

“ *Everyday life is like programming, I guess. If you love something, you can put beauty into it.*

—Donald Knuth

Polling

- 1 De nood aan persistentie
- 2 localStorage
- 3 Nieuwe add.html pagina
- 4 Functie verwijderCitaat refacoren
- 5 Toevoegen van een nieuw citaat
- 6 DRY: updateLS
- 7 Uitbreiding: voeg willekeurig citaat toe
- 8 Alle citaten verwijderen + CSS update
- 9 Testen

1 De nood aan persistentie

De huidige toestand van onze JS-app heeft enkele beperkingen:

- Alles speelt zich af in één pagina. Als we een nieuwe pagina toevoegen, dan heeft die geen verbinding met de bestaande pagina.
- Als je de browser sluit, ben je alle aanpassingen kwijt. De volgende keer dat we de applicatie openen, zal die weer in de standaardversie getoond worden.

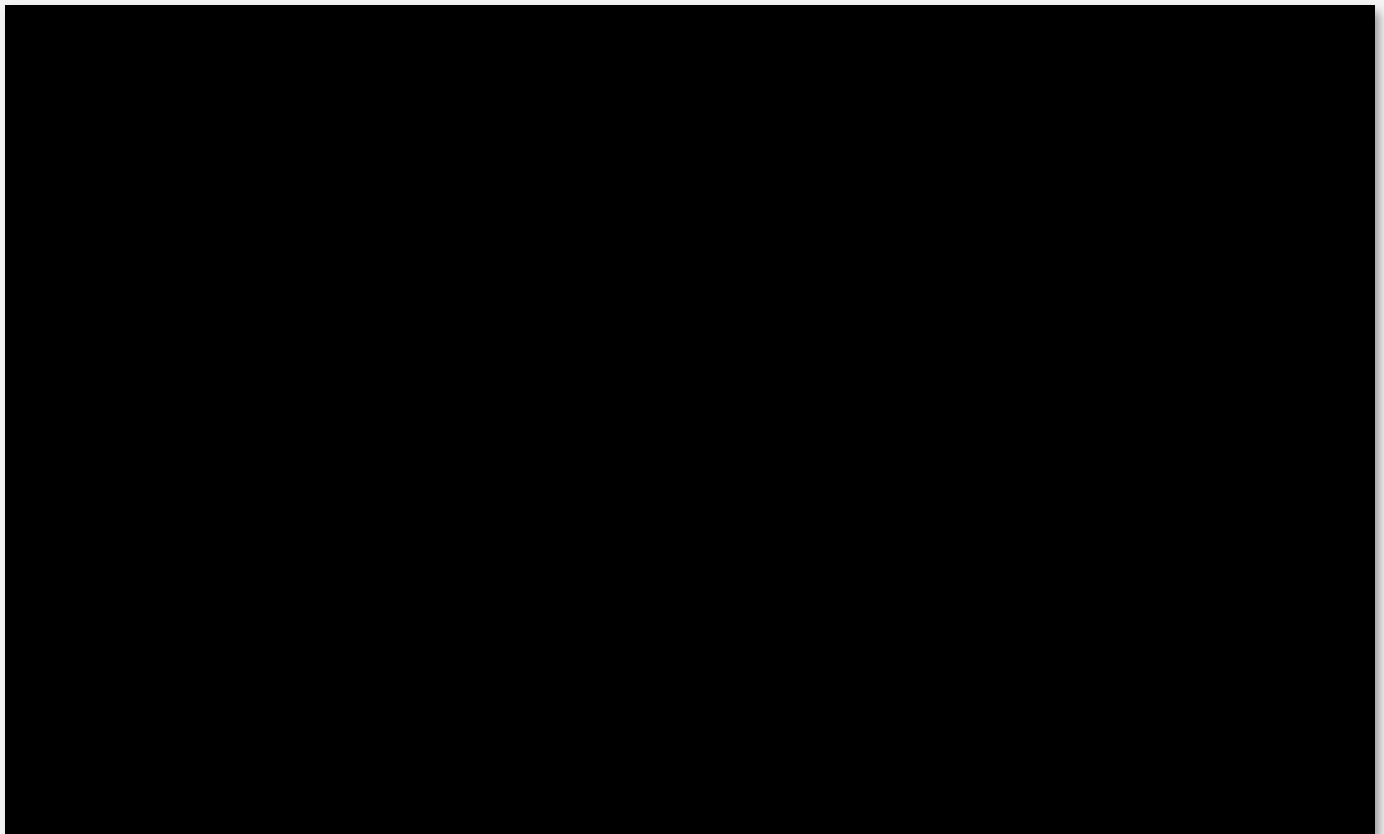
2 localStorage

Om deze beperkingen op te lossen, kunnen we gebruik maken van de **localStorage** van de browser. Dit is een soort van databank die in de browser zelf zit. Hierin kunnen we gegevens opslaan die we later weer kunnen opvragen. We spreken in dit verband over ‘persistente opslag’ (Latijn: ‘persistere’ = ‘blijven bestaan’).



Om een nieuwe technologie uit te testen is het meestal een goed idee om een klein voorbeeld te maken en hiermee wat dingen uit te proberen. Dat is wat we in het eerste filmpje doen:

- we maken twee HTML pagina's, één om dingen te bewaren in localStorage en een andere om alles op te vragen uit de localStorage;
- de methodes `getItem` en `setItem`;
- een item in localStorage heeft is een key / value paar;
- de value is altijd een string, daarom nood aan `JSON`;
- localStorage in de developer tools van de browser.



In de volgende filmpjes voeren we localStorage in onze applicatie in. Daarvoor moet de citatenapp grondig *gerefactored* worden! Er komt o.a. een tweede HTML-pagina en heel wat functies moeten herschreven worden.

