

JAVASCRIPT

C'est un langage côté client qui s'exécute dans le navigateur comme le HTML et le CSS

Le contenu de notre page peut être mis à jour dynamiquement

Les trois langages s'empilent

JS= ensemble d'éléments qu'on utilise pour mettre en place des fonctionnalités sur nos pages web.

Le JS peut être utilisé avec des API et Bibliothèques, ce sont des structures complexes déjà prêtes à l'utilisation

-API = Application Programming Interface ou Interface de Programmation ensemble de codes pour réaliser certaines actions : google maps (on l'utilise mais on ne la manipule pas directement)

-Library = fichiers de codes JS structure de codes prêts à l'emploi : JQUERY (fichiers à télécharger)

JS s'écrit de préférence dans un fichier séparé comme le CSS , il est lié au HTML comme le CSS :

```
<script type="text/javascript" src="main.js"></script>
```

Cette ligne est à mettre de préférence à la fin du HTML, juste avant la fermeture du body.

Si on veut mettre des éléments script dans le HTML, mettre les instructions entre deux balises <script></script> et les mettre à la fin, avant la fermeture du body car le JS doit savoir quels éléments il doit traiter.

Les commentaires : multi lignes /* hxydhgskajxdhg*/
Mono ligne //

Indentation : toujours aussi importante.

Le code JavaScript est composé **d'instructions** : par exemple, on déclare une variable ou une fonction , on crée une boucle ou une condition

A la fin de chaque instruction « ; »

Une variable est un conteneur qui stocke des informations de manière temporaire, elle peut varier, on peut considérer une variable comme une boîte dans laquelle on met différentes choses.

Lorsqu'on crée une variable, on **déclare** une variable

Lorsqu'on stocke une valeur dans une variable, on **assigne** une valeur.

Pour déclarer une variable on utilise le mot clé **var** ou **let** suivi du nom qu'on voudra donner

Le nom de la variable peut commencer par une lettre ou _ mais pas par un chiffre , pas de caractères spéciaux, pas d'espaces.

JS est sensible à la casse, on utilise très souvent le lower camel case « lowerCamelCase » syntaxe permettant de mettre plusieurs mots collés tout en pouvant lire assez facilement grâce au majuscules.

Attention ! certains mots sont réservés car JS les utilise déjà pour désigner des éléments de code vous les découvrirez au fur et à mesure.

Une **constante** se déclare exactement comme une variable : `const prenom= 'Anne' ;`
C'est également un conteneur pour une valeur mais celle ci ne peut pas être modifiée.

EXO 1-JS

Découvrir JS, vous pouvez interagir sur une page. Ne cherchez pas à comprendre tout le code, il s'agit là de vous faire saisir l'interaction que vous pouvez avoir avec votre page. Ouvrez d'abord le HTML avec le navigateur, voyez ce qui se passe au clic et allez ensuite voir le code.

Dans le HTML, j'ai bien ajouté le lien vers le fichier JS (fichier enregistré comme on fait pour le CSS, dans le même dossier mais en .js nommé « main.js » par convention.)

Dans mon `<input>`, j'ai indiqué qu'à l'action « `onClick` », j'exécutais la fonction « `afficheMessage` » que j'ai définie dans le JS.

EXO JS-2 syntaxe des variables

Ouvrir JS-2 dans VSCode

Création d'une variable : `var variable` ou `let maVariable` par ex créée mais sans valeur
On **assigne** une valeur pour la première fois : on **l'initialise**.

On initialise la variable après l'avoir déclarée ou en même temps que sa déclaration.

On utilise `=` qui est l'opérateur d'assignation ou d'affectation ce n'est pas le `=` mathématique

`let` et `var` sont utilisés pour la déclaration, quand on manipule la variable, on utilise son nom.

Pour modifier la valeur de la variable, on utilise à nouveau l'opérateur d'affectation la nouvelle valeur écrase l'ancienne.

Cet exercice est juste pour vous familiariser avec la syntaxe recopier le juste dessous pour mémoriser.

