

# JavaScri

## Table of Contents

Objectifs .....	2
Les cookies c'est quoi ? .....	2
Usage des cookies .....	2
Démonstration .....	2
Les cookies : Création via un serveur .....	2
Les cookies : Création via un serveur .....	3
Les cookies : Création via un programme JavaScript .....	3
Les cookies : Créer un cookie en JavaScript .....	3
Démonstration .....	3
Les cookies : Lire un cookie en JavaScript .....	3
Démonstration .....	4
Local Storage c'est quoi ? .....	4
Local Storage : Ecrire des données .....	4
Démonstration .....	4
LocalStorage : Lire les données .....	4
Démonstration .....	4
Le localStorage ne gère que des String .....	4
Démonstration .....	5
JSON c'est quoi ? .....	5
SessionStorage c'est quoi ? .....	5
Session Storage : La syntaxe .....	5
Démonstration .....	6
A toi de jouer .....	6
Résultat de la réflexion .....	6
Démonstration .....	6
Conclusion .....	6
Objectif / Niveau .....	6
A toi de jouer .....	7

## Module 06 - Stockage des données

### Discours formateur

Bienvenue dans ce module traitant du stockage des données sur un navigateur.

# Objectifs

- Connaître les différents façon de stocker des données côté client
- Savoir manipuler les cookies avec Javascript
- Savoir manipuler le local storage
- Savoir manipuler le session storage
- Savoir bien choisir entre cookies/local ou session storage
- Savoir utiliser JSON

## Les cookies c'est quoi ?

Un cookie est un petit fichier stocké sur le navigateur de l'utilisateur et associé à un domaine web. Ce fichier est automatiquement renvoyé lors de contacts ultérieurs avec le même domaine.

## Usage des cookies

- Couple clé/valeur
- 4 ko maximum
- Stocké sur le navigateur
- Durée de vie aléatoire

Les cookies ont de multiples usages : ils peuvent servir à mémoriser votre identifiant client auprès d'un site marchand, le contenu courant de votre panier d'achat, la langue d'affichage de la page web, un identifiant permettant de tracer votre navigation à des fins statistiques ou publicitaires, etc. Certains de ces usages sont strictement nécessaires aux fonctionnalités expressément demandées par l'utilisateur ou bien à l'établissement de la communication et donc exemptés de consentement. D'autres, qui ne correspondent pas à ces critères, nécessitent un consentement de l'utilisateur avant lecture ou écriture.

## Démonstration

Allez voir ses cookies sur son navigateur

## Les cookies : Création via un serveur

[cookies 1.drawio] | *images/cookies\_1.drawio.png*

# Les cookies : Création via un serveur

[cookies 2.drawio] | [images/cookies\\_2.drawio.png](#)

# Les cookies : Création via un programme JavaScript

[cookies 3.drawio] | [images/cookies\\_3.drawio.png](#)

## Les cookies : Créer un cookie en JavaScript

```
// document.cookie = 'cle=valeur';  
// ajoute un cookie avec la clé lang et la valeur français  
document.cookie = 'lang=français';
```

## Démonstration

Création d'un cookie en JavaScript

## Les cookies : Lire un cookie en JavaScript

```
console.log(document.cookie); // donne toute la grande String contenant tous les  
cookie  
  
// pour lire un cookie il faut "un peu" de logique  
const cookiesAsArray = document.cookie.split('; ');  
// on parcourt le tableau  
for (const cookie of cookiesAsArray) {  
    // si la chaîne commence par le début du cookie qui nous intéresse  
    if (cookie.startsWith('nomDuCookieRecherché')) {  
        // on découpe le contenu du cookie  
        const cookieInfo = cookie.split('=');  
        // pour récupérer que la valeur  
        const value = cookieInfo[1];  
        // la valeur est trouvée  
        console.log(value);  
    }  
}
```