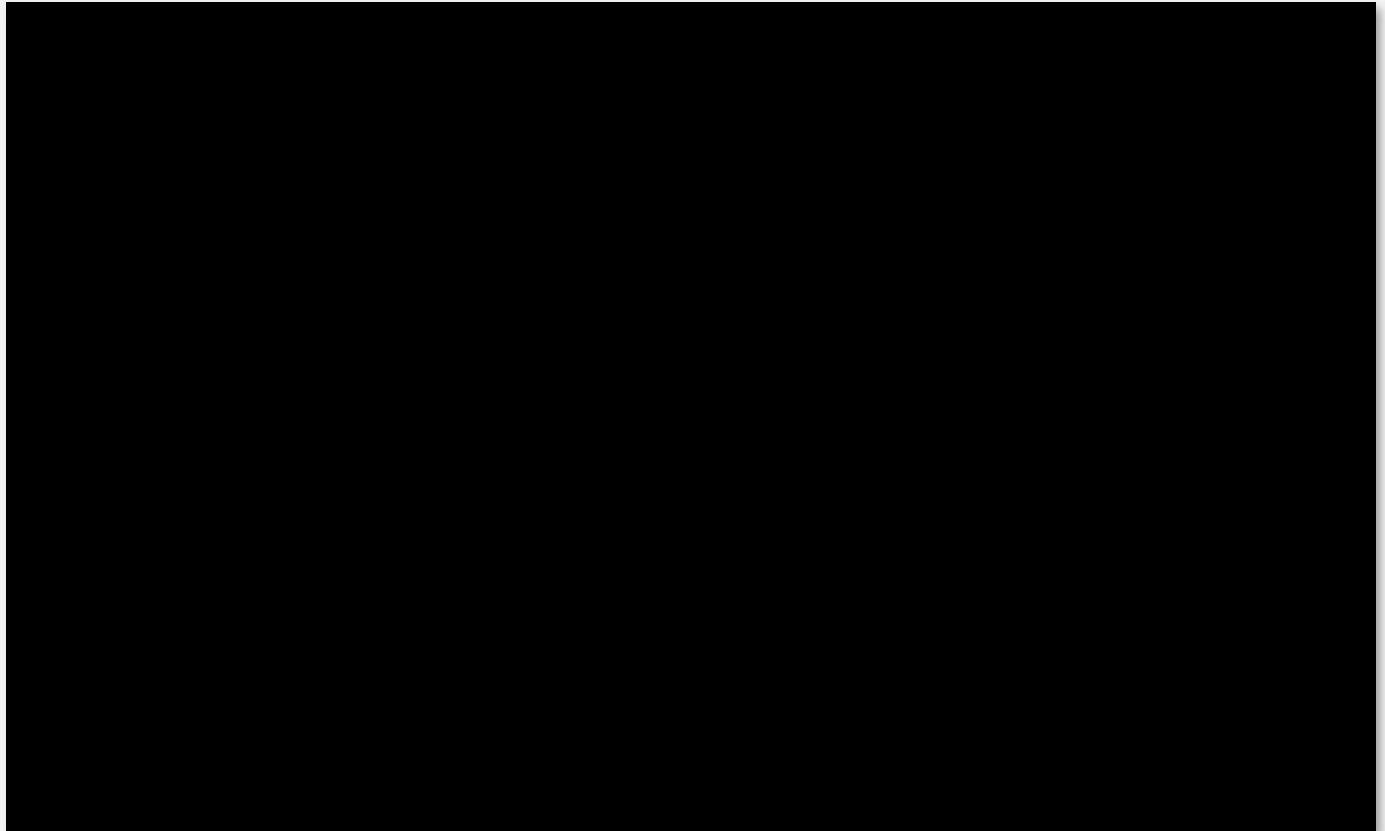
3 Nieuwe add.html pagina

In een eerste stap verhuist het toevoegformulier naar een aparte HTML-pagina 'add.html'. Aangezien het toevoegen daar zal gebeuren, maken we ook een JS-script 'add.js'.



Het filmpje toont volgende topics:

- formulier in add.html;
- add.js importeert alle basiscitaten en slaat die op in localStorage;
- script.js leest het item met alle basiscitaten uit localStorage;
- inspectie van de localStorage in de browser developer tools.



4 Functie verwijderCitaat refacoren

In een volgende stap wordt de functie `verwijderCitaat` herschreven. Deze functie moet nu ook de localStorage aanpassen.



In het filmpje overlopen we nog eens de flow van de applicatie. Wat gebeurt er juist als een citaat verwijderd wordt? Hoe moet deze flow worden

aangepast aan het gebruik van localStorage? Gelukkig is die aanpassing niet zoveel werk: we moeten er enkel voor zorgen dat het item in localStorage wordt vervangen door een nieuwe versie. Zoals altijd: test elke update die je doet grondig!

