

Wellcome !

Vous allez intégrer pour 2 séances un groupe.

Vous débutez en JS, bienvenue dans le groupe 1, sinon vous intégrez le groupe 2.
(Si vous hésitez encore sur votre niveau [->test](#))

 Un projet est à rendre. Le sujet est commun aux deux groupes.

Environnement VS + Nodejs		
groupe 1	groupe 2	groupe 3 ¹
introduction JS	Increase awareness of most helpful features, either new or old	Si vous avez des besoins en techno flex ou grid : contactez moi directement. Je vous accompagnerai.
Projet		

Quel que soit votre groupe, vous programmez avec l'environnement (NodeJS, VS, Chrome).

 Si vous communiquez en direct avec moi, vous intégrez votre code dans codepen ou [pythontutor](#).

Groupe 1

Un programme est un texte, l'histoire qu'il raconte un algorithme. La programmation son écriture. Et le langage varie suivant sa rédaction.

La rédaction dépendra du style de programmation (programmation impérative, fonctionnelle, programmation logique, programmation orientée objet ...)

Prenons un exemple : *"On doit retrouver la grand-mère maternelle, connaissant la mère"*.

¹ Mise à niveau sur CSS, pour ceux qui en ont besoin.

Le tableau suivant résumerait les différents styles d'écriture :

impératif	fonctionnelle	logique	objet
Partie Déclaration Y Partie Affectation Y = mère de X; Y = mère de Y;	Fonction afficher, mère Pour X afficher la mère de la mère de X	Prédicat mère(U,V) si mère(x,y) et mère(y,z) alors grand-mère(x,z)	Class individu méthode : mère grd-mère : [[this mère] mère]

La mauvaise nouvelle est qu'il va falloir ajouter une ième colonne. JS est un langage objet à prototype !

Il n'y a pas de classes en JS. La fonction est la pierre angulaire.

Mais, si on s'amuse à comparer les index des cours de langages de programmation²



Le constat serait frappant : les plans de cours se ressemblent.

En effet, si l'ordinateur représente les données en binaire, nous voulons penser nos données de façon plus naturelle (des nombres, des dates ...), les manipuler facilement à l'aide de boucles dans des structures de plus hauts niveaux (les tableaux ...) ; et ceci quelque soit le langage utilisé.

Il faut donc apprendre pour un langage les

1. Éléments lexicaux
2. Types
3. Structures de contrôle
4. Fonctions

Le plan du cours suit ce schéma⁴.

1. Les littéraux
 - a. Les chaînes de caractères (Strings) en JavaScript [MOL05]
 - b. Les nombres en JavaScript [MOL06]

² Il faudrait définir plus clairement ces langages (hormis XML, HTML, ...),

³ langage C openclassrooms.

⁴ Vous devez faire un effort pour vous mettre à niveau rapidement.

- c. Les booléens en JavaScript [M0L07]
- 2. Les Variables
 - a. Les variables en JavaScript [M0L08]
- 3. Les conditions
 - a. Les conditions en JavaScript [M0L10]
- 4. Les boucles
 - a. Les boucles en JavaScript [M0L13]
- 5. Les fonctions
 - a. Les fonctions en JavaScript (partie 1) [M0L15]
 - b. Les fonctions en JavaScript (partie 2) [M0L16]
 - c. Les fonctions en JavaScript (partie 3) [M0L17]

Vous avez au moins trois ans de programmation derrière vous. Si le nombre de vidéos est important, les notions abordées ne sont pas nouvelles.

Groupe 2

Il nous faudrait cinq ans pour faire "un tour du côté de chez Chrome".
(<https://developers.google.com/web>)

Je vous propose de travailler la vidéo de " Kayce Basques (@kaycebasques)".

Make a talk that is useful for both beginners and expert

- *Experts: Increase awareness of most helpful features, either new or old*

Cette vidéo dure 30 mn, La densité de son contenu, nous oblige à un gros travail d'assimilation de mise en pratique, tout ceci devrait nous prendre un minimum de 3H.

Voici le fichier source pour suivre les démo :

<https://touredevtools.glitch.me/20191008/#goals>

Ensuite, je vous propose un challenge : [trouvez le bug](#)⁵ !

Se rafraîchir la mémoire en js : [JS tout en Un](#)

 Si vous avez du temps : notez l'incroyable travail de synthèse sur les langages du WEB <https://youtu.be/0pThnRneDjw>

⁵ Vous comprendrez facilement la nature du bug !

Vous pouvez également aider le groupe 1 dans sa progression !

Projet

Création d'un jeu.

Règle du jeu : le joueur a trois chances pour trouver un nombre secret !

Imaginez que vous soyez chef de projet, vous rédigez en urgence un prototype pour les membres de votre équipe. Ce prototype permet à tous de fixer les idées ([proto](#)).

 Il faudra l'implémenter en JS.

-> Test pour connaître votre groupe

Etudiez le code JS suivant et répondez à la question :
Que vaut maxFinder.max ?

```
234
1. const maxFinder = {
2.     max: 0,
3.     find: function (numbers) {
4.         numbers.forEach(
5.             function (element) {
6.                 if (element > this.max) {
7.                     this.max = element;
8.                 }
9.             });
10. });
11.
12. maxFinder.find([2, 3, 4]);
13.
14. console.log(maxFinder.max);
```

-> [Une mauvaise surprise vous attend](#)

