

Programmation Web – Avancé

BINV2150 A : JavaScript (& JAVA SERVLETS)

Week 4

R. Baroni / J.L. Collinet / C. Damas



*Presentation template
by [SlidesCarnival](https://www.slidescarnival.com/)*

0

Table des matières

Tous les sujets traités pendant ce cours...



Table des matières

1. Engagement pédagogique
2. Introduction au contexte d'utilisation de JS
3. Introduction au langage JS côté client, à l'utilisation d'APIs du navigateur et de librairies JS
4. Introduction aux communications (synchrone) client /serveur



Table des matières

- 5. Introduction aux single-page web applications et aux communications asynchrones client / serveur
- 6. Introduction à l'authentification sécurisée d'un utilisateur et aux cookies
- 7. Projet mettant en œuvre une SAP et des librairies JS

3

Introduction au langage JS côté client, à l'utilisation d'APIs du navigateur et de librairies JS

Découvrons le langage côté client...

Table des matières :



3. Introduction au langage JS côté client, à l'utilisation d'APIs du navigateur et de librairies JS

1. Introduction au JS côté-client
2. Interaction de base avec ou sans un browser :
quels programmes utiliser ? où mettre le code ?
3. Instruction JS
4. Les commentaires
5. Déclaration, initialisation et mise à jour de variables

Table des matières :



3. Introduction au langage JS côté client, à l'utilisation d'APIs du navigateur et de librairies JS

- 6. Les opérateurs
- 7. Les conditions
- 8. Les fonctions personnalisées et anonymes
- 9. Interactions de base avec l'API DOM
- 10. Introduction à JQuery en interaction avec le DOM
- 11. Introduction à la gestion d'événements

Table des matières :



3. Introduction au langage JS côté client, à l'utilisation d'APIs du navigateur et de librairies JS

12. HTML5 : Contraintes de Validation

13. Les boucles

14. Interaction avec l'API Canvas pour créer une animation

15. Introduction à une librairie JS pour créer une animation

16. Les tableaux

Table des matières :



3. Introduction au langage JS côté client, à l'utilisation d'APIs du navigateur et de librairies JS

17. Les exceptions

18. Les objets en JS

19. Introduction aux modules (ES6)

20. Introduction aux modules (Node.js)

21. Introduction aux modules (Node.JS) mis à disposition du browser (ES6)



Introduction à une librairie JS pour créer une animation

- Existence de nombreuses librairies JS pour animer différentes propriétés : CSS, attributs DOM, objets JS & SVG.
- Exemples de librairies open source :
 - Pour la 2D : <https://animejs.com/>
 - Pour la 3D : <https://threejs.org/>
 - Pour les jeux : <https://phaser.io/>



Introduction à une librairie JS pour créer une animation

- Librairie commerciale mais intéressante :
<https://greensock.com/> (GSAP)
- Bootstrap : <https://getbootstrap.com>



Introduction à une librairie JS pour créer une animation

- **DEMO-06** : faire tourner un mot comme une hélice, en déclenchant l'animation au passage sur le mot. En cliquant sur la flèche haut ou bas, faire monter ou descendre l'hélice. En cliquant sur le mot, celui-ci arrête de tourner et se met à grandir puis rapetissir.



Les tableaux

- Type **Array** (objet natif) : créer un ensemble ordonné de valeurs auxquelles ont fait référence avec un nom et un indice.
- Création d'un tableau rempli:
 - Recommandé : **var arr = [élément0, élément1, ..., élémentN];**
 - Autre façon : **var arr = Array(élément0, élément1, ..., élémentN);**
 - **var arr = new Array(101) ;** Qu'est-ce que ça fait ?



Les tableaux

- Création d'un tableau vide :
 - Recommandé : `var arr = [];`
 - Autre façon : `var arr = new Array();`
- Parcourir un tableau
 - Boucle classique : `for` et `.length`
 - Méthode `forEach()` d'un `Array`



Les tableaux

- Création d'un tableau vide :
 - Recommandé : `var arr = [];`
 - Autre façon : `var arr = new Array();`
- Parcourir un tableau
 - Boucle classique : `for` et `.length`
 - Méthode `forEach()` d'un `Array`
 - **DEMO-07**: parcourir un tableau et afficher ses éléments sous forme d'une liste à puces, tout en continuant de découvrir bootstrap.



Les tableaux

- Tableau multi-dimensionnel : un élément d'un **Array** est un **Array**

```
var myTab = new Array(line_count);
for (let x = 0; x < line_count; x++) {
    myTab[x] = new Array(column_count);
    for (let y = 0; y < column_count; y++) {
        // do something with myTab[x][y]
    }
}
```




Les tableaux

- Autres méthodes associées à un **Array** :
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array



Les exceptions : lancer une exception

- **throw expression**
- « JS built-in error object » a deux propriétés intéressantes : **name & message**
- Construction de vos propres erreurs :

```
throw { name: "Division_Exception", message: "Division par 0 !" };
```



Les exceptions : lancer une exception

- Utilisation de l'objet Error ou d'un autre (**RangeError**, **SyntaxError**...) :

```
throw new Error("Division par 0 ! " );
```

- Construction de vos propres classes d'erreur (héritage de la classe **Error**)



Les exceptions : lancer une exception

- Détails sur les types d'erreurs :
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Error#Error_types
- Exemples de gestion d'erreurs :
https://www.w3schools.com/js/js_errors.asp



Les exceptions : intercepter une exception

● try ... catch

- **try{...}**: partie de code monitorée
- **catch(err){...}** : instructions en réponses à une exception
- Code à exécuter après **try... catch** : **finally{}**

● Les exceptions de manière générale :

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control_flow_and_error_handling



Les exceptions

- **DEMO-08** : Introduire a la gestion d'exceptions au sein de la validation d'un formulaire de login (reprise partielle de l'exercice EX-01A). Comment pourrions-nous introduire la gestion d'erreur dans ce code ?



