

# Web Development

JavaScript – Labo 9



# Lokale vs. globale variabelen

- Lokaal
  - Enkel beschikbaar binnen één functie / code block
- Globaal
  - Overall binnen het script beschikbaar
- **MO**: een variabele nodig in twee functies betekent niet automatisch dat het een globale moet zijn!

# Globale variabelen

- Liever zo veel mogelijk vermijden
  - *Naming conflict* wanneer twee stukken code toevallig dezelfde naam gebruiken
  - *Moeilijker debuggen* aangezien verschillende delen een variabele kunnen wijzigen
- Risico's beperken met één “global” object

```
let global = {  
  score: 0,  
  players: []  
}
```

# Globale variabelen: demo

- Demo 1 globale variabelen
- Blijft de data behouden als je de pagina herlaadt?
- Blijft de data behouden als je het tabblad sluit en de pagina opnieuw opent?
- Gebruiken twee tabbladen die dezelfde pagina tonen, dezelfde data?

# Gegevens over pagina's heen

- Bijhouden op server (zie volgend jaar)
- Bijhouden op client
  - Cookies
  - Web Storage API: session storage
  - Web Storage API: local storage
  - Web SQL Database

# Cookies

- Een stukje tekst bewaard op vraag van de server of JavaScript
- Inhoud: key/value pair
- Wordt mee verstuurd met elke request naar de server
- Soorten
  - *Permanente cookies*
    - Blijft bestaan na afsluiten browser
    - Heeft wel een houdbaarheidsdatum
  - *Session cookies*
    - Verloren na afsluiten van browser (niet tabblad!)



# Cookies: toepassing

- Meest voorkomende
  - Bijhouden van een session ID
  - Een unieke identifier om jouw sessie op de server terug te vinden
- Andere vaak voorkomende
  - “Gebruikersnaam onthouden” zoals op Toledo
  - Zolang de cookie niet vervallen is, blijf je aangemeld
- Aanmaken via *document.cookie*

# Web Storage API

- Tekst bewaren in de browser, per domein
- Net zoals cookies: key/value pairs
- Data is enkel toegankelijk via JavaScript, dus heel vaak JSON gebruikt
  - Hoe zetten we tekst ook alweer om naar JSON?
  - En omgekeerd?



# Web Storage API: soorten

## Session storage

- Gekoppeld aan tabblad
  - Behouden zolang tab open blijft
  - Gewist bij het sluiten van de tab

```
sessionStorage.setItem('key', value);  
sessionStorage.getItem('key');
```

```
sessionStorage.removeItem('key');  
sessionStorage.clear();
```

[Demo 2 session storage](#)

## Local storage

- Gekoppeld aan je toestel
  - Blijft behouden, ook na sluiten van de tab / browser

```
localStorage.setItem('key', value);  
localStorage.getItem('key');
```

```
localStorage.removeItem('key');  
localStorage.clear();
```

[Demo 3 local storage](#)

# Web SQL Database

- Opslaan zoals in een andere database
- Ophalen via SELECT statements
  - Voordelen:
    - Betere opslag bij grotere volumes
    - Aan de hand van WHERE makkelijk te filteren (vs. zelf filteren van je JSON)
  - Nadelen:
    - Nog niet in iedere browser ondersteund
- (Wordt verder niet behandeld in dit vak)