



Minor Web Design & Development

Web Apps From Scratch

**In theory, there is no difference
between theory and practice.**

In practice, however, there is.

Inhoud

1. Hello
2. Introductie
3. Programma
4. Hello again
5. Opdracht (individueel)
6. Opdracht (team)
7. Werkwijze - github templates - even maken
8. WAFS : development cycle - UI stack intro - schetsen

Inhoud

1. **Hello**
2. Introductie
3. Programma
4. Hello again
5. Opdracht (individueel)
6. Opdracht (team)
7. Werkwijze
8. WAFS



Bas de Greeuw

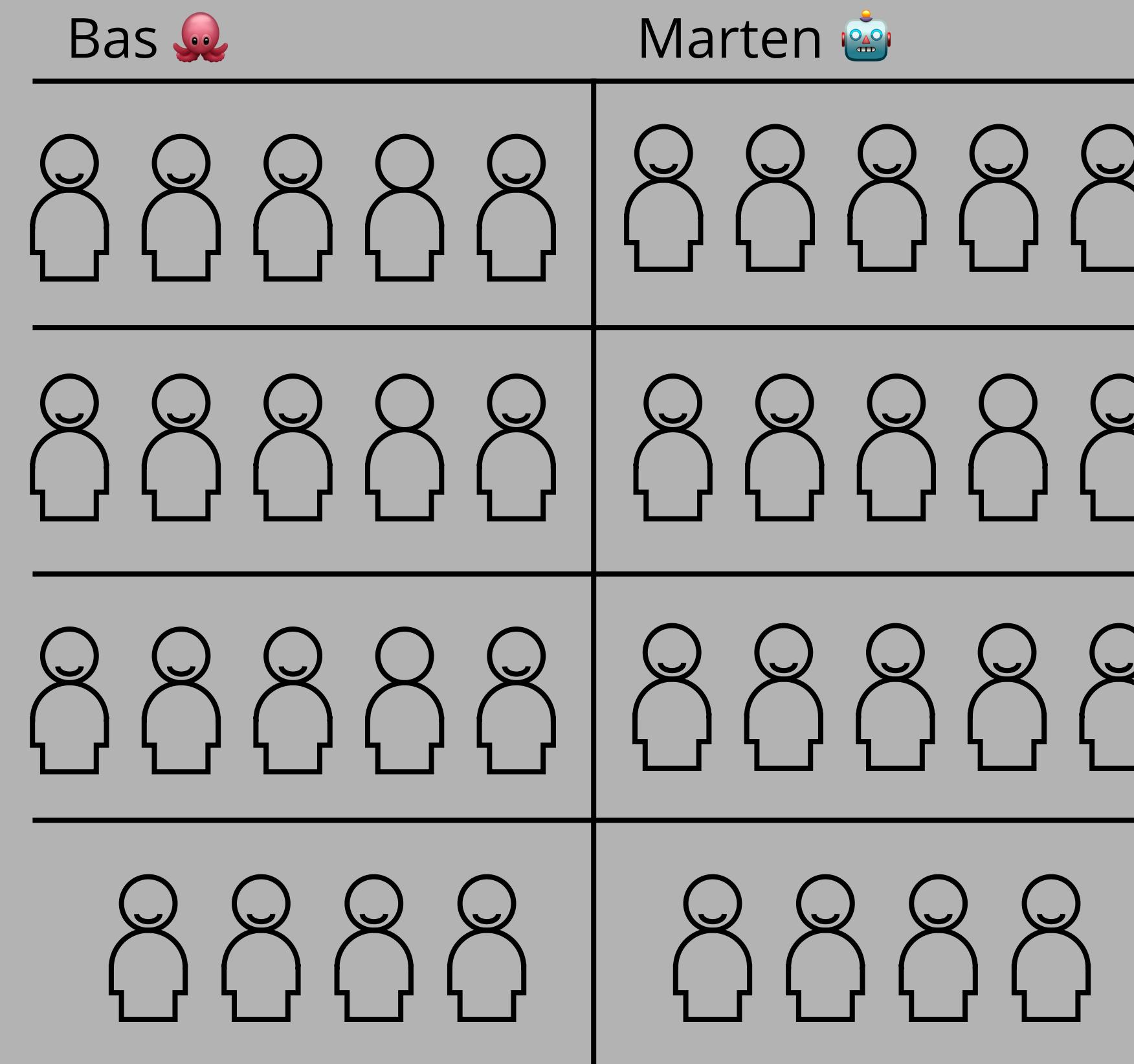
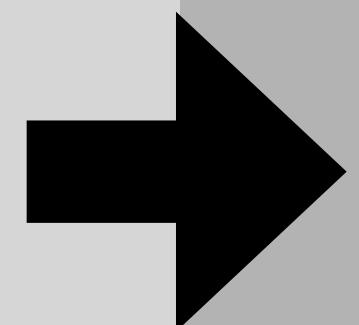
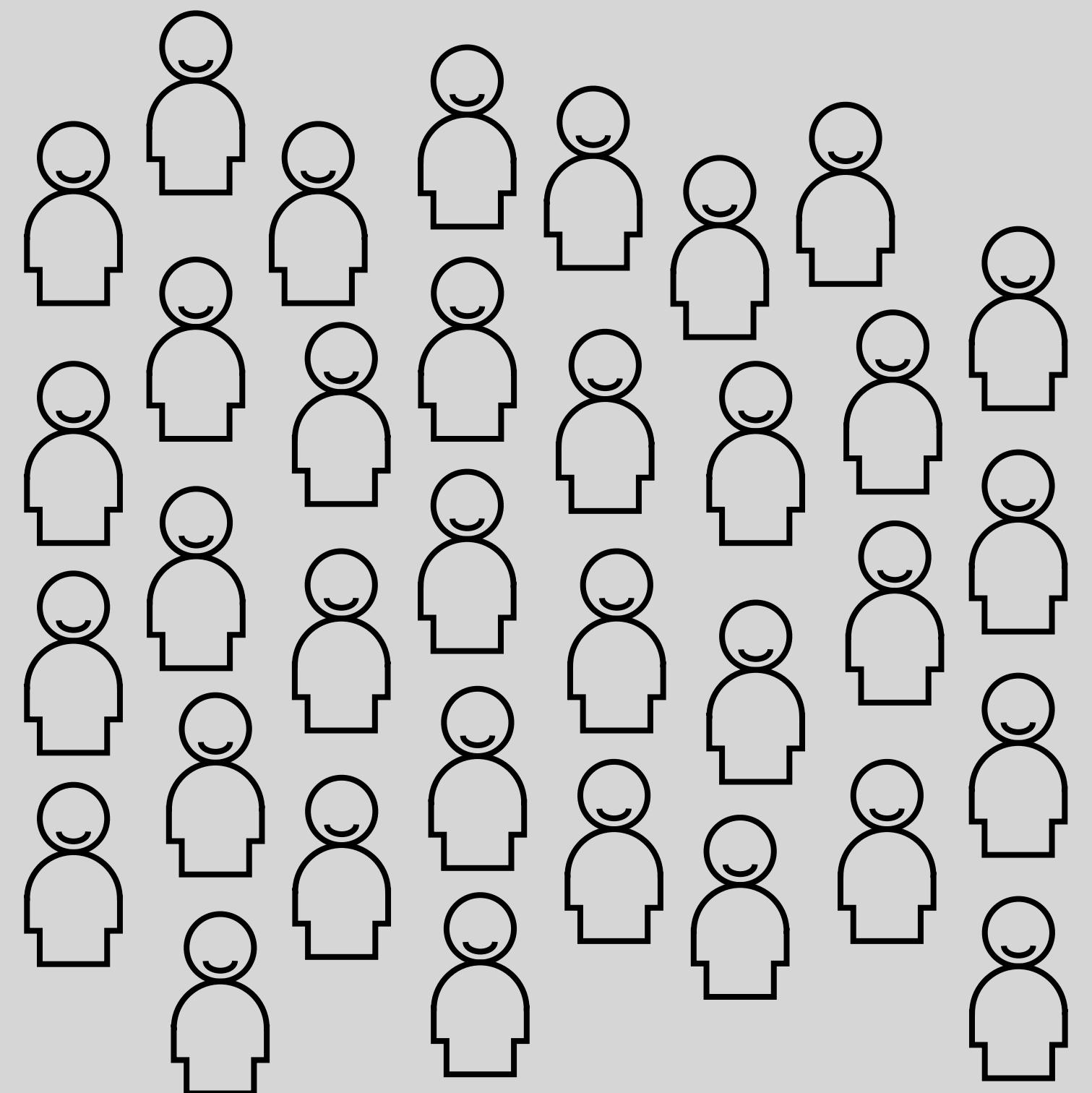
@bazottie



Marten Moolenaar

@mrtnm

± 38 studenten → 8 teams
(over 2 squads)



Inhoud

1. Hello
2. **Introductie**
3. Programma
4. Hello again
5. Opdracht (individueel)
6. Opdracht (team)
7. Werkwijze
8. WAFS

Wat we gaan doen

Het eindresultaat is een modulaire webapp (SPA) met één pagina. En dat twee (2) keer.

Zonder frameworks, preprocessors of onnodige bibliotheken, maar zoveel mogelijk met standaard HTML, CSS en JavaScript.

