



INTERACTIEVE WEBSITES

JavaScript

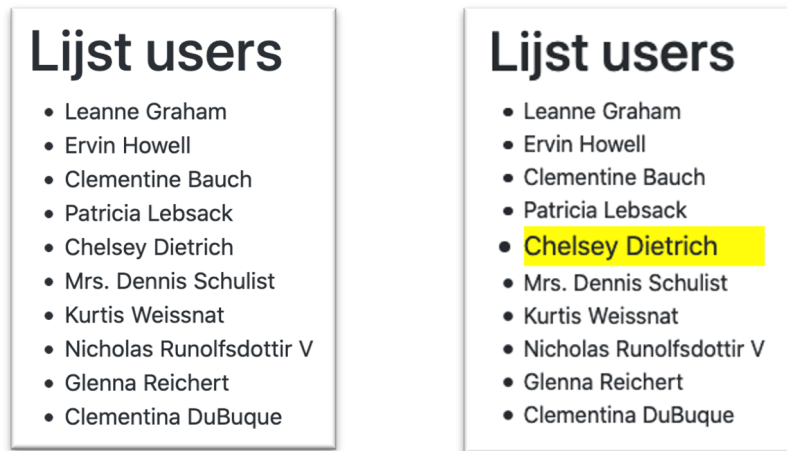
JS Fetch API en Promise

THOMAS
MORE

OEFENINGEN

Opdracht 1 Lijst users

Maak een webapplicatie waarmee je de fake **JSONPlaceholder** aanspreekt en een lijst van de **users** toont. Vul de **ul-lijst** die klaar staat (**lijst**) op met **list-items**.



uitbreiding

Bewaar deze oefening als **opdracht_fetch_fake_json.html**

Opdracht 1 Lijst users (uitbreiding)

Maak gebruik van Event Delegation om de (dynamisch gegenereerde) list-items te accentueren (highlight) wanneer we met onze muis er over hoveren. Verwijder steeds de class van het vorige element.

Bewaar deze oefening als **opdracht_fetch_fake_json_event_delegation.html**

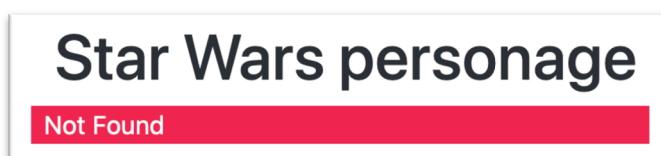
Opdracht 2 Star Wars personages

Maak een webapplicatie waarmee je de personages (<https://swapi.dev/api/people/>) van **Star Wars** toont in een "List group" van Bootstrap. Enkel de eerste pagina (10) tonen is voldoende.

Vul de **ul-lijst** die klaar staat (**starWarsPeopleList**) op met **list-items**.

Achter de **naam** toon je het **geslacht**. Indien het geen man/vrouw is dan **toon je niets**.

Zorg voor een foutafhandeling wanneer we de URL bijvoorbeeld wijzigen naar "https://swapi.dev/api/peooooople/". De foutmelding toon je in de div "divErr".



Gebruik de **https://swapi.dev** om de API aan te spreken. Werk met **fetch** en **Promise**.

Bewaar deze oefening als **opdracht_fetch_starwars.html**

Star Wars personage

Luke Skywalker (male)

C-3PO

R2-D2

Darth Vader (male)

Leia Organa (female)

Owen Lars (male)

Beru Whitesun lars (female)

R5-D4

Biggs Darklighter (male)

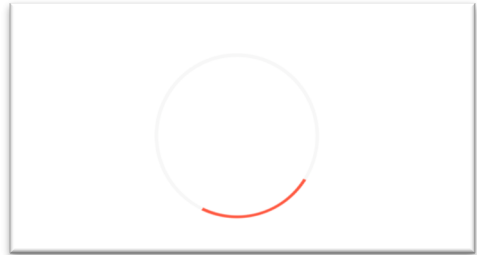
Obi-Wan Kenobi (male)

