Firefox (all the versi
Opera Mobile (all the versio

USP: Provide a news service tailored to new hardware/software

"HTML4" browsers:
8-
Blackberry OS5-
Nokia S60 v6-
Nokia S40 (all versions)
ants
-Mango)
t are too numerous to

USP of the core experience is speed

```
if('querySelector' in document
    && 'localStorage' in window
    && 'addEventListener' in window) {
        // bootstrap the javascript application
    }
```

http://responsivenews.co.uk/post/18948466399/cutting-the-mustard

Progressive Enhancement - Bouwen in 3 lagen

1. Focus op HTML; *basic functionaliteit*, structuur, de basis

   Samen met HTTP, URLs, `<form>`, `<input>` en `<a>` kom je enorm ver.

2. CSS; presentatie, stijl, ~~*optioneel*~~

   Bedenk per feature wat de impact is als deze niet ondersteund wordt.

3. JavaScript; gedrag, interactiviteit, *optioneel*

   Bedenk per feature wat de impact is als deze niet ondersteund wordt.

Feature Detection - 1. HTML

- HTML Design Principles
- Zoek eerst of iets al bestaat in HTML

  `<audio controls>`
- Parser negeert onbekende tags, gratis fallback

```
<video src=”foo.mp4”>
    <a href=”foo.mp4”>Downloaden kan ook</a>
</video>
```
- Parser negeert onbekende attributen, gratis fallback

  `<input type=”date”>` naar `<input type=”text”>`

Feature Detection - HTML resources

- De specificatie, *first stop*: whatwg.org/html5
- Pas op met w3.org/TR/ voor HTML (HTML 5.1, diffofhtmls.herokuapp.com)
- Design Principles: www.w3.org/TR/html-design-principles/
- Dive into HTML5: Dive into HTML5
- Gebruik POSH: microformats.org/wiki/posh
- Writing forward-compatible websites: developer.mozilla.org/en-US/docs/Web/Guide/Writing_forward-compatible_websites
- Can I use… (en dan met name “Known issues” en “Resources” tabjes): caniuse.com
- HTML5test: html5test.com
- HTML5 Validator: html5.validator.nu

Feature Detection - 2. CSS

- Parser negeert wat ‘ie niet kent

```
transition: .3s ease-out;
background-image: linear-gradient(red, blue);
```

- Daardoor forward compatible

```
display: flex;
```

- Fallback

```
height: 200px; height: 20vh;
```

- Enhancements (animaties, transities, transforms, etc.)
- Gebruik `@supports { ... }`
- `Vendor prefixes (-webkit-)`

Feature Detection - CSS resources

- `@supports` op MDN: developer.mozilla.org/en-US/docs/Web/CSS/@supports
- CSS Reference: developer.mozilla.org/en-US/docs/Web/CSS/Reference
- QuirksMode.org - CSS (échte tests): quirksmode.org/css
- CSS3Test.com: http://css3test.com/
- The limits of `@supports`: https://www.quirksmode.org/blog/archives/2016/07/the_limits_of_s.html
- (Niet gerelateerd) CSS Nesting: https://drafts.csswg.org/css-nesting-1/

En let op: http://dowebsitesneedtolookexactlythesameineverybrowser.com/

Feature Detection - 3. JavaScript

- Foutgevoelig
- Parser stopt bij een fout (+ foutmelding in je console)
- Object detection (*Cutting the mustard*)

  ```
  if ('foo' in window)
  if (window.foo)
  ```

- Defensive coding

  `event.preventDefault()` op de juiste plek (ná al je code)

Feature Detection - JavaScript resources

- QuirksMode.org - JS: quirksmode.org/js/contents.html
- Object detection: quirksmode.org/js/support.html
- Detecting HTML5 Features: diveintohtml5.info/detect.html
- Progressive enhancement and JavaScript failure: molily.de/javascript-failure/
- Data Visualizations: css-tricks.com/progressive-enhancement-data-visualizations/
- The Hijax approach, “Plan for Ajax from the start, Implement Ajax at the end”: domscripting.com/presentations/xtech2006/

En let op: http://dowebsitesneedtobeexperiencedexactlythesameineverybrowser.com/

Browser Technologies Les 3

Briefing opdracht 2 - Progressive Enhanced Browser Technologie

Opdracht 2 - Progressive Enhanced Browser Technologies

Maak een demo op basis van een use case. Zorg dat alle gebruikers, met alle browsers, in iedere context minimaal de core functionaliteit te zien/horen/voelen krijgen.

