

# Programmation Web – Avancé

**BINV2150 A : JavaScript (& JAVA SERVLETS)**

**Week 3**

**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

Bootstrap & callback avec argument(s)

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)



## HTML5 : Contraintes de Validation

- Nouveau concept en HTML5 : contraintes de validation
- Input obligatoire : attribut HTML **required**
- Validation par rapport à une RegEx : attribut HTML **pattern**
- Contrainte de longueur : attributs HTML **minlength** et **maxlength**



## HTML5 : Contraintes de Validation

- Sélectionner (pour changer le style) les inputs en fonction de leur statut de validation: CSS selector : **input:valid** ou **input:invalid**
- Gestion personnalisée de la validation (Messages d'erreurs personnalisés) via JS et la "constraint validation API" :  
**setCustomValidity()**

...



## HTML5 & CSS : Bootstrap

---

- Découverte de Bootstrap :  
<https://getbootstrap.com/docs/4.3/layout/grid/>
- Callback avec argument(s)
- Revue de l'exercice précédent



## HTML5 : Contraintes de Validation

- Détails : [https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Form\\_validation](https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Form_validation)
- **DEMO-04** : Reprise du formulaire précédent tout en réalisant la validation de la manière la plus simpliste via HTML/CSS.



## Les boucles

☉ Les boucles : **for**, **for/in**, **for/of**, **while**,  
**do/while**

```
myArray=[];  
for(let i=0 ; i < 100 ; i++){  
    myArray[i] = i;  
}
```



## Les boucles

---

- Exemples : [https://www.w3schools.com/js/js\\_loop\\_for.asp](https://www.w3schools.com/js/js_loop_for.asp)
- Détails : [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building\\_blocks/Looping\\_code](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Looping_code)



## Interaction avec l'API Canvas pour créer une animation

- **<canvas>** : élément HTML utilisé pour dessiner (2D ou 3D) via des scripts
- Créer une animation en 2 D sans librairie externe
  - HTML : Ajouter l'élément **<canvas>** :

```
<canvas id="Canva2D" ></canvas>
```





## Interaction avec l'API Canvas pour créer une animation

- CSS : définir le style du canvas et éventuellement du body (si pas de marge) :

```
<link rel="stylesheet" type="text/css" href="./DEMO-07.css">
```

- JS :
  - Accéder à l'objet canvas : **getElementById** ou **querySelector**

```
var myCanva = document.querySelector("canvas");
```



## Interaction avec l'API Canvas pour créer une animation

- JS :
  - Accéder au contexte de rendu du canvas pour pouvoir dessiner dessus en 2D :

```
var myContext = myCanva.getContext("2d");  
myContext.fillStyle = ...  
myContext.fillRect( ...
```

- Mise à jour d'une animation (mieux que **setInterval**) en appelant la fonction callback quand le browser est prêt (~60 callbacks / s):  
**window.requestAnimationFrame(callback)**



## Interaction avec l'API Canvas pour créer une animation

- Tutoriel sur l'élément **<canvas>** :  
[https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API/Tutorial](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial)
- Info sur la fonction **requestAnimationFrame()** :  
<http://javascriptkit.com/javatutors/requestanimationframe.shtml>



## Interaction avec l'API Canvas pour créer une animation

---

- **DEMO-05** : Générer une animation 2D de manière aléatoire reproduisant 101 fois un rectangle (ou une étoile) à des endroits différents du browser. L'animation est appelée suite au chargement de la page.



# Exercices

A vous de jouer...



## Interaction avec l'API Canvas pour créer une animation

- **EX-02A** : A partir de la démonstration précédente, retravailler l'animation pour :
  - qu'elle ne prenne qu'une partie de l'écran, en introduisant textuellement celle-ci au sein de votre page web ;
  - qu'elle s'arrête ou démarre lors d'un click d'un utilisateur ;
  - que les carrés grandissent ou rapetissent au clic sur la touche + ou la touche - ;



## Interaction avec l'API Canvas pour créer une animation

- **EX-02A** : A partir de la démonstration précédente, retravailler l'animation pour :
  - que lors d'un clic droit de la souris, la couleur change de manière aléatoire ; optionnellement, affichez une nouvelle forme lors d'un clic droit.



## Interaction avec l'API Canvas pour créer une animation

- ◎ **EX-02B** (optionnel) : Animer des balises DIV (pas besoin de l'objet **<canvas>**) afin d'afficher un nuage de mots se déplaçant de manière aléatoire, pas trop rapidement, afin de permettre à un utilisateur de cliquer dessus. Au clic sur l'élément, l'animation s'arrête et met en valeur cet élément.

NB : Vous pouvez soit reprendre la technologie précédente (**requestAnimationFrame()**), soit utiliser JQuery et la fonction **animate()**, soit utiliser une librairie JS (Anime.js, Three.js...).





## Interaction avec l'API Canvas pour créer une animation

- Info sur JQuery et la fonction **animate()** animant des propriétés CSS :  
[https://www.w3schools.com/jquery/eff\\_animate.asp](https://www.w3schools.com/jquery/eff_animate.asp)



## Références

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



## Références

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