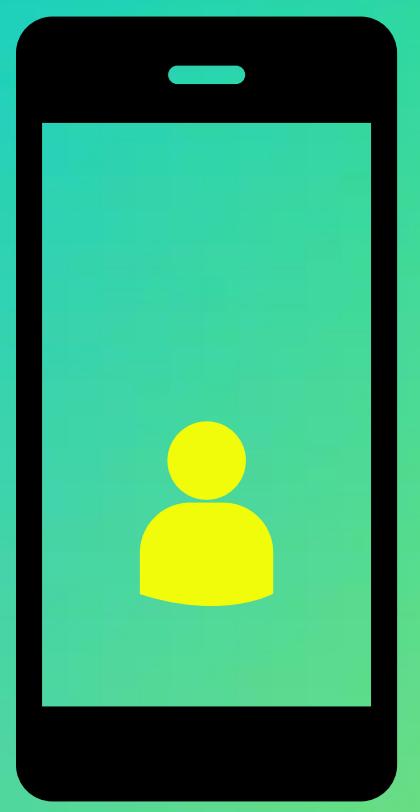
Opleveren:

- Een web app per individu
- Een web app per team
- Een procesverslag

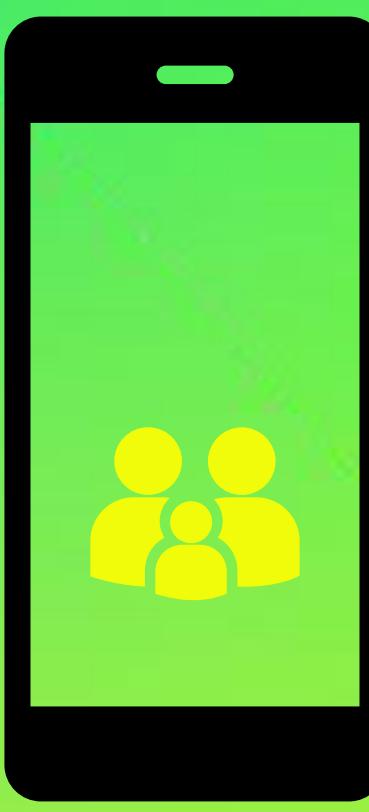
Experimenten worden zeer gewaardeerd, ook als ze niet (helemaal) slagen.

Met de opgedane kennis ben je in staat interactieve prototypes te bouwen en krijg je een beter inzicht in hoe API's, frameworks en bibliotheken werken.

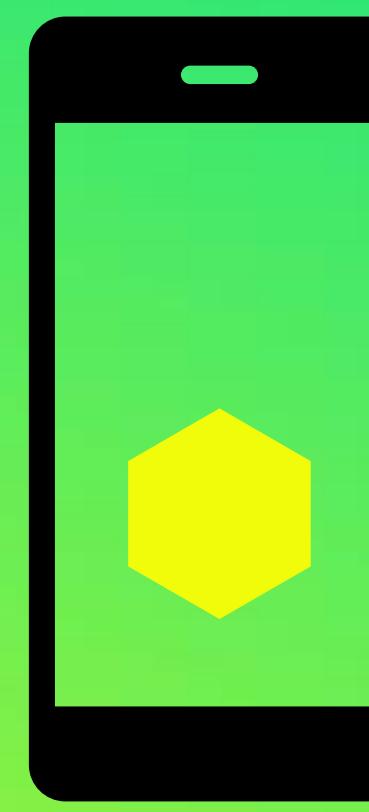
Je werkt (bijna) twee weken aan de opdracht. Ondertussen wel je proces vastleggen.



{ }



{ }



HTML



CSS



JS



HTML



Nette, semantisch
correcte HTML

CSS



Nette, leesbare CSS

JS



Vanilla JavaScript

Zonder frameworks, preprocessors of onnodige bibliotheken.

Get to work, b*tch!

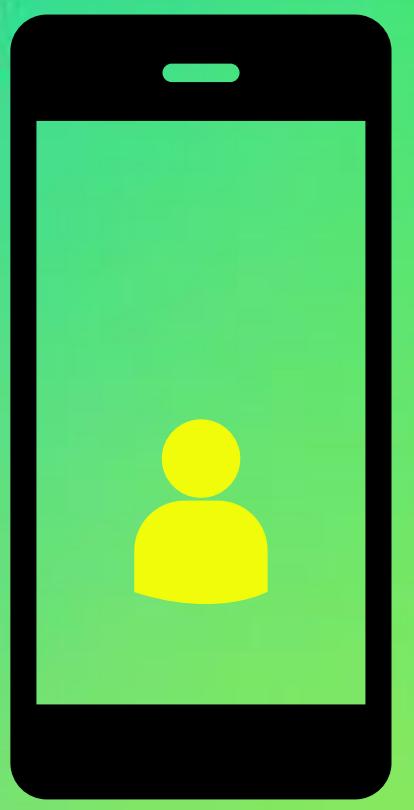
Britney S.

Overall criteria

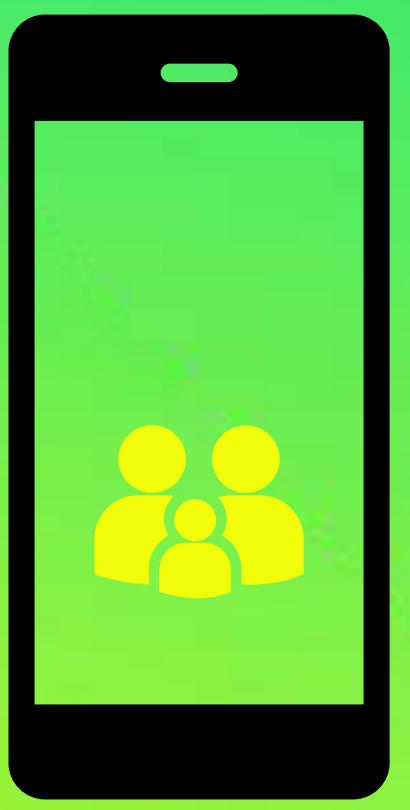
- Gerealiseerd met 'nette' vanilla HTML, CSS en Javascript,
- Diverse content,
- Minimaal één tot in detail uitgewerkte micro-interactie,
- Minimaal één data-connectie (API call),
- Bevat de juiste data voor integratie met de team app,
- Toegankelijk,
- Professionele vormgeving (en styling),
- Volledig responsive,
- Gedocumenteerd

Pas de richtlijnen toe die je hebt geleerd voor het schrijven van cleane, elegante code:

responsive, toegankelijk, custom properties, valide, semantisch, comments, relatieve eenheden, grid/flex, naming conventions, enz...



</>
{ }



Opleverproducten

Wat:

- Twee apps
- Voortgangsrapportage
- Documentatie van de app



Waar:

- Op GitHub
- In een zip-bestand op DLO



Wanneer:

De dag voor je eindgesprek voor 23:59

Herkansing:

6 mei - zelfde proces

Voortgangsrapport

Je legt je voortgang dagelijks vast in een readme.md die je tegelijkertijd met je werk synchroniseert op GitHub.

- De opdracht en de opties die je kiest
- Waar liggen jouw (grote) uitdagingen?
- Voeg schets(en) van het ontwerp toe.
- Eventueel ook een eerste breakdown.
- Voortgang in tekst, code en afbeeldingen.
- Wat ging vlot en wat was uitdagend?
- Welke experimenten heb je uitgevoerd die 'mislukt' zijn?
- Nieuwe inzichten.
- De uitdagingen voor het vervolg.

Bronnen

Vermeld je bronnen in de code en vermeld ze allemaal in de readme.md.

Nb. Wees specifiek ('css-tricks' is bijvoorbeeld niet specifiek genoeg - voeg het artikel/de URL toe).

Nb. Inclusief Co-pilot, ChatGPT en andere AI.

Eindgesprek

- Bespreek het eindresultaat
- Wat ging vlot, wat was uitdagend, en waar ben je het meest trots op?
- Wat je verder nog wil ontdekken

Leerdoelen

- **Procesmatig samenwerken in en met teams**
- **Een app bouwen van idee tot uitwerking**

Een volledige realiseren van begin tot eind met aandacht voor toegankelijkheid, responsiveness, professionele uitstraling en details.

- **Prototypen in 'nette' vanilla HTML, CSS en JS.**

Gebruikersinterface

Je ontwerpt, bouwt en test de gebruikersinterface door ontwerprincipes toe te passen

Codestructuur

Je schrijft modulaire, consistente en efficiënte HTML-, CSS- en JavaScript-code door structuur en best practices toe te passen. Je beheert de status van de applicatie en de gebruikersinterface.

Databeheer en -uitwisseling

Je begrijpt hoe je met een externe API kunt werken met behulp van asynchrone code. Je kunt gegevens ophalen, manipuleren en dynamisch omzetten naar gestructureerde html.

Project

Je app werkt en is gepubliceerd op GitHub-pages. Je project is grondig gedocumenteerd in het README.md-bestand in je repository en bijvoorbeeld een wiki.

Werkwijze

- Informatie over alle vakken staat op DLO
- Voorbeeldcode op GitHub en CodePen
- Digitale communicatie op Teams
- Elkaar helpen in de klas en op Teams*
- Opdrachten inleveren op DLO en GitHub