verwijderCitaat

Aanpassen aan localStorage



Frontend basis
Jan Van Hee

Graduaat programmeren

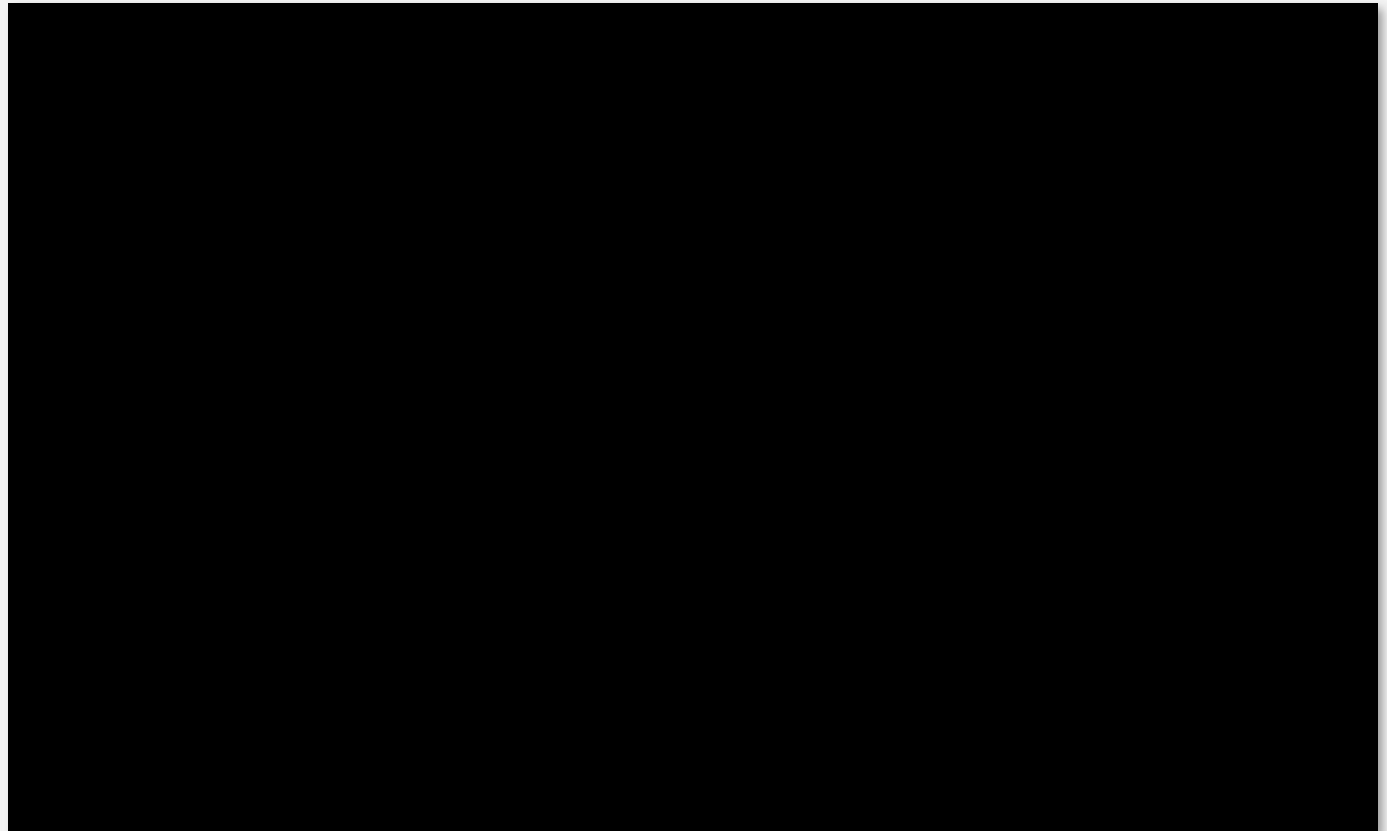
5 Toevoegen van een nieuw citaat

In de laatste stap voegen we een nieuw citaat toe. Dit is een beetje complexer dan het verwijderen van een citaat. We moeten immers een nieuw citaat toevoegen aan de bestaande lijst van citaten. Dit betekent dat we de bestaande lijst moeten ophalen, een nieuw citaat toevoegen en de aangepaste lijst terug moeten opslaan in localStorage.



Ook in het volgend filmpje gaat het eerder over verplaatsen van code en enkele aanpassingen aan de opslag in localStorage. We hebben immers de toevoeglogica al geschreven in script.js. Bij een eerste test blijkt al snel dat het fout loopt omdat er nog geen array `alleCitaten` is. In dit stukje zullen we trouwens meer dan de helft van de tijd steken in het grondig testen en veel

minder in het schrijven van nieuwe functionaliteit (want we kunnen gewoonlijk bestaande code verplaatsen).



6 DRY: updateLS

In de laatste stap van de refactoring van de citatenapp maken we een functie `updateLS` die de localStorage bijwerkt. Deze functie zal gebruikt worden in de functies `verwijderCitaat` en `voegCitaatToe`.



In het filmpje kunnen we nog eens de kracht van het modulesysteem van JS illustreren. We maken een nieuwe functie die een aantal regels JS combineert en definiëren die functie in een nieuw 'hulpfuncties.js' bestand. De nieuwe functie importeren we dan in 'script.js' en 'add.js'.