EXO JS-3

Les données sont manipulées différemment selon leur type (liste dans le mémo)
Les principaux types sont :

String : chaîne de caractères : texte s

toute valeur stockée avec des guillemets " ou ' sera de type string par ex « 76 » ou « maison » sont tous 2 de type string on peut indifféremment utiliser les guillemets simples ou doubles, le principal c'est d'utiliser toujours les mêmes dans le code pour éviter les confusions ou les erreurs (pour échapper une apostrophe utiliser l'antislash comme on l'a vu dans le premier exercice).

Number : pas de guillemet ni d'apostrophe, positif négatif ou décimal

Boolean : valeur binaire 0 ou 1 en algèbre, true ou false en informatique.

Null : absence de valeur, très pratique dans l'algorithme quand on a besoin d'indiquer l'absence d'une valeur connue.

Undefined : Non définie

Dans cet exercice, on voit comment demander le type de données.

Le + est utilisé pour afficher les éléments les uns à la suite des autres c'est ce que l'on appelle la **concaténation**.

EXO JS-4

Attention au = :

Opérateurs d'affectation pour effectuer des actions sur les variables et leurs valeurs

Opérateurs arithmétiques : opérations mathématiques entre les valeurs de type number

On voit en même temps dans cet exo la concaténation et le \nb

JS-5 Priorité des calculs

Petit rappel des maths du collège ;)

1 Puissances

2 multiplication, division et modulo

3 addition et soustraction

Si même ordre de priorité, ordre d'écriture sauf les puissances (dans l'autre sens)

JS-6

Changement d'affichage pour cet exercice, on ne demande plus d'ouvrir une boîte d'alerte, les résultats s'affichent dans la console (avec l'inspecteur, 2^{ème} onglet sur chrome « console ») avec consol.log.

Opérateurs d'affectation :

= donne une valeur

Opérateur combiné effectue une opération et affecte en même temps.

les opérateurs combinés par ex += permettent d'ajouter une valeur à la valeur contenue dans la variable et affecte ce résultat dans la variable, la nouvelle valeur écrase l'ancienne. (les opérateurs combinés sont récapitulés dans le mémo)

JS-7 Concaténation

Dans le let x le + est un opérateur arithmétique pour additionner car les 2 opérandes (éléments situés à droite et à gauche de l'opérateur) sont des nombres.

Dans let z + est utilisé pour mettre bout à bout, pour concaténer la valeur de let x et le mot stagiaires

Ouvrez les 4 exercices de « exos-sup » et épluchez le code, décortiquez la syntaxe.
Allez voir d'autres exemples dans la doc, sur « openclassroom » ou « Pierre Giraud »

Si vous avez fini et que vous voulez aller plus loin:

Regardez la suite juste par curiosité comment cela fonctionne, allez éventuellement sur la doc. Souhila vous expliquera plus en détail.

STRUCTURES DE CONTRÔLE : **CONDITIONS**

Conditions : essentielles pour rendre un site dynamique,

On va pouvoir exécuter différentes actions si une condition donnée est vérifiée.

On exécute un code si une variable contient telle valeur, sinon on exécute un autre code .

On va donc comparer la valeur d'une variable à une autre valeur donnée.

Voir tableau des opérateurs de comparaison dans le memo-js.

JS-8

Avec ces opérateurs de comparaison, on demande à JS de comparer les 2 opérandes et de nous dire si la comparaison est vérifiée si oui : true sinon false.

Condition if = la plus simple

Le code s'exécute si le résultat est true :

JS-9

C'est la première condition qui est vraie donc c'est le premier code qui s'exécute.

JS-10

On peut avoir besoin d'inverser la valeur logique du test et vouloir exécuter le code si le résultat est false. Il suffit de comparer le résultat false à false `if(false==false)` , qui est évalué à true donc le code s'exécute .

Condition if...else

Condition plus complète car alternative si...sinon

JS-11

Pas de condition précisée au sein du else car tous les autres cas sont pris en charge.

Condition if...else if...else

La plus complète car on peut considérer autant de cas qu'on le souhaite.

JS-12

Toujours terminer la condition par un else . Si plusieurs else if sont évalués à true, seul le code du premier else if sera exécuté les autres tests seront ignorés.

LES OPERATEURS LOGIQUES

JS-13

ET &&

OU ||

NON !

Switch.

JS-14

L'instruction `switch` évalue une expression et, selon le résultat obtenu et le cas associé, exécute les instructions correspondantes.

For

JS-15

Le principe des boucles.