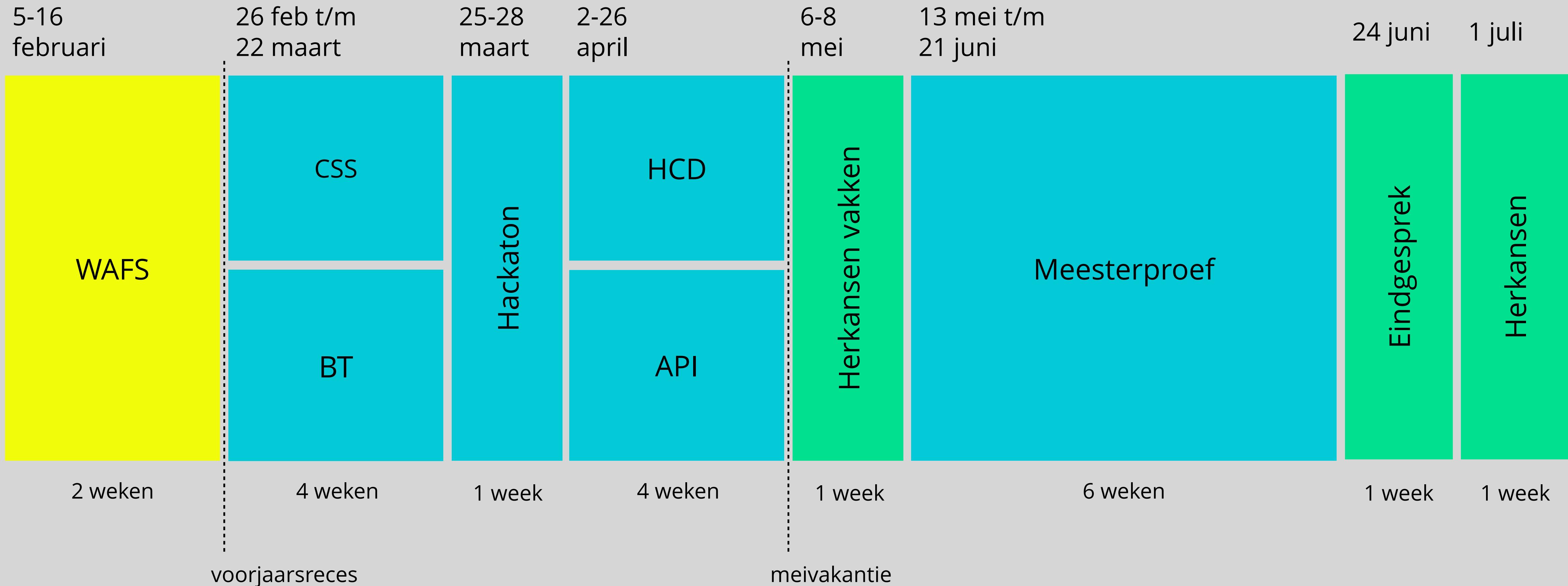
* Tip: Discord is ook een goeie plek.

Inhoud

1. Hello
2. Introductie
- 3. Programma**
4. Hello again
5. Opdracht (individueel)
6. Opdracht (team)
7. Werkwijze
8. WAFS

