Remédiation JS

1. JS au juste
2. Méthodes « string »
3. Méthodes Math.
4. Arrays (tableaux) et ses méthodes
5. Conditions (if – else if – else)
6. Variables temporaires
7. JS au juste :

Wikipedia 🡺 Dans un navigateur web, un script écrit en javascript peut être utilisé pour apporter une touche interactive ou dynamique (page ou site web), qui sans cela serait une page statique figée (exemple : site du chef). Le langage Javascript permet par exemple d'écrire des scripts pour afficher ou cacher un paragraphe, une image ou un popup, selon les interactions de l'utilisateur, ou d'informer le serveur du temps passé à lire une page.

BACKEND !!! Tout se passe derrière, très peu visuel, on ne code plus, on PROGRAMME, on doit des ordres précis à l’ordinateur dans un ordre précis afin d’arriver à un but précis. PROCESSUS

PROCESSUS ? Un **processus** est un ensemble d'activités corrélées ou en interaction qui utilise des éléments d'entrée pour produire un résultat escompté

Quand on codait, on pouvait donner les éléments de positionnement dans l’ordre qu’on voulait

Exemple : pose les fleurs sur l’étagère, en haut à gauche ou à gauche en haut, c’est pareil.

On avait très peu de contraintes d’ordres, si ce n’était de mettre les sections dans le bon ordre dans l’html et les blocs a l’interieur de bon bloc, que du placement de boites physiques, qui prenait de l’espace concrètement a l’ecran 🡺FRONTEND

La programmation c’est UN PROCESSUS, on doit exécuter obligatoirement les choses dans un ordre bien précis !!

Exemple : je vais à la boulangerie : je sors de chez moi, je traverse la rue, je rentre dans la boulangerie, je demande, j’achète, je sors, je rentre 🡺 si on met les étapes dans le désordre, ça ne MARCHE PLUS

Les variables :

Qu’est-ce qu’une variable ?

Comme son nom l’indique, c’est une donnée, une valeur, que le pc aura à sa disposition pour pouvoir interagir avec l’utilisateur en face de lui. Elle peut changer de valeur en fonction des conditions, d’où son nom de variable (qui varie).

Comment déclarer une variable :

* let nomDelavariable = donnéesDeLaVariable ; (le plus courant)
* var (pr déclarer des variables globales qui seront accessibles partout dans votre code, plus trop utilisé depuis invention du let assez récente)
* const (pour déclarer des variables qui ne changeront jamais de valeur (données) : CONSTANTE)

il existe 4 sortes/types de valeur pour les variables :

1. String : pour ficelle, ce sont les chaines de caractères, toujours entourées de guillemets (« « ) pour ne former qu’un bloc, une ficelle, string quoi
2. Int : pour integral, entiers en français, fait reference a tout l’ensemble des nombre entiers en mathématiques, en gros un chiffre ou un nombre
3. Booleens : boolens en française ça veut dire qui ne peut avoir que deux valeurs distinctes, en gros soit c’est blanc soit c’est noir, soit c’est VRAI (true), soit c’est FAUX (false). Sera utilisée pour vérifier si certaines conditions ont bien été respectées, ou non
4. Objet : Un [Objet](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/Object) est une structure contenant des données et des instructions en rapport avec ces données (comme une .fonction(accompagnéeDeSesParametresQuiL’Influencent)

Syntaxe base d’une commande (littéralement un ordre que l’on donne a la machine, on lui ordonne quelque chose, on lui commande de🡺processus):

variable.méthodeAttribuée(paramètresDeLaMéthode) ;

Comment faut-il le lire ?

