Formation

Notre expertise est votre avenir

JavaScript initiation Travaux pratiques

JVS-IN

Sommaire

1. Les Bases de JavaScript

1.1 Variables

Point cours : Pour voir le résultat de la fonction console.log(), ouvrez les outils en tapant F12 et affichez la console.

- Créez deux variables contenant des nombres. Affichez à l'aide d'un console.log().

- Ajoutez une variable ***Resultat*** qui contiendra la somme de ces 2 variables. Affichez cette variable ***Resultat***.

- Affichez le résultat de la soustraction, de la multiplication, de la division et du modulo des deux variables à l'aide de la fonction console.log()

- Á l'aide de votre première variable numérique et de la fonction console.log(), affichez le message "laVariable contient *valeurDeLaVariable*"

- Á partir de seulement deux variables contenant les chaînes de caractères "Hello" et " World", affichez "Hello World".

Options

- Créez deux variables var1 et var2 contenant chacune un nombre ainsi qu'une variable ***Resultat*** à laquelle vous affecterez la valeur de la somme des deux variables. Créez une quatrième variable qui contiendra la chaine de caractères "valeurDeVar1 + valeurDeVar2 = valeurDeResultat". Afficher ce message à l'aide d'un console.log()

- Créez deux variables auxquelles vous affecterez les valeurs de votre choix. Intervertissez leurs valeurs (à la fin var1 doit contenir valeurDeVar2 et inversement).

- Créez une variable contenant une donnée de type nombre, vérifiez ce fait à l'aide de la fonction console.log() et l'opérateur typeof. Changez le type de la variable à l'aide de la fonction String() et vérifiez que le changement a bien fonctionné grâce à console.log() et typeof.

- En tapant F12 vous avez accès à d'autres outils que la console. Observez les possibilités offertes et ce dans différents navigateurs.

1.2 Conditions et boucles

Point cours : Les fonctions Prompt() et Confirm().

Pour que l'utilisateur puisse entrer la valeur d'une variable à la volée : variableARemplir = prompt("texte à afficher");
Lors de l'exécution de ce code, une fenêtre s'ouvrira, indiquant à l'utilisateur "texte à afficher" et proposant un champ de texte. La valeur que prendra la variable sera celle indiquée dans le champ de texte.

La fonction variableARemplir = confirm("texte à afficher"); fonctionne de manière similaire, mais au lieu d'un champ de texte, la fenêtre propose deux boutons : "ok" et "annuler". Si l'utilisateur clique sur "ok", variableARemplir prendra la valeur True, s'il clique sur "annuler", elle prendra la valeur False.

- Demandez un nombre à l'utilisateur à l'aide de la fonction prompt(), ce sera le nombre de personnes invitées à un repas. En fonction de ce nombre, indiquez s'il s'agit d'un repas en solitaire, d'un repas intime (4 personnes max), d'un rassemblement (12 personnes max) ou d'un banquet. Faites-en sorte que les saisies de nombres négatifs affichent