Programma

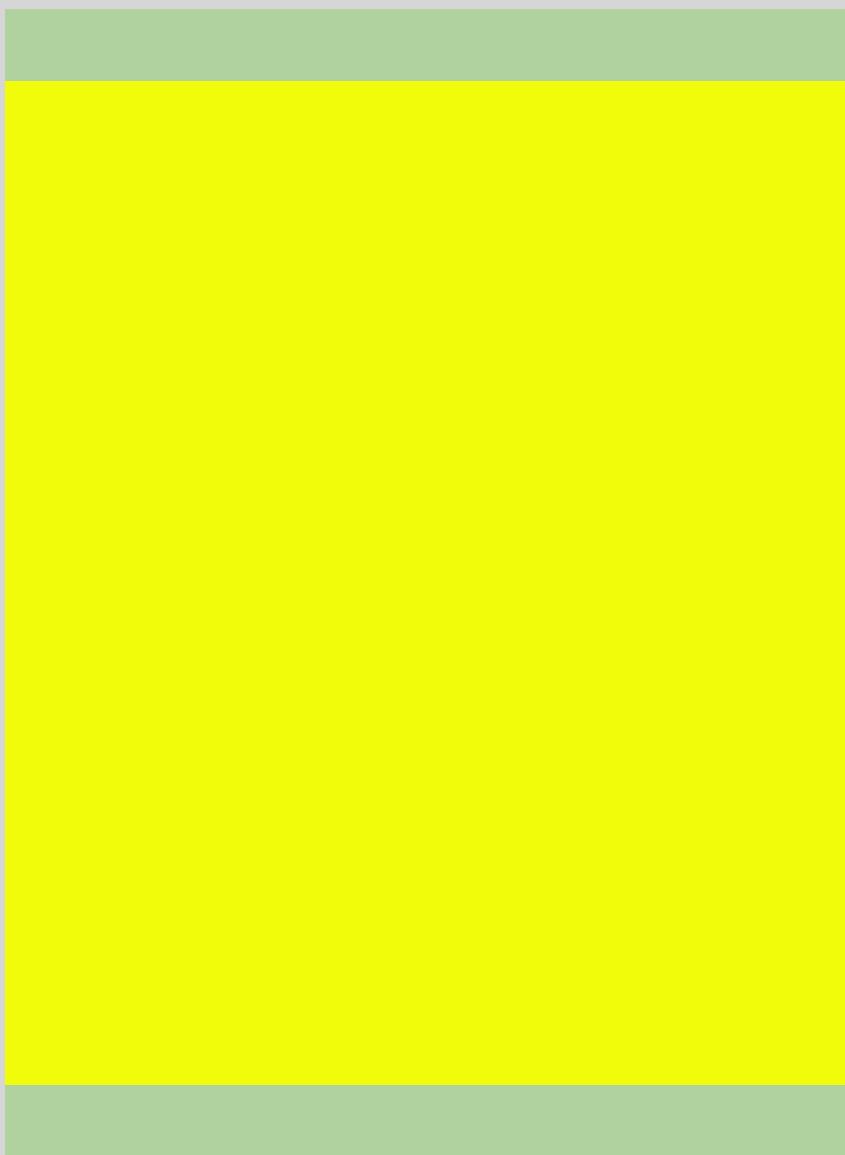
TTH04B08 - Achter het gordijn



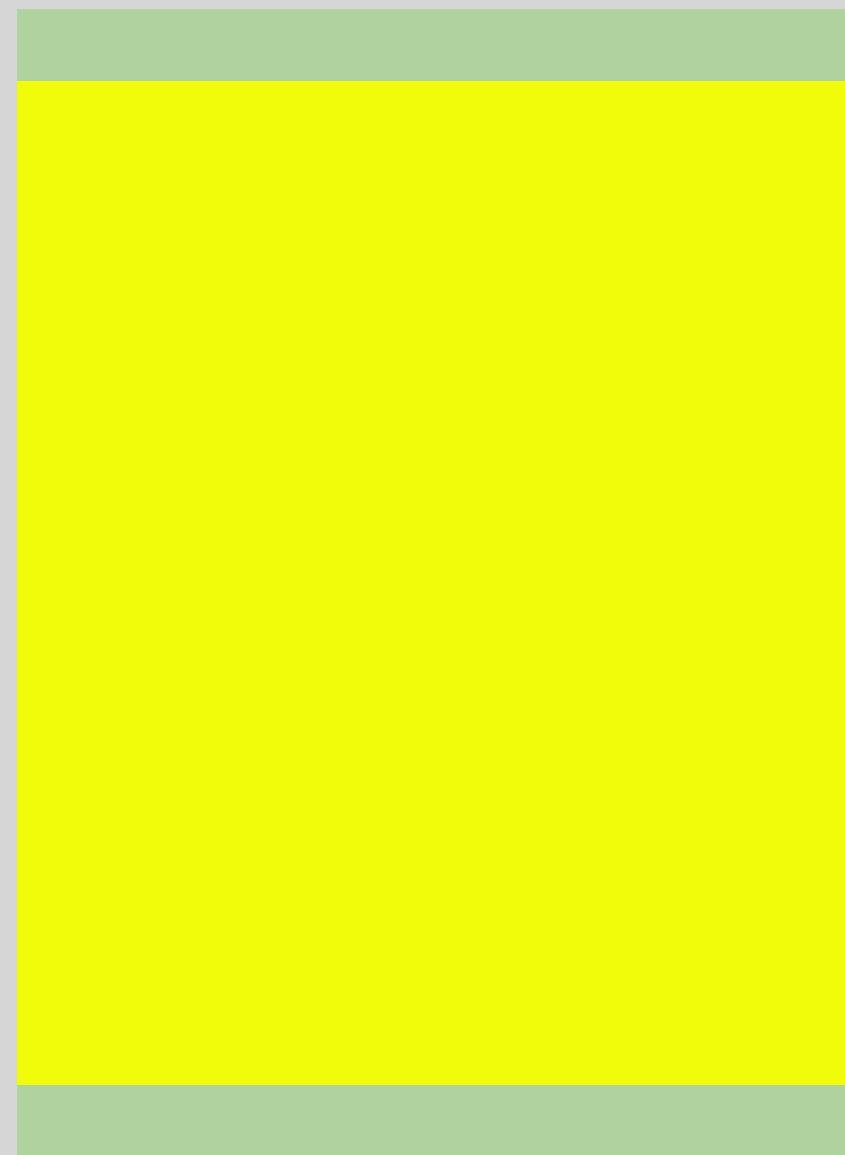
Web Apps From Scratch

Week 1

Maandag



Dinsdag

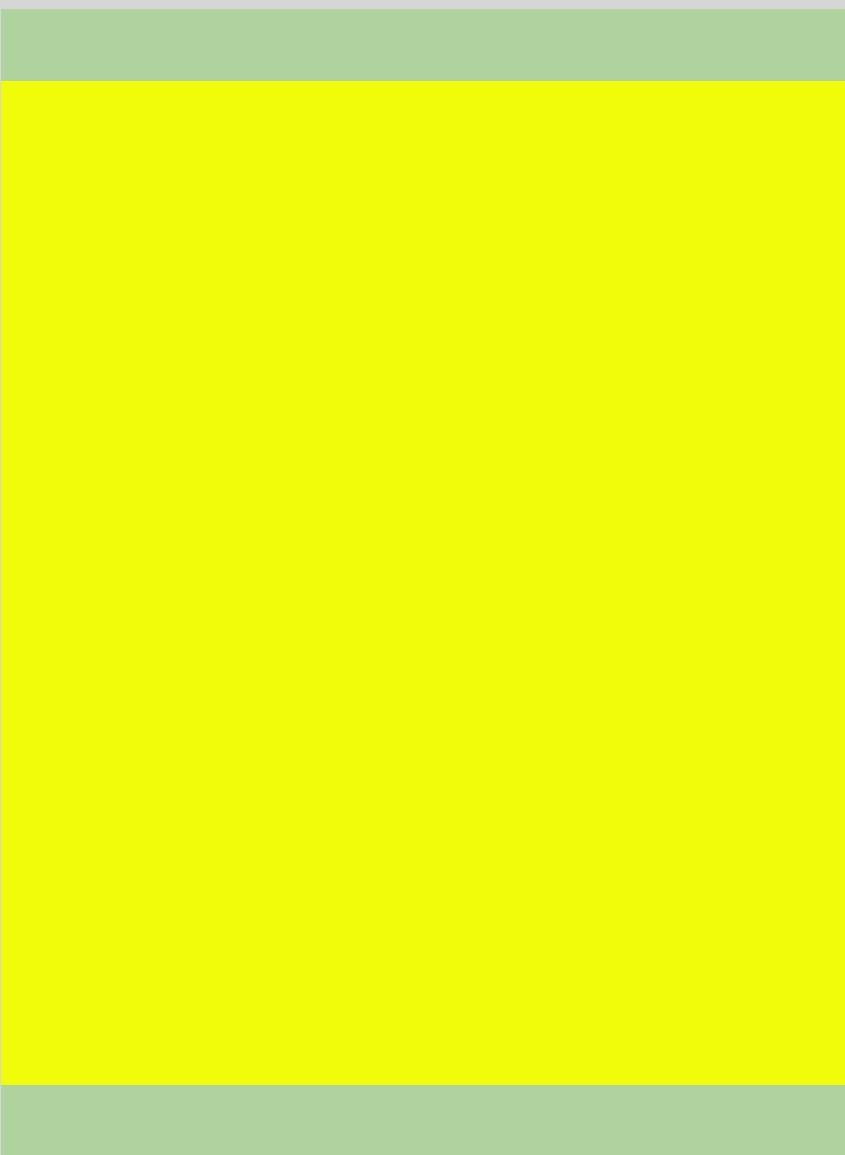


Woensdag



Weekly Nerd

Donderdag



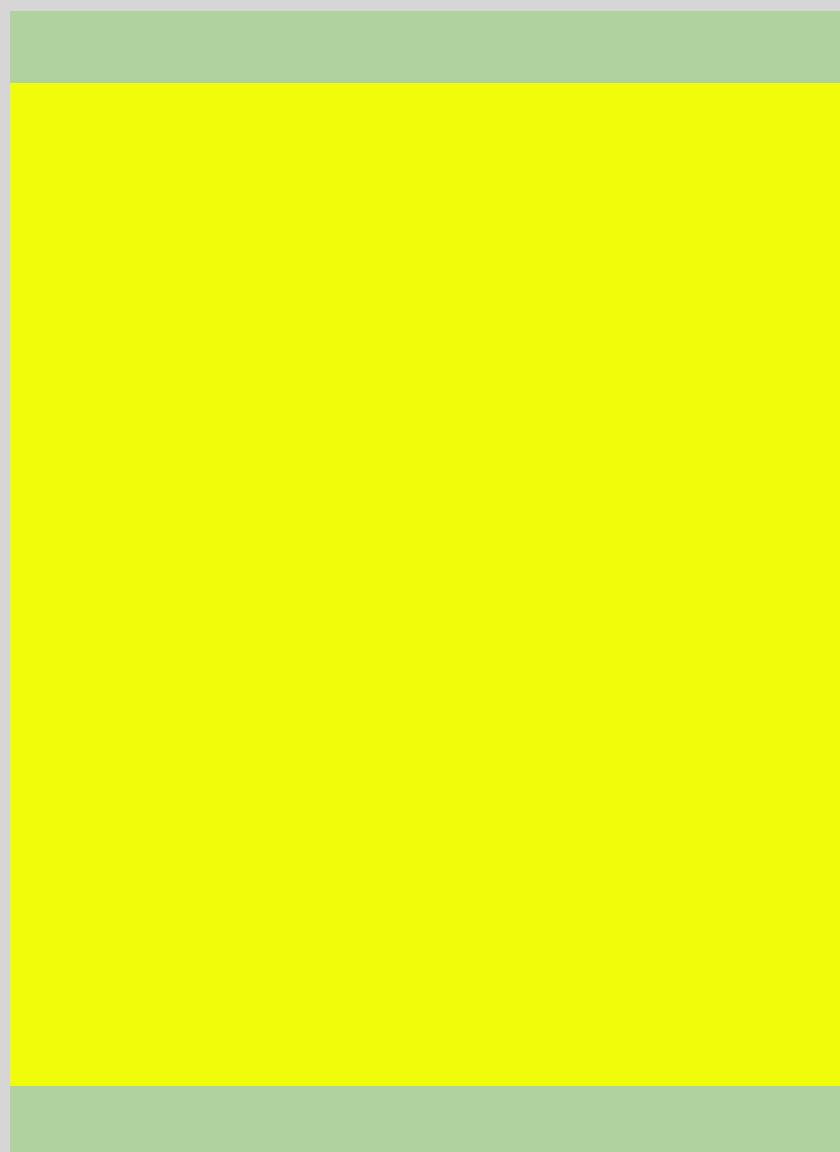
Vrijdag



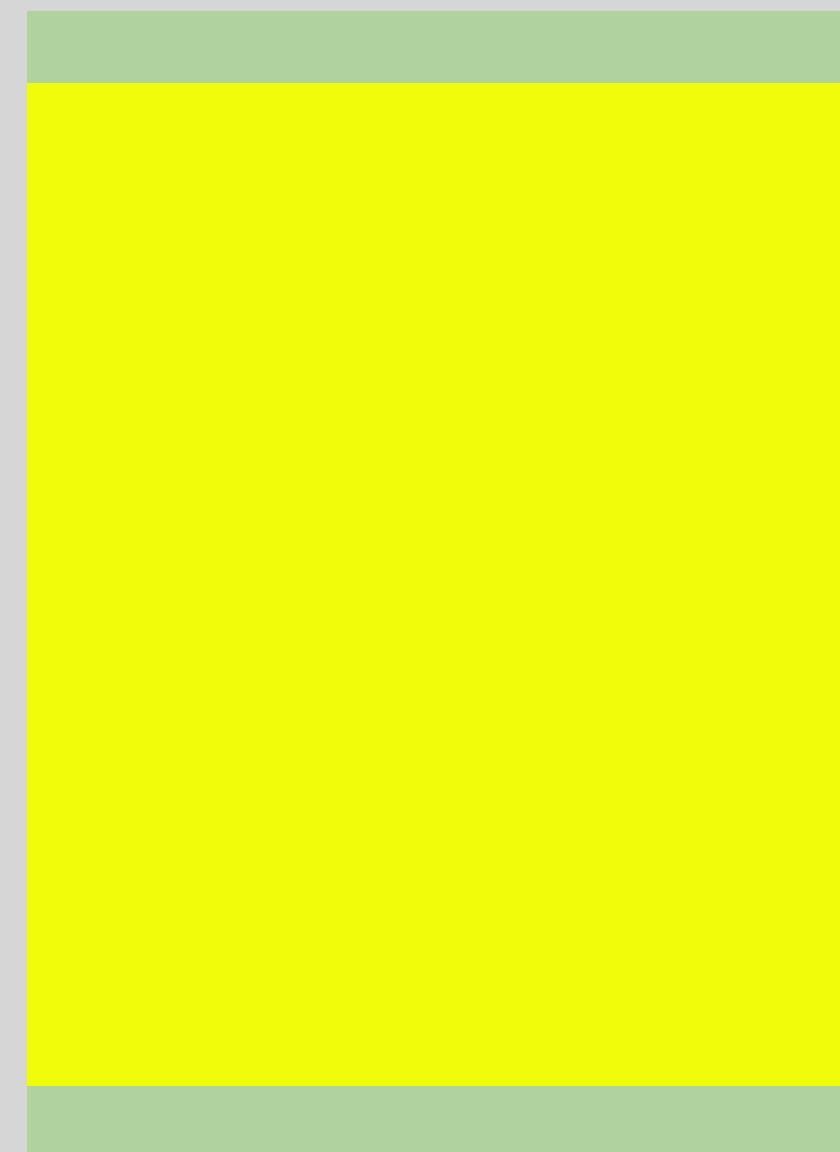
Web Apps From Scratch

Week 2

Maandag



Dinsdag



Woensdag



Donderdag



Vrijdag



23:59 Inleveren

Elke dag

- **9:30 Stand-up** - wat ga ik doen vandaag, wat ga ik opleveren, welke input is noodzakelijk, waar heb ik hulp nodig
- **16:00 Retro en groupe** - waar liep je groepje tegenaan? Wat zijn toffe ideeën en ontdekkingen?

Op de dag hebben de **college/workshops**

Werken aan de opdracht: Analyseren, onderzoeken, ontwerpen, bouwen, integreren, testen.

Workshops

- Bij voorbeeld
 - Git (en hup!)
 - UI patterns
 - Components
 - Data ophalen - data vs API vs JS.
 - ES modules
 - Opslag - local storage, external services



Eindgesprek

- **Donderdag 15 februari**
- **Deadline:** de dag voor het gesprek voor 23:59 uur.

Je **levert je werk en procesverslag in** op GitHub **én** in een zip op DLO.

Tijdens het gesprek laat je je werk zien (ook op je telefoon), vertel je waar je blij mee bent, wat lastig is/was en stelt de docent vragen over je code.

Herkansing - idem (indien nodig).

Inhoud

1. Hello
2. Introductie
3. Programma
- 4. Hello again**
5. Opdracht (individueel)
6. Opdracht (team)
7. Werkwijze
8. WAFS

Even kennismaken

- 1. Een array**
- 2. Voorkeuren**
- 3. Ren je rot**
- 4. Wat weet jij dan?**

1. Een array op naam

Een array op naam

- Ga je een nieuwe array maken?
- Zelf bepalen.
- Kan dit sneller?
- Hoe doen mensen het vs computers?

