```
tt()
```

Schedule

- Code standards
- Refactoring

Schedule

- **Code standards**
- Refactoring

Code standards?

C Prototypen en uitwerken	2	6	8	10
Gebruikt API's om tot een prototype te komen waarin data wordt getoond die altijd up-to-date is **Overziet [technische] implicaties van het ontwerp** *Bewijslast: de datahandling in het prototype, evt PB*	Werkt met **statische data**, uit een bestand die lokaal is opgeslagen en niet wordt ververst. (tenzij daar een heel goede reden voor is…)	Haalt één keer data remote op, en **verwerkt die lokaal.** Begrijpt de implicaties van deze keuzes. Haalt data asynchroon op en verwerkt die tot een interactieve visualisatie.	De data wordt – met reden - op **verschillende momenten ververst** of geïntegreerd met data uit een andere bron. User keuzes worden vertaald in opties voor filters en sortering.	Combineert aangereikte best practices met passende **technieken, methoden, en/of activiteiten** die door **eigen onderzoek** zijn verworven
Combineert principes, conventies en best-practices op het gebied van techniek en zet deze in om tot een onderhoudbaar, overdraagbaar prototype te komen *Bewijslast: motivatie tech keuzes tijdens mondeling, evt. PB en/of README*	Maakt **geen gebruik van conventies en best practices**. Komt daardoor niet tot een overdraagbaar prototype	Gebruikt de technologie (Svelte, D3) en de functionaliteiten die zij biedt. **Past conventies toe** om de **overdraagbaarheid** van het prototype te **verbeteren**	Structureert het prototype in overdraagbare **componenten**. En/of deelt bruikbare informatie over de structuur van het prototype in README.md	Gebruikt **geavanceerde methoden en technieken** die door **eigen onderzoek** zijn verworven om de **kwaliteit** van het prototype te **optimaliseren**

Code standards

- **Principes, conventies**
- Javascript (ES6), d3, Svelte
- ~~**var**~~
- ~~function() {…}~~: voorkeur voor named functions of arrow notatie
- ~~<p style=‘color: red’>~~ voorkeur voor css of d3

Best practices

- Naamgeving variabelen (wat bevat de variabele?)
- Naamgeving functies (wat doet de functie?)
- Overweeg het gebruik van hulpfuncties
  (data fetchen - UI renderen - Charts renderen)
- Comments, in-line omschrijving verantwoordelijkheid functies

Schedule

- Code standards
- **Refactoring**

Refactoring

In computer programming and software design, **code refactoring** is the process of restructuring existing computer code (…) without changing its external behavior. Refactoring is intended to improve the design, structure, and/or implementation of the software (…), while preserving its functionality. Potential advantages of refactoring may include improved code readability and reduced complexity; these can improve the source code's maintainability

Bron: https://en.wikipedia.org/wiki/Code_refactoring

Refactoring

why bother?

- code vereenvoudigen
- leesbaarheid verbeteren
- onderhoudbaarheid verbeteren

Bron: https://nl.wikipedia.org/wiki/Refactoren

Refactoring

- code vereenvoudigen

DRY, KISS,

Vermijd spaghetti code

lange functies vermijden / uit elkaar halen

Componetization (break code down into reusable semantic units)

Refactoring

- leesbaarheid verbeteren

Naamgeving variabelen (wat bevat de variabele?)

Naamgeving functies (wat doet de functie?)

Overweeg het gebruik van hulpfuncties

Refactoring

- onderhoudbaarheid verbeteren

Conventies (als zij er zijn)

Kan je je eigen code volgend jaar nog lezen?

Comments, in-line omschrijving verantwoordelijkheid functies

Refactoring

- Het proces documenteren

In de wiki

Wat heb je gedaan om de kwaliteit van de code te verbeteren?

Uncaught SyntaxError
Unexpected end of input