Bouw je demo in 3 lagen, volgens het principe van Progressive Enhancement. Gebruik als enhancement een (hippe, innovatieve, vooruitstrevende) Browser Technologie die je gaat onderzoeken op functionaliteit, toegankelijkheid en (browser) ondersteuning. De meest 'enhanced' versie is super vet, gaaf en prettig om te gebruiken …

Opdracht 2 - Kies een Use Case

1. Ik wil een enquête kunnen invullen over de minor Web Development, met verschillende antwoordmogelijkheden. Als ik de enquête niet afkrijg, wil ik later weer verder gaan met waar ik ben gebleven.
2. Ik wil mijn eigen t-shirt-met-nerdy-tekst kunnen ontwerpen, printen, opslaan, en een volgende keer dat ik de site bezoek kunnen gebruiken.
3. Ik wil tijdens een college aan studenten een poll kunnen voorleggen met over-de-streep stellingen en de resultaten meteen laten zien.
4. Ik wil de scores of tijden van een sportwedstrijd kunnen bijhouden tijdens de wedstrijd en opslaan en doorsturen.
5. Ik wil een notificatie krijgen als mijn favoriete sporter of sportteam heeft gescoord tijdens een wedstrijd, en ik wil een reactie kunnen schrijven bij de melding.
6. Ik wil de routebeschrijving van mijn huis tot aan het Device Lab stap voor stap kunnen zien.
7. Ik wil een serie foto's van mijn vakantie kunnen bekijken, ik wil nieuwe foto's aan de serie kunnen toevoegen en een selectie in een carrousel kunnen bekijken.
8. Heb je zelf een idee? Dat kan, schrijf een use case en kom even overleggen.

Opdracht 2 - Werkwijze

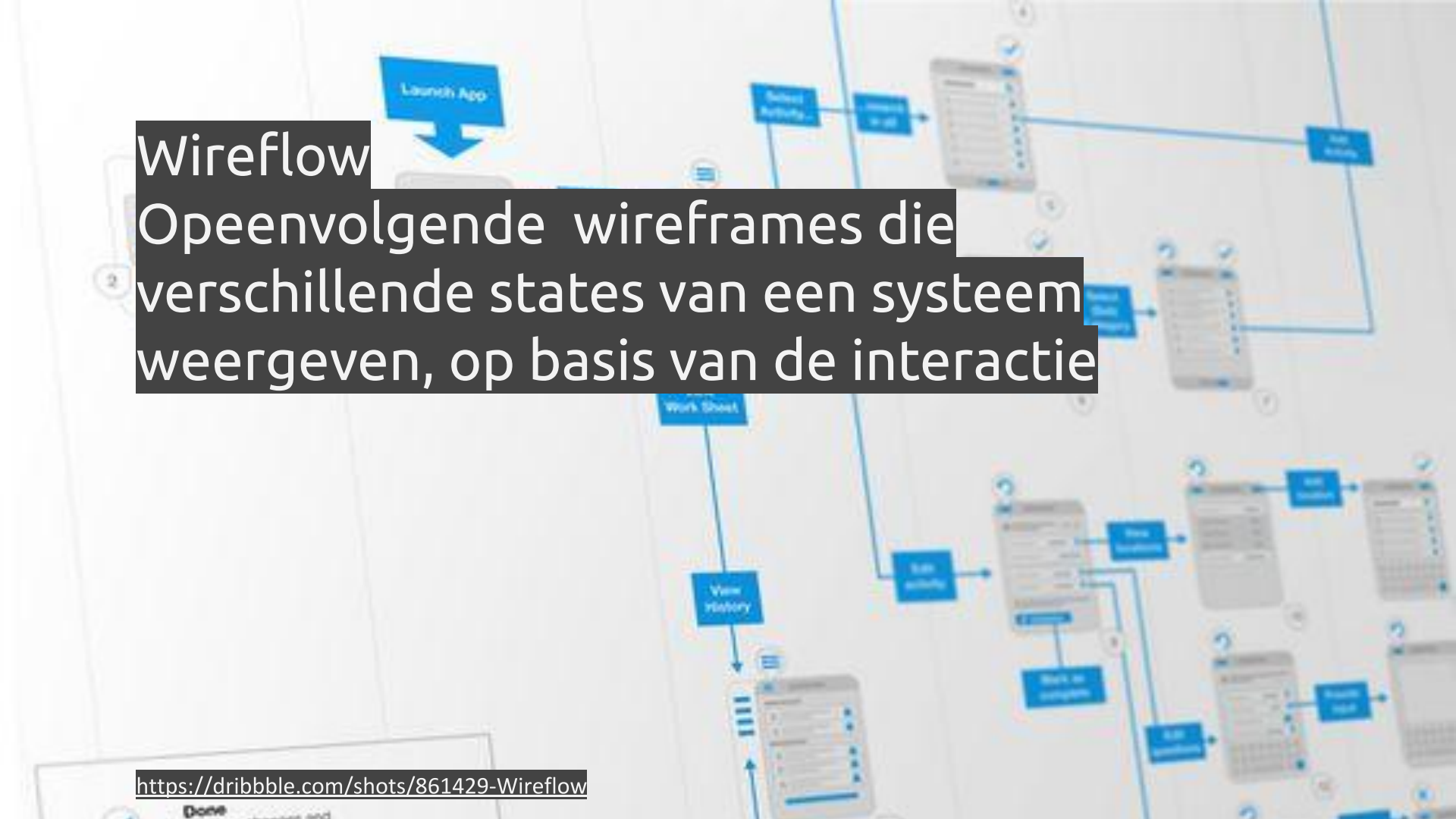
Schets een wireflow met hoe de demo moet gaan werken en hoe het eruit komt te zien. Bepaal de functional, reliable, usable en pleasurable laag *