2. Voorkeuren

levend
Kahoot!

potlood

giro

spaces

tabs

Flex

Grid

const | let

var

<article />

<section />

function

class

object-oriented

functional

procedural

Even voorsorteren op kennis en voorkeuren (kahoot survey)

- Flex of Grid?
- const, let, var?
- Article / Section?
- function / class?
- Object-oriented / Procedural / Functional
- Potlood / Miro
- Spaces / Tabs

3. Ren... je... rot!

(of wandel peinzend)

A

Junior. Als je FED en TECH "stiekem best moeilijk" vond.

C

Senior. Als je stage hebt gelopen (of erger, werkt).

B

Medior. Daar ergens tussenin.



4. Wat weet jij dan?

levend
Kahoot!

(neee... niet weer)

Even opwarmen

Op welke plek komt three?

Op de tweede plek

```
<ul>
  <li>one</li>
  <li>two</li>
  <li>three</li>
  <li>four</li>
</ul>
```

```
ul {
  display: flex;
  justify-content: flex-start;
}
li:nth-of-type(4) {
  order: 2;
}
li:nth-of-type(2) {
  order: 4;
}
li:nth-of-type(3) {
  order: 2;
}
```

Op de derde plek

Even opwarmen

Op welke plek komt three?

one three four two

Op de tweede plek

```
<ul>
  <li>one</li>
  <li>two</li>
  <li>three</li>
  <li>four</li>
</ul>
```

```
ul {
  display: flex;
  justify-content: flex-start;
}
li:nth-of-type(4) {
  order: 2;
}
li:nth-of-type(2) {
  order: 4;
}
li:nth-of-type(3) {
  order: 2;
}
```

Op de derde plek

Specificity

Welke kleur krijgt `#zalJeEmHebben` ?

```
<body class="bodyclass">
  <section class="sectionClass">
    <p class="parentClass">
      <input id="zalJeEmHebben" />
    </p>
  </section>
</body>
```

green

```
#zalJeEmHebben {
  color: green;
}
.bodyClass .sectionClass .parentClass [id="zalJeEmHebben"] {
  color: yellow;
}
```

yellow

Specificity

Welke kleur krijgt `#zalJeEmHebben` ?

```
<body class="bodyclass">
  <section class="sectionClass">
    <p class="parentClass">
      <input id="zalJeEmHebben" />
    </p>
  </section>
</body>
```

green

```
#zalJeEmHebben {
  color: green; /* 1-0-0 - WINS!! */
}
.bodyClass .sectionClass .parentClass [id="zalJeEmHebben"] {
  color: yellow; /* 0-4-0 */
}
```

yellow

What's your type?

```
let myTest = 42 / "someOne";  
  
// outputs: NaN  
console.log( myTest );  
  
// what does this output?  
console.log( typeof myTest );
```

undefined

NaN

strikvraag

What's your type?

```
let myTest = 42 / "someOne";  
  
// outputs: NaN  
console.log( myTest );  
  
// what does this output?  
console.log( typeof myTest );
```

Console

"number"

strikvraag

Cascading order

Bij het bepalen van de waarde in de casacade op basis van **origin** en **importance**, welke volgorde is bepalend (van laag naar hoog)?

1. de stylesheet van de **user-agent (browser)**
2. de stylesheet van de **user**
3. de stylesheet van de **author (developer)**

1. de stylesheet van de **author (developer)**
2. de stylesheet van de **user**
3. de stylesheet van de **user-agent (browser)**

Inhoud

1. Hello
2. Introductie
3. Programma
4. Hello again
5. **Opdracht (individueel)**
6. Opdracht (team)
7. Werkwijze
8. WAFS

Opdracht

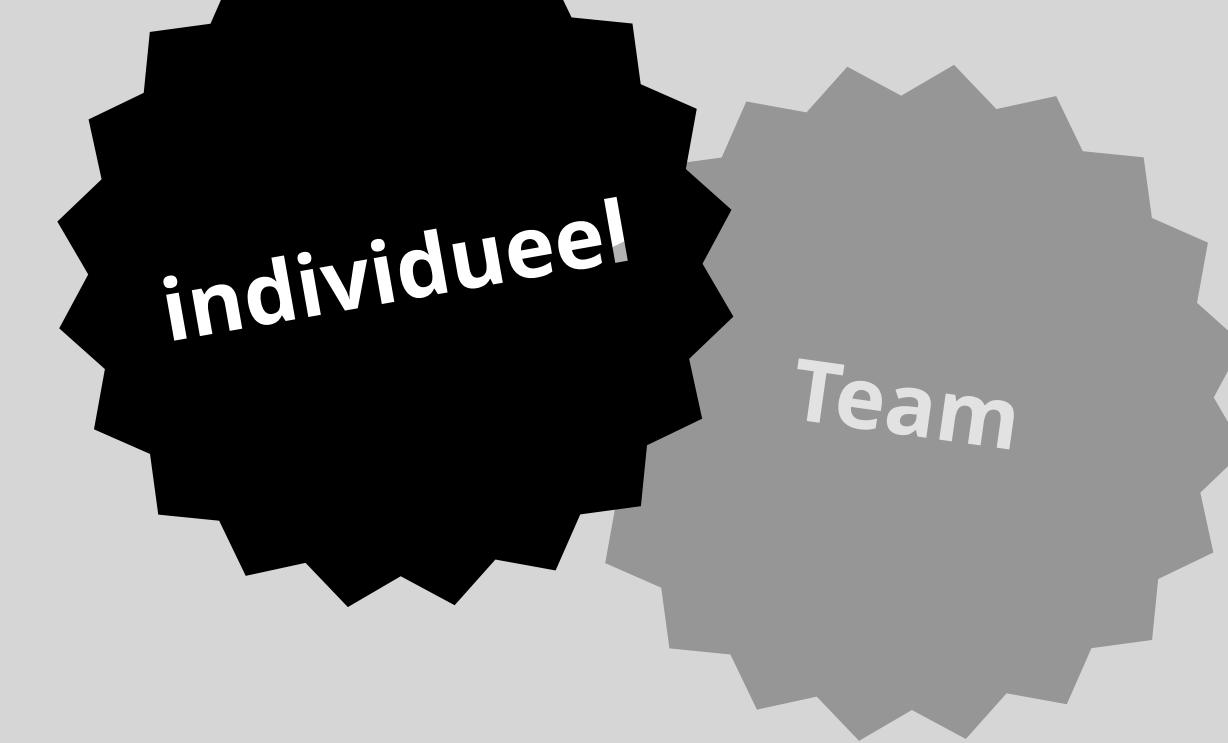
Maak een eigen web app met "persoonlijke" informatie.

Zorg ervoor dat de app volgens de afgesproken team-richtlijnen de vereiste data beschikbaar stelt.

Opleveren:

- Een web app
- Documentatie (in code, readme.md)
- Een procesverslag

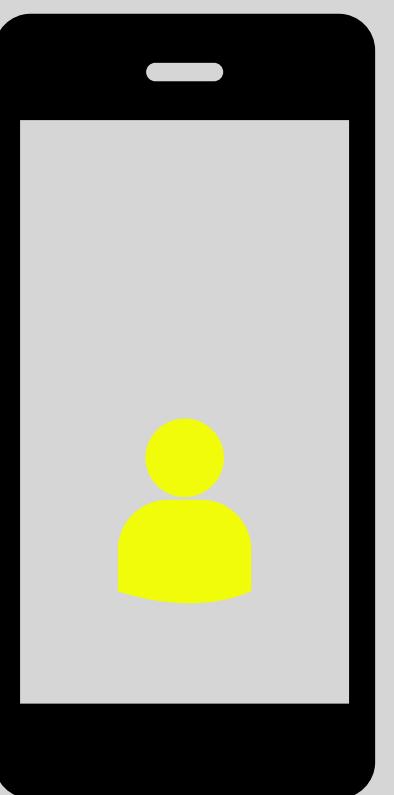
Gebruik nette vanilla HTML, CSS en JavaScript.



Personlijke web app

Wat dan?

- **Verplichte data** TBD
- **TBD Verschillende componenten.** Menu - dragers - card + avatar - carrousel - form - pagination - toast - etc
- **Data van elders.** xhr (fetch it)
- **Uitwisselingsinformatie voor de team web app.**
Microdata/structured data (json of...)
- **Eigen leukigheid**



Make it work
Make it right
Make it fast pretty

≈ Kent Beck

Inhoud

1. Hello
2. Introductie
3. Programma
4. Hello again
5. Opdracht (individueel)
- 6. Opdracht (team)**
7. Werkwijzen
8. WAFS