Je prends un objet (un mot/phrase, un chiffre, un tableau etc)

Je lui applique une méthode (c’est-à-dire des morceaux de code préprogrammés qui permettent d’executer certaines tâches repetitives rapidement au lieu d’à chaque fois les recoder, un peu comme bootstrap)

A cette méthode, j’ajoute des paramètres supplémentaires, des données, qui lui permettent de les faire influencer d’une certaine manière sur l’objet de base)

Ici en gros, il prend la console, il lui demande de logger (se connecter en anglais) dessus son parametre qu’on lui a appliqué, à savoir la string « Hello World ».

notez que la variable n’est pas tjr nécessaire

Exemple de base : afficher « Hello World » (dans la console 🡺 backend) :

Hello world> premier mot que l’on apprend à afficher ds tous les langages de programmation, clin d’œil au fait qu’internet soit un réseau liant les réseaux du monde entier (International Network)

console.log(« Hello World ») ;

2.Méthodes « string » :

Point important : le plus important n’est pas de toutes les retenir par cœur, mais de savoir qu’il existe des commandes pour faire ce qu’elles font !!!

C’est très différent, les concepts sont plus faciles a retenir que les mots, si vous vous rappelez de ce que fait chaque méthode, il vous suffira de la googler pour la retrouver !

Voici quelques méthodes à appliquer aux valeurs « string » (à compléter ensemble !) :

* console.log() : pour afficher du texte dans la console ;
* .length : donne la longueur de la variable sur laquelle tu l’appliques
* .trim() : enleve les espaces blancs a la variable a laquelle il est appliqué
* .trimEnd() : ça fait pareil, mais à la fin
* .indexof() : trouver l’index du caractère de la string
* .charAt() : prend comme paramètre un index
* .toUpperCase() : permet de passer en majuscule

3. Méthodes Math.

Point important : bien mettre le « M » de Math en MAJUSCULE

Les méthodes prennent toujours des parenthèses car elles sensées être accompagnées de paramètres (interaction)!!!

Voici quelques méthodes (à compléter ensemble !) :

* Math.random() :
* Math.floor() :
* Math.ceil() :
* Math.round() :
* Math.trunc() :
* Math.abs() :
* Math.pow() :
* Math.sqrt() :

4.Arrays (tableaux) et ses méthodes

Les tableaux (en anglais arrays), est le nom que l’on donne aux variables que l’on va transformer en boites de rangement individuelles, comme une étagère, avec chacune case bien numérotées, dans lesquelles on pourra y placer des int, des string, etc etc

On les déclare comme les variables classiques, on les différenciera par les crochets [ ] qui entourent leurs valeurs.

let monTableau=[ unContenuDeMonTableau , »Un autre », 4538, true]

On peut lui appliquer plusieurs méthodes, en voici quelques-unes (à compléter ensemble !) :

* .pop() :
* .push() :
* .shift() :
* .unshift() :
* .replace() :
* .reverse() :
* .indexOf() :
* .slice() :
* .splice() :
* .sort() :

5. Conditions (if – else if – else)

C’est là où on commence à découvrir comment la programmation a été pensée (et qui la différencie du codage) 🡺 la logique humaine du processus dont on parlait au début

If = Si

Else if = Sinon si

Else = Sinon

Exemple :

Je me coupe le doigt en coupant mes légumes

SI (je ne suis pas coupé profondément) {ALORS

je me rince ;

}

SINON SI (je ne suis pas coupé très profondément, mais quand même) {ALORS

Je me rince ;

Je me mets un pansement ;

}

SINON (sous entendu tous les autres cas possible, dans ce cas-ci je me suis presque arraché le doigt sa mèèèèèèèèèère) {ALORS

JE COURS A L HOPITAL ;

}

Avec les conditions, vous pouvez donc déterminer les comportements qu’aura votre code en fonction de ce que l’utilisateur encodera 🡺 INTERACTION, PROCESSUS

Les opérateurs logiques :

&& : et

|| : ou

== : egal

=== : egalité stricte (contenu et type de valeur identiques !!! donc si tu as un «  12 » et un 12, ça ne sera pas strictement pareil car l’un est string et l’autre est int !!)

!= :inégal,différent

!== : inégalité stricte

> : plus grand que

< : plus petit que

>= : plus ou egal

<= : plus petit ou egal

Tous les symboles de calculs classiques sont également valables (addition + soustraction – division / multiplication \*)