Opdracht 3 Lorem Picsum

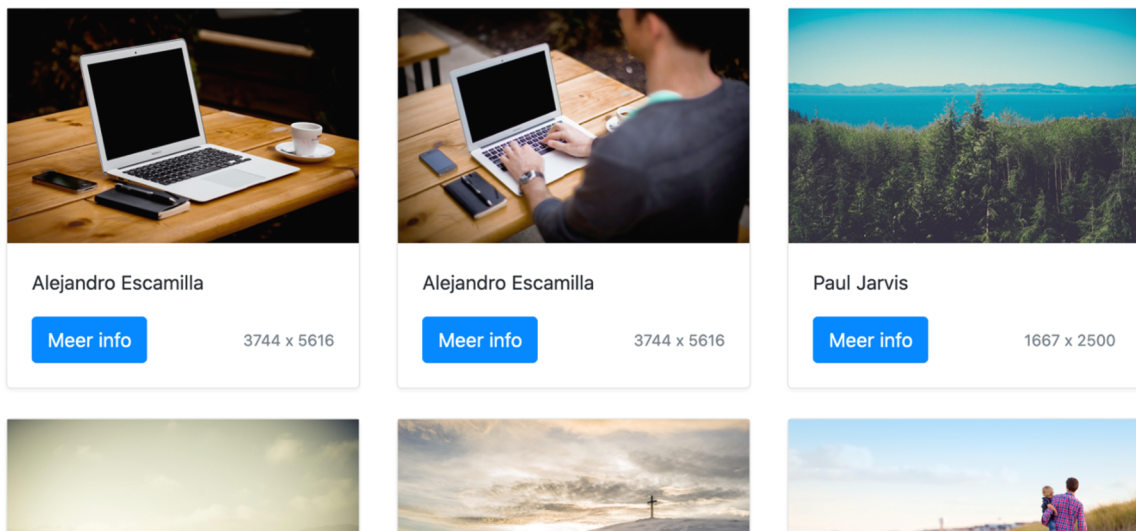
Maak een webapplicatie waarmee je (random) afbeeldingen gaat ophalen van unsplash.com. Haal een **lijst** van **25 afbeeldingen** binnen. Zoek de URL even op via <https://picsum.photos>. We tonen enkel de liggende afbeeldingen!

Maak gebruik van de spinner in het startbestand.

- Wanneer de DOM geladen is voer je volgende zaken uit:
 - **Toon** de **spinner**
 - Roep de **asynchrone** functie **makeRequest** op
 - Roep de functie **makeGall** op
 - **Verberg** de **spinner**
 - *Let op: de spinner verdwijnt wanneer de html-code van de makeGall in de DOM geladen is. De (grote) afbeeldingen moeten dan nog doorkomen!*
- Functie **makeRequest**
 - Deze functie geeft het resultaat van de fetch terug
 - Zet de response om naar json
 - Gebruik de **.filter** functie om enkel liggende afbeeldingen in de array over te houden
- Functie **makeGall**
 - Deze functie bouwt op basis van de resultaat-array de html-code op
 - Gebruik een **col-md-4** en de html-code voor een **card** met **afbeelding**
 - Probeer onderstaande view zo goed mogelijk na te bouwen
 - Gebruik de **author**, **url**, **height** & **width** en voor meer info de **download_url**



Lorem Picsum The Lorem Ipsum for photos.



Bouw optioneel een vertraging in via een **fakeTimeout** functie van bijvoorbeeld 3 seconden.

```
await fakeTimeout(3000); // Roep deze functie (als eerste) in de makeRequest functie op.  
  
fakeTimeout = (ms) => {  
  return new Promise(resolve => setTimeout(resolve, ms));  
}
```

Bewaar deze oefening als **opdracht_fetch_async_gallery.html**