

Onderzoek functie fetchData():

a. Wat doet de functie volgens jou?

Het is een manier om iets op te halen van bijvoorbeeld een server, maar in

het voorbeeld van meester Michiel is het als voorbeeld zonder server.

b. Wat betekent het dat deze functie een promise retourneert?

Dit betekent dat de promise af spreekt dat deze data terugkeert die getypt is tussen het haakjespaar.

c. Welke gegevens verwacht je na de uitvoering van deze functie?

Ik verwacht dat er na 2 seconden al de namen van de leerlingen van mijn

klas verschijnen.

Denk na over het asynchrone proces:

a. Wat zou er gebeuren als je een asynchrone functie gebruikt om data op te halen?

De data zou niet in een rijtje worden verzameld maar het programma zou verder

gaan met werk totdat alle data is verzameld. De "lopende band" valt niet stil.

b. Schrijf kort op wat volgens jou het verschil is tussen een asynchrone functie en een normale synchrone functie.

Bij synchroon wordt alles stap voor stap gedaan en als er bv. data moet

worden verzameld kan het programma niet verder totdat het proces is

voltooid, dit neemt dus meer (kostbare) tijd in beslag. Bij asynchroon doet het programma verder met andere processen terwijl de data wordt

verzameld, dit neemt minder tijd in beslag.

Onderzoek de bestaande HTML

a. Welke elementen zijn aanwezig en wat is hun functie in het oplossen van het vooropgestelde probleem.

Ik zie een knop die het asynchroon proces start. Er is ook een spinner

voor de duidelijke feedback van de gebruiker zodat hij weet dat hij moet

wachten.

b. Op welke manier zal je de nodige elementen kunnen selecteren?

Aan de hand van de id's en de classes.

Onderzoek de bestaande CSS code

a. Hoe kan feedback gegeven worden tijdens het laden van de gegevens?

Door een spinner te gebruiken zo weet de gebruiker dat het proces bezig

is en dat hij/zij even zal moeten wachten.