DATA

- JSON
- Object aanmaken in JSON notatie
- Een object omzetten naar een JSON string : stringify()
- Een JSON string omzetten naar een object : parse()
- localStorage
- destructuring
- Spread... operator
- ...Rest parameter

JSON formaat

Om onze gegevens in javascript bij te houden of door te sturen, kunnen we gebruik maken van het JSON formaat (Javascript Object Notation).

Met dit lichtgewicht formaat is het heel makkelijk om je state bij te houden, door te sturen of als mens te lezen.

JSON notaties beginnen met { en eindigen met }

Binnen een JSON bestand kom je naam/waarde paren en geordende lijsten tegen (arrays)

Object in Javascript

Een Object in javascript aanmaken via de JSON notatie ipv new Object :

```
const persoon = {
        naam : 'Steven',
        kinderen : [ 'Tristan', 'Aurian' ]
};
```

[codepen]

Een object omzetten naar een JSON string : stringify()

Om een javascript object te bewaren of te verzenden naar een andere locatie, moeten we het object eerst omzetten naar een string met JSON.stringify()

[codepen]

Een JSON string omzetten naar een object : parse()

Als we een JSON string ontvangen moeten we die met JSON.parse() omzetten naar een echt object om er met te kunnen werken.

[codepen]

localStorage

Je kan lokaal gegevens in de vorm van naam/waarde paren bewaren in localStorage.

Deze gegevens zijn enkel beschikbaar in de browser en op het domein waarop je het script uitvoert.

Deze gegevens kunnen (automatisch) gewist worden als het toestel plaatsgebrek heeft. Doorgaans kan je rekenen op 5MB voor het toestel (alle domeinen samen)

Let op : je kan enkel strings wegschrijven.

[codepen]

localStorage

Een weinig-gedocumenteerde syntax van localStorage werkt zonder getItem / setItem en gebruikt de standaard javascript dot notatie (*obj.key*).

Je zal deze syntax veel minder terugvinden in documentatie, maar het werkt, zolang je niet werkt met de reserved keys *length*, *key*, *getItem*, *setItem*, *removeItem* en *clear* 🙇

[codepen | Stack Overflow]

destructuring

- != destroying
- javascript syntax waarmee je waarden uit arrays of eigenschappen uit objecten in aparte variabelen kan stoppen
- werkt met een *array* of *object* aan de linkerzijde van een toewijzing (=)

[codepen | MDN]

Spread… operator

- geschreven met drie opeenvolgende puntjes …
- om itereerbare objecten (Arrays, Sets, …) in meerdere elementen uit te spreiden
- …naamVariabele

[codelab]

…Rest parameter

- Als je op voorhand het exacte aantal parameters niet kan voorspellen, kan je werken met de …rest operator.
- De …rest parameter combineert meerdere parameters tot 1 itereerbaar object (Array).

```
functie naamFunctie(param1, param2, …alleAndereParametersSamen){

        // doe iets

}
```

[codelab]