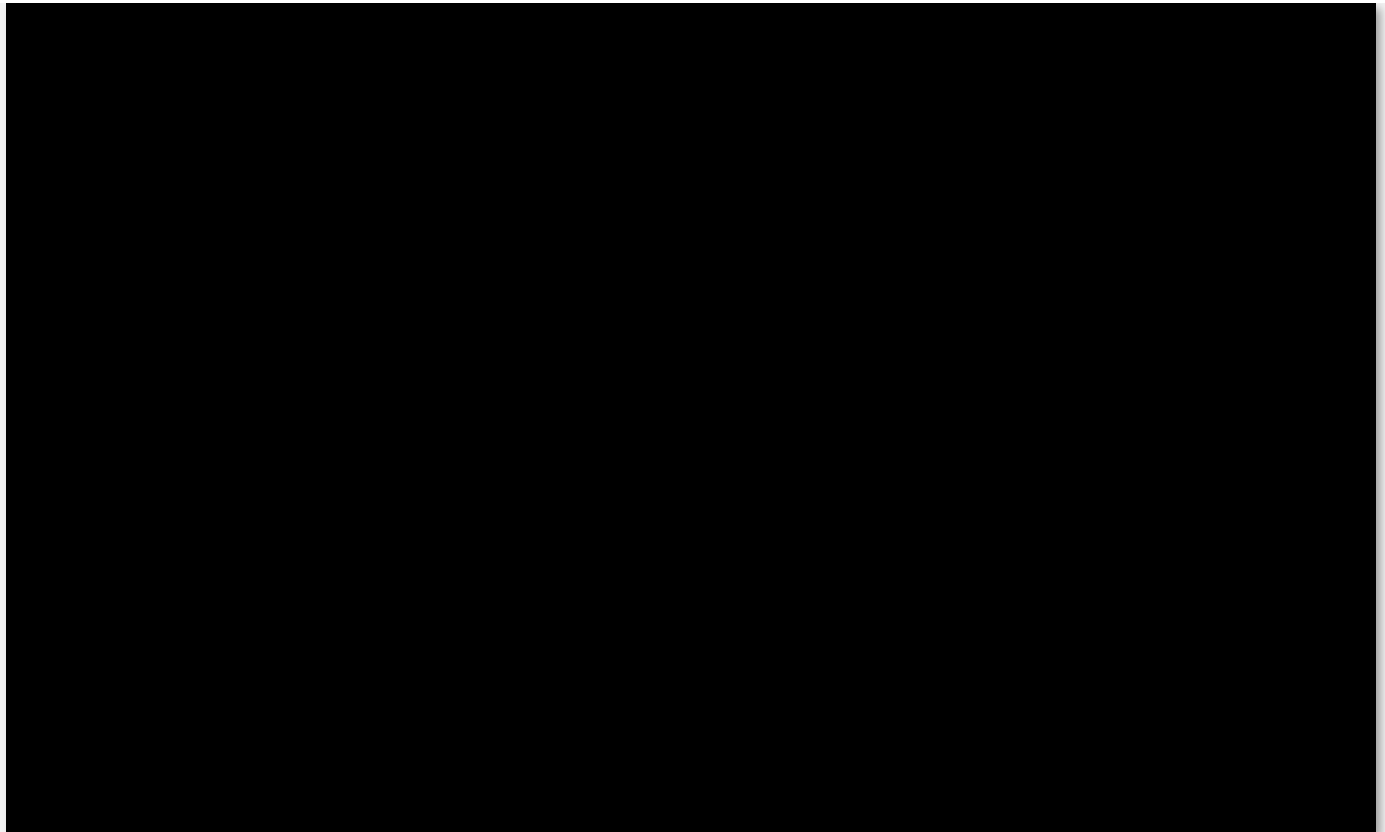
7 Uitbreiding: voeg willekeurig citaat toe

We maken een knopje om willekeurig één van de 21 voorgedefinieerde citaten toe te voegen. Dat ken je al: button in HTML, eventListener in JS enz. De uitdaging is een nieuwe functie `voegBasisCitaatToe` schrijven.



Onderstaand filmpje behandelt het volgende:

- gelijkheid van objecten;
- eerst de flow van de functie in commentaar uitschrijven;
- nieuwe array `alleTitels`;
- lussen met `forEach`;
- arraymethodes `push()`, `includes()`;
- wiskundige methodes: `Math.floor()`, `Math.random()`.



8 Alle citaten verwijderen + CSS update



Voorlaatste filmpje waarin we eindelijk ook eens even een klein beetje in detail kunnen gaan qua User Interface (UI). We tonen:

- Knop en functie om alle citaten te wissen;
- CSS: witruimte toevoegen (margin);
- boek 'Refactoring UI': button realistischer maken;
- :active pseudoklasse.

9 Testen

Als je grondig test (en waarom zou je niet grondig testen?) vind je vaak nog speciale gevallen waarin je applicatie niet helemaal correct werkt.



We eindigen dus met extra testen die nog enkele problemen blootleggen. Bij een bepaalde opeenvolging van stappen is er nog een probleem. De oplossing vergt wat denkwerk over de juiste flow van de applicatie en slechts enkele regels code.



[↑ Naar top](#)

© 2024 — Jan Van Hee