# Démonstration

Lecture d'un cookie en JavaScript

## Local Storage c'est quoi ?

- Lié au domaine qui fait tourner notre application, on ne peut pas récupérer des infos qui auraient été mises de côté sur Facebook alors qu'on est sur Youtube
- Plus pratique à utiliser que les cookies
- ☐ Les informations du localStorage ne sont pas incluses dans les requêtes HTTP
- Pas de limite de conservation des données

## Local Storage : Ecrire des données

```
// l'objet localStorage est global, il est toujours disponible :  
// - via la méthode setItem on sauvegarde une information, même si je quitte la page  
//   et que je reviens, l'information est persistante  
// - on passe 2 arguments, la clé pour retrouver la valeur plus tard, et la valeur  
localStorage.setItem('age', 31);
```

# Démonstration

LocalStorage : Ecrire des données avec JavaScript

## LocalStorage : Lire les données

```
// via la méthode getItem, on récupère la valeur, il faut passer la clé en argument  
const userAge = localStorage.getItem('age');  
console.log(userAge); // '31'
```

# Démonstration

LocalStorage : Lire des données avec JavaScript

## Le localStorage ne gère que des String

Si on veut stocker n'importe quel type de données (un objet, un tableau, un booléen, ...), on peut le transformer en String via :

```
const valeurATransformer = true;
JSON.stringify(valeurATransformer);
```

## Démonstration

Transformer des données en String

## JSON c'est quoi ?

Le JSON (JavaScript Object Notation) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Comme c'est du texte, on peut donc le stocker dans le localStorage ☐

## SessionStorage c'est quoi ?

SessionStorage fonctionne de la même manière que LocalStorage mise à part que les données du sessionStorage ont une durée de vie limitée à la session de navigation. Si on ferme le navigateur, les données sont effacées.

## Session Storage : La syntaxe

- On pourra donc :
  - ajouter/modifier une information

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

- récupérer une information

```
sessionStorage.getItem('key');
```

- supprimer une information

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

- supprimer toutes les informations

```
sessionStorage.clear();
```

# Démonstration

Utilisation de SessionStorage

## A toi de jouer

Sur Trip’Odvisor, quel méthodologie utiliserais tu pour sauvegarder l’activation du dark mode ?

## Résultat de la réflexion

- Résultat de la réflexion :
  - cookie, non, car pas besoin d’envoyer ce choix dans chaque requête HTTP (pas utile côté serveur)
  - sessionStorage, non, car on veut conserver la préférence de l’utilisateur même s’il ferme son navigateur
  - on part donc sur le localStorage ☐

## Démonstration

Sur Trip’Odvisor

Code JSON, localStorage, objets, méthodes, DOM lecture/écriture, évènements, classList, conditions

On ajoute la méthode `saveToLocalStorage` qui sauvegarde l’état du thème dans le `localStorage` On appelle cette méthode après avoir changé le thème dans `toggleDark` On crée la méthode `initLocalState` qui vérifie le thème sauvegardé et on l’appelle au chargement de la page dans `init` ☐☐  
Montrer le contenu du `localStorage` via les Devtools/Application/Storage/LocalStorage.

☐☐ Comme on utilise le domaine `localhost`, il y aura d’autres valeurs présentes dans le `localStorage` et pas seulement celles de notre site. Préciser qu’en production, on aurait notre propre nom de domaine et qu’on ne rencontrerait pas ce problème.

## Conclusion

- Vous connaissez les différentes méthodologies de sauvegarde de données sur un navigateur
- Vous savez créer et lire des cookies en JavaScript
- Vous savez créer et lire dans le `LocalStorage` de votre navigateur en JavaScript
- Vous savez créer et lire dans le `SessionStorage` de votre navigateur en JavaScript

## Objectif / Niveau

- **Essentiel** : Comprendre le fonctionnement de tous les mécanismes de persistance côté client.

- **Attendu** : Savoir utiliser tous les mécanismes de persistance côté client.
- **Avancé** : Savoir choisir le meilleur mécanisme de persistance selon une situation et savoir utiliser tous les mécanismes de persistance côté client.

## A toi de jouer

Challenge