Opdracht

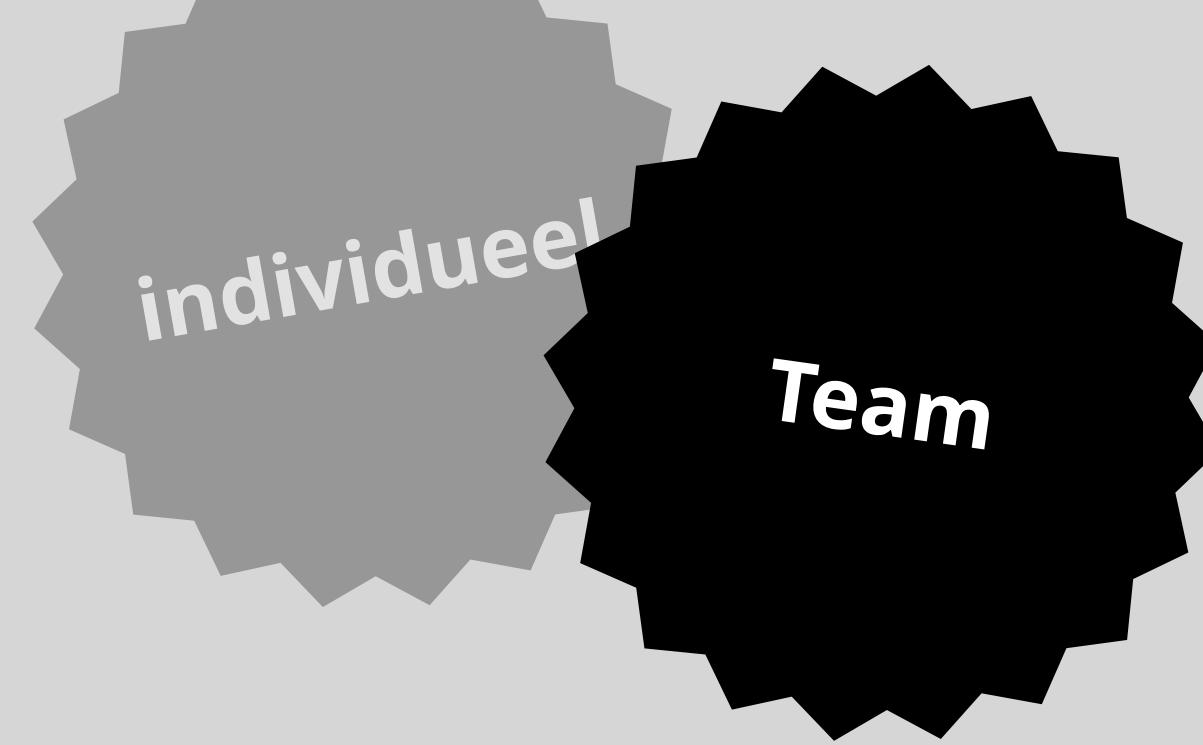
Maak een team app met de belangrijkste info van de teamleden.

**Haal de teamledeninfo op via data requests.
Zorg ervoor dat de app volgens de afgesproken richtlijnen data beschikbaar stelt voor de Squad app.**

Opleveren:

- Een web app
- Documentatie (in code, readme.md)
- Een procesverslag

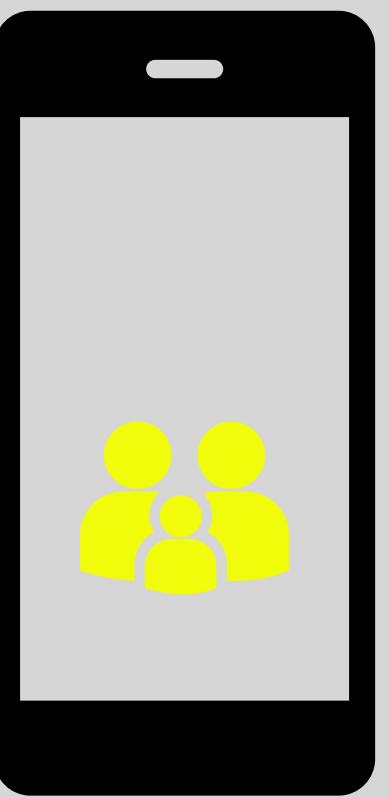
Gebruik nette vanilla HTML, CSS en JavaScript.



Team web app

Wat dan?

- **Verplichte data TBD**
- **Een soorter en/of zoekfunctie**
- **Data van de persoonlijke apps.** xhr (fetch it)
- **Uitwisselingsinformatie voor de Squad app.**
Microdata/structured data (json of...)
- ...



Waarom moeilijk doen als het samen kan?

Loesje

Inhoud

1. Hello
2. Introductie
3. Programma
4. Hello again
5. Opdracht (individueel)
6. Opdracht (team)
7. **Werkwijze**
8. WAFS

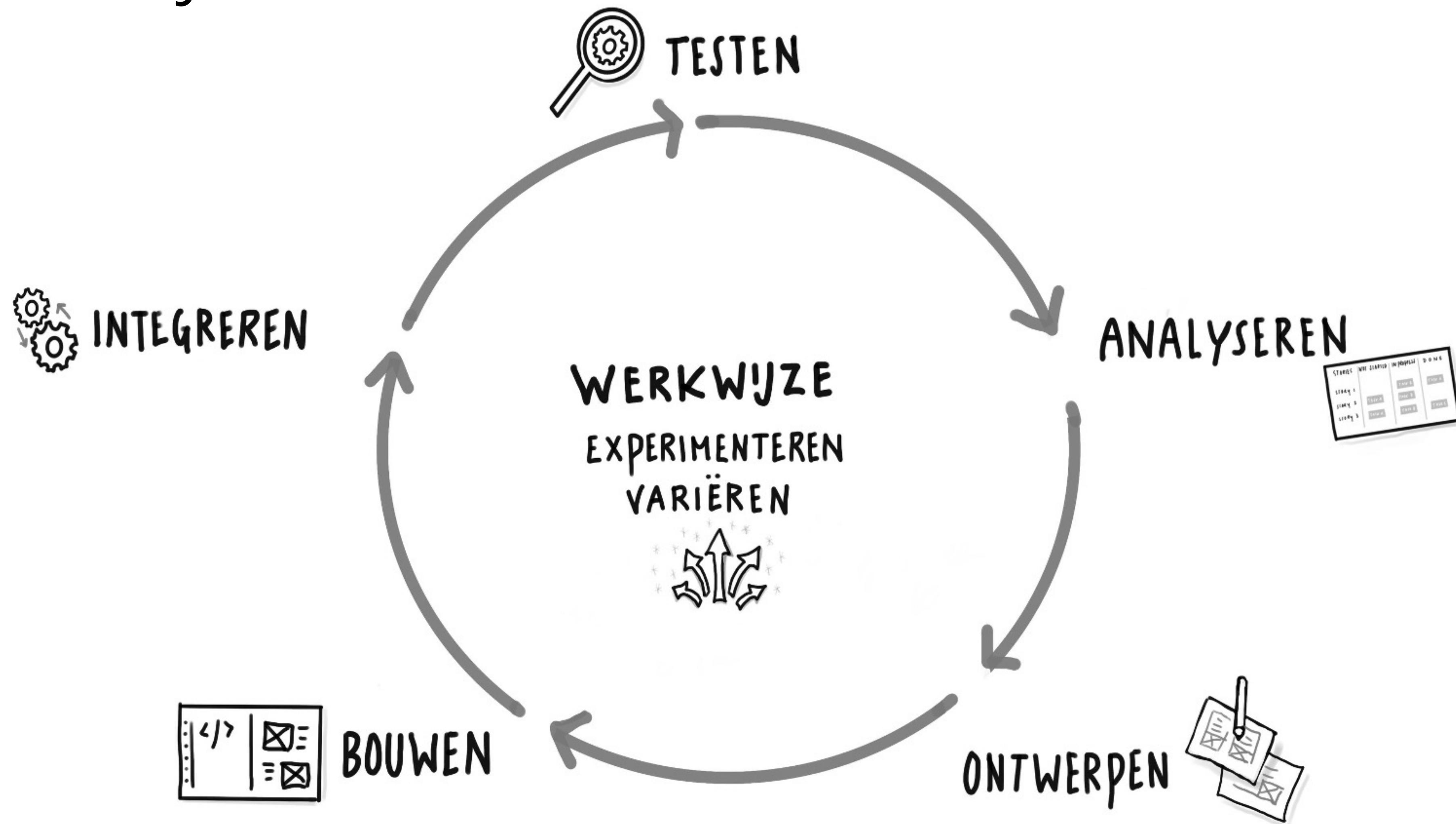
Werkwijze

Pak je laptop

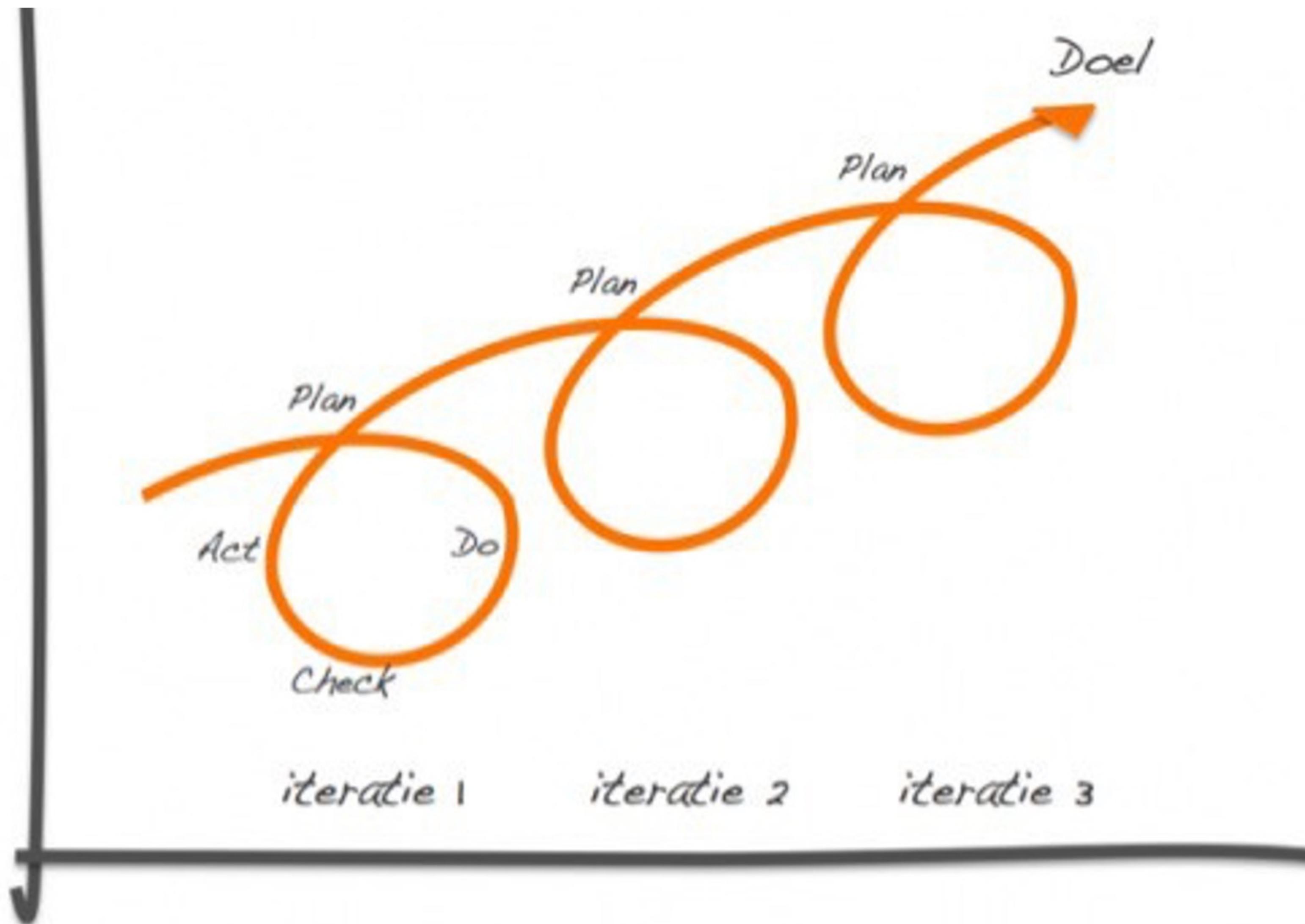
Inhoud

1. Hello
2. Introductie
3. Programma
4. Hello again
5. Opdracht (individueel)
6. Opdracht (team)
7. Werkwijze
- 8. WAFS**