- Onderzoek voor de functional/reliable laag naar semantische HTML elementen
- Kijk voor de usable laag naar design patterns die je zou kunnen toepassen
- Pleasurable laag? De meest 'enhanced' versie is super vet, gaaf en prettig om te gebruiken.

* Bron: The Role of Enhancement in Web Design
Bron: A Theory of User Delight (functional, reliable, usable, pleasurable)

Wireflow

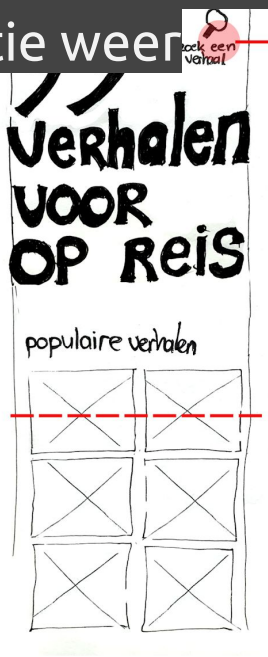
Opeenvolgende wireframes die verschillende states van een systeem weergeven, op basis van de interactie



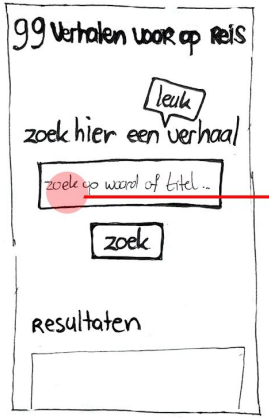
https://dribbble.com/shots/861429-Wireflow

Een Wireflow geeft

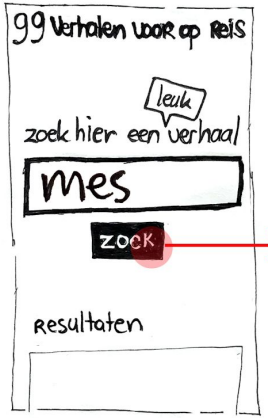
- alle mogelijke output,
- de belangrijkste userflow
- en de interactie weer



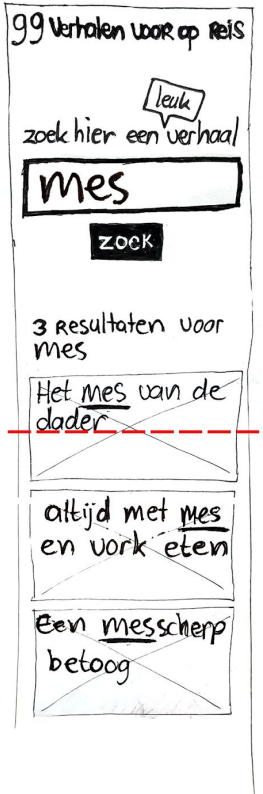
2. Zoek scherm



3. Zoekactie uitvoeren



4. Zoekresultaten



2.1 Zoekformulier focus

</section>