Exercices

A vous de jouer...



Introduction à une librairie JS pour créer une animation

- **EX-03** : Réalisez une animation esthétique avec plusieurs mots qui bougent. Au passage sur un mot, il doit s'agrandir. Au clic sur un mot, faites disparaître tous les autres mots et ajoutez une autre action de votre choix.



Interaction avec l'API Canvas pour créer une animation

EX-04A :

- a) Créez une fonction qui génère un **Array** à deux dimensions avec :
 - comme valeur pour chaque élément : chaîne de base + [numéro de ligne] + [numéro de colonne],
 - sur base de 3 arguments : le nombre de lignes, de colonnes, et la chaîne de base à afficher dans chaque élément du tableau.
- b) Créez un formulaire permettant d'introduire un nombre de lignes, un nombre de colonne, et une chaîne de base.

Number of lines

Number of columns

Initial string

Create table



Les tableaux et interaction avec le DOM

EX-04A :

- c) Créez une fonction qui génère une table HTML basé sur l'Array créé par la première fonction. N'utilisez pas de la création d'HTML à partir de la propriété innerHTML d'une <div>, mais créez des éléments DOM que vous ajoutez dynamiquement à une <div> existante.
- d) Optionnellement : Rendre l'HTML esthétique en utilisant bootstrap.

Number of lines

Number of columns

Initial string

Create table

CELL[0][0]	CELL[0][1]	CELL[0][2]
CELL[1][0]	CELL[1][1]	CELL[1][2]
CELL[2][0]	CELL[2][1]	CELL[2][2]
CELL[3][0]	CELL[3][1]	CELL[3][2]
CELL[4][0]	CELL[4][1]	CELL[4][2]



Les tableaux et interaction avec le DOM

● EX-04A :

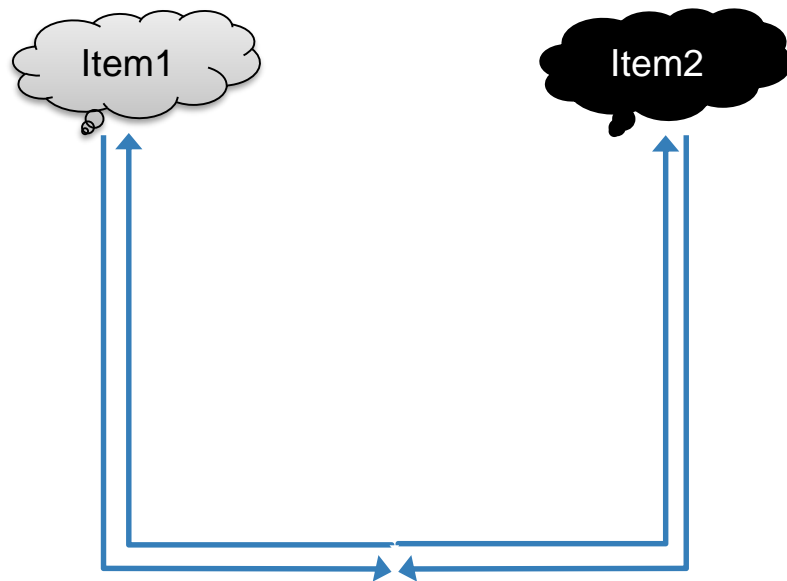
○ Notions :

- Création en mémoire d'un élément HTML DOM de type **table**, **tr**, **td** à l'aide de la méthode **document.createElement()** de l'API HTML DOM.
- Ajout d'un élément HTML DOM à un autre élément à l'aide de la méthode **appendChild()** ou via JQuery et **append()**.



Les tableaux et interaction avec le DOM

- 🌟 **EX-04B** (optionnel) : Via Anime.js (ou autre), mettez en oeuvre le concept de timeline pour faire bouger deux éléments, et pour chaque élément, mettez en oeuvre le concept de keyframes (via des Arrays) pour le faire voyager en plusieurs points. Essayez que cela soit esthétique. N'hésitez pas à jouer sur des éléments qui frôlent les bords du navigateur. Essayez-vous à un design responsive.





Les exceptions

- **EX-05A** : sur base de la démo précédente de validation d'un formulaire de login, trouvez une manière élégante d'intégrer la gestion d'exceptions afin de bénéficier du changement de couleurs de l'input qui ne serait pas valide :
 - rouge quand input NOK
 - vert quand input OK
- **EX-05B** (optionnel) : veuillez intégrer la « Constraint validation API » dans votre gestion de ce formulaire.



Références

- | | |
|-----|---|
| [1] | MDN web docs, Introduction to web APIs. Lien :
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction |
| [2] | MDN web docs, JavaScript Guide. Lien :
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide |
| [3] | w3schools.com, JavaScript Tutorial. Lien :
https://www.w3schools.com/js/default.asp |
| [4] | tutorialspoints.com, Javascript Tutorial : Lien :
https://www.tutorialspoint.com/javascript/index.htm |



Références

[5]	Medium.com, Neal Burger, The end of life of IE11. Lien : https://medium.com/@burger.neal/the-end-of-life-of-internet-explorer-11-12736f9ff75f
[6]	w3schools.com, JS HTML DOM. Lien : http://www.w3schools.com/js/js_htmlDOM.asp