Dev Life Cycle



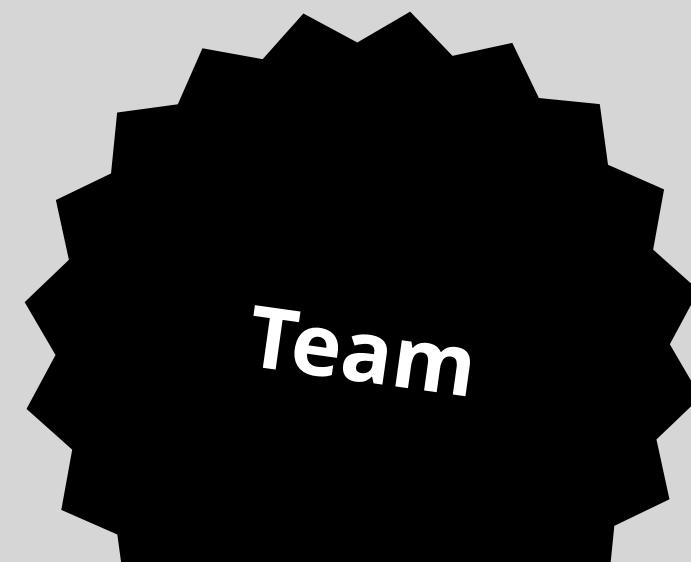
Iteratief werken



De specs

All together now

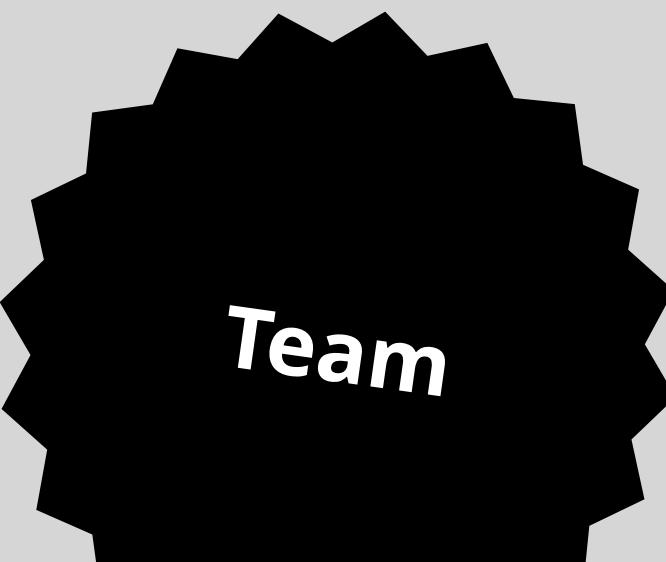
- **Bedenk per team welke data ALLE apps gemeenschappelijk moeten hebben. Denk groot.**
10 min.
- **Regroup**



Wat heeft iedereen?

All together now

- **Team naam**



User interface



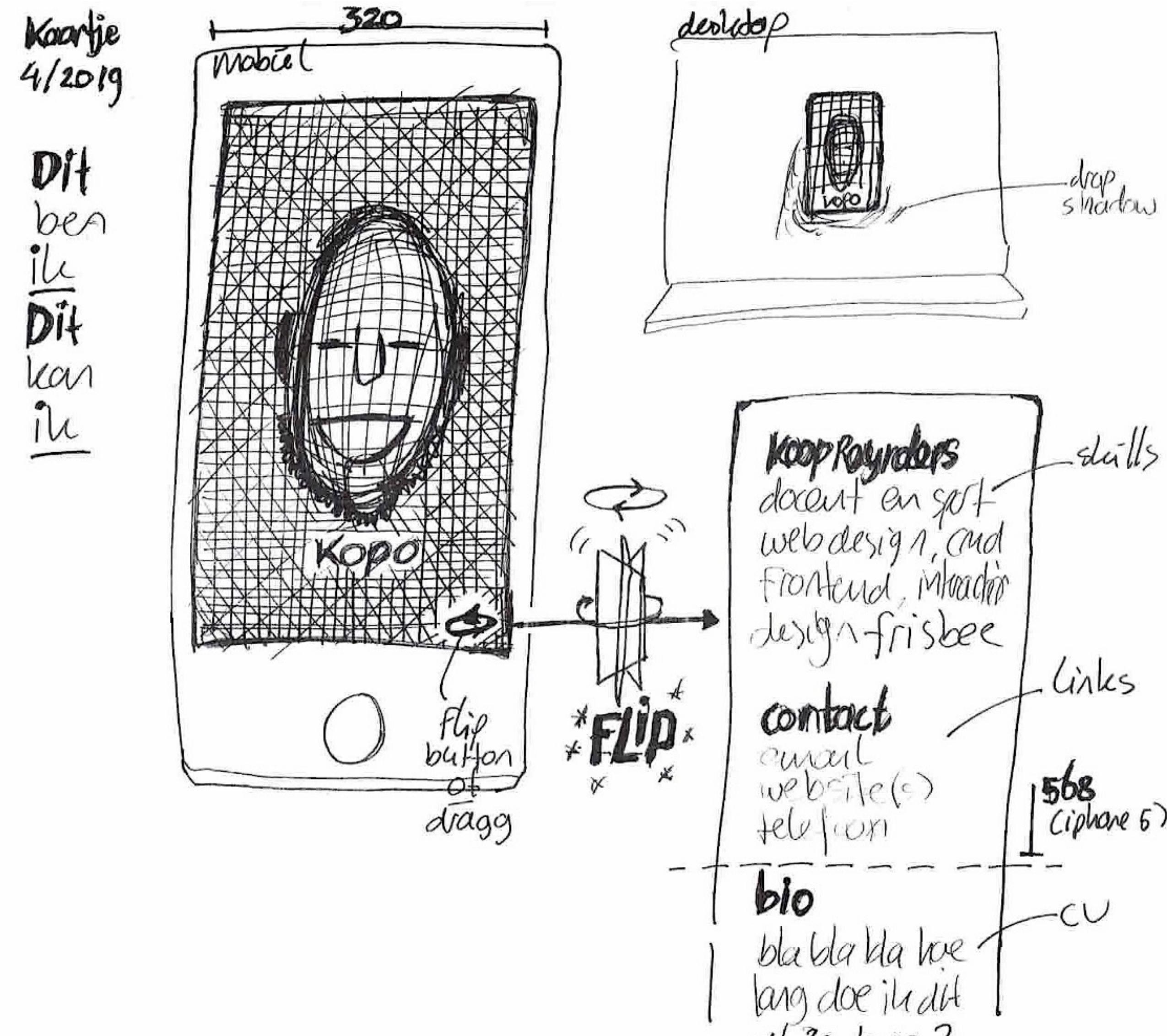
**Don't worry about stating the
obvious... the obvious almost
never is.**

Joshua Porter

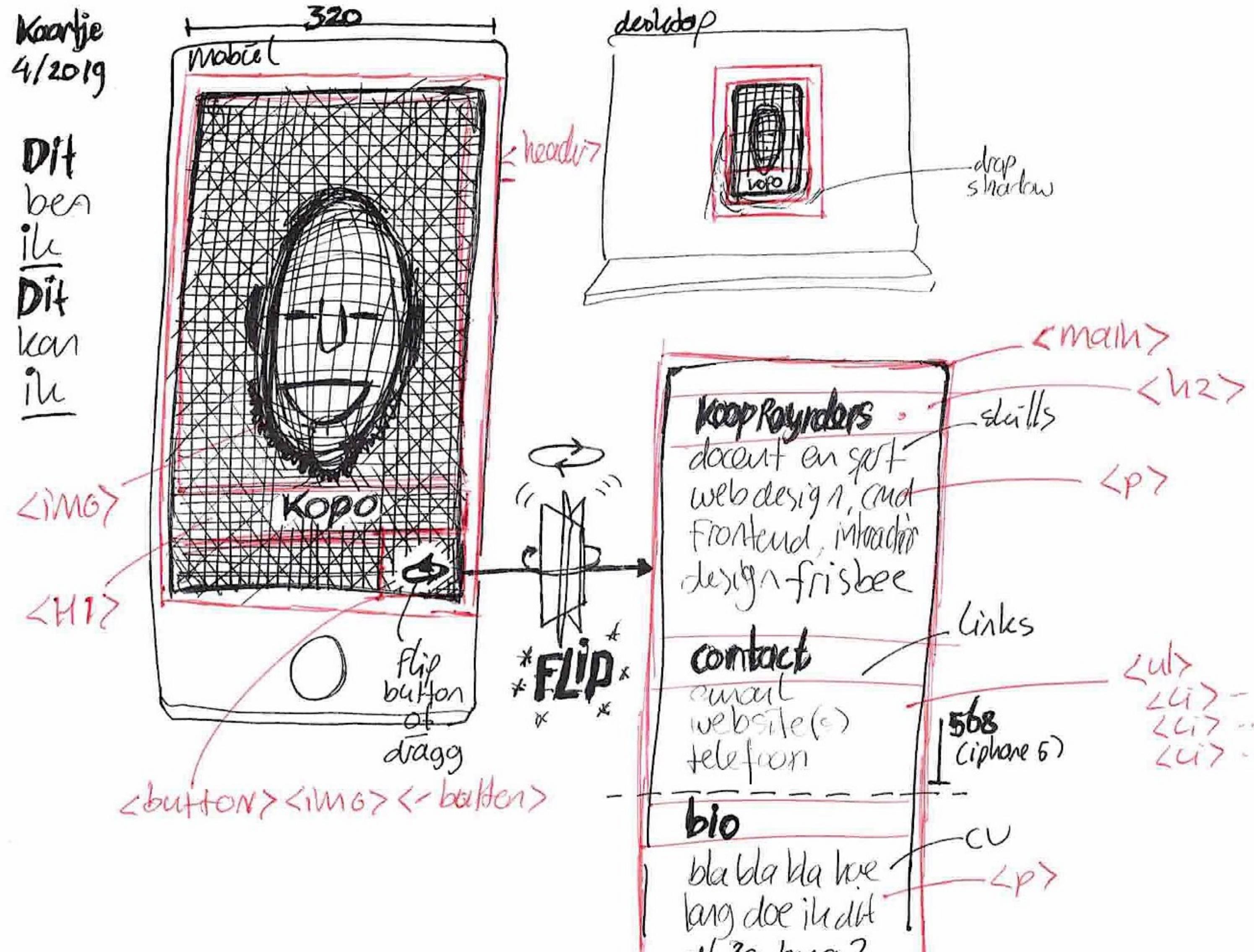
User interface



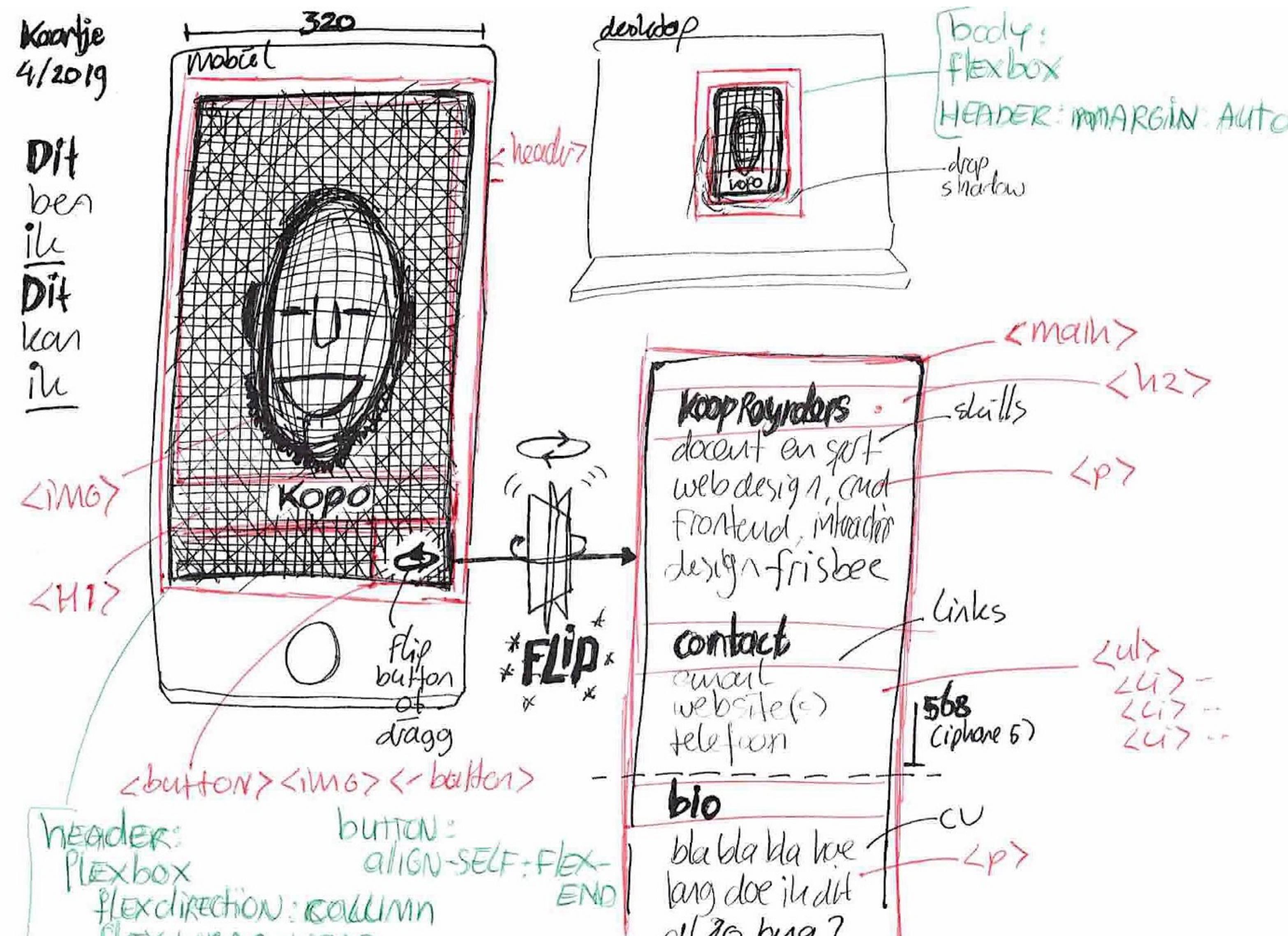
User interface



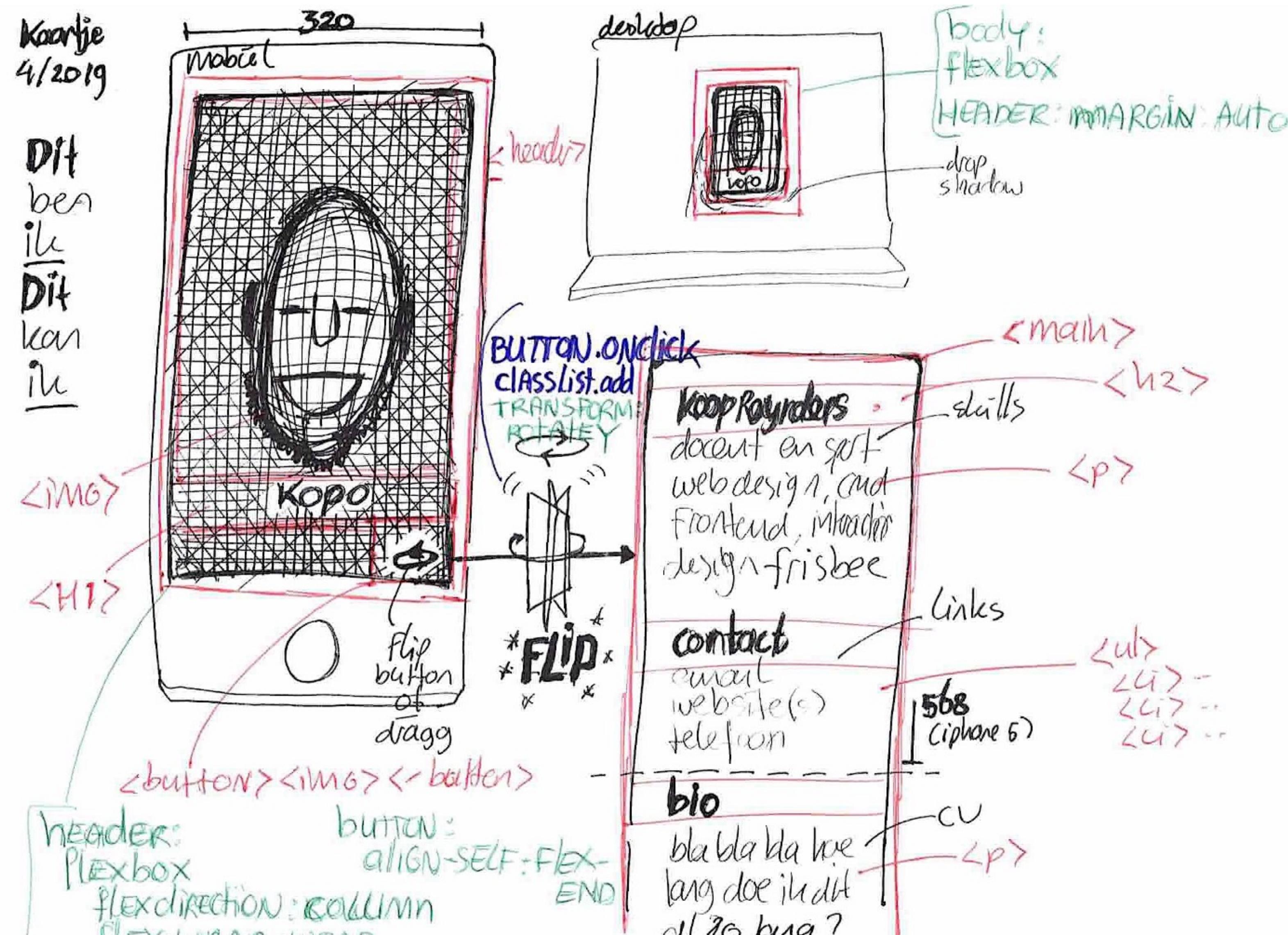
User interface



User interface



User interface



**A user interface is like a joke.
If you have to explain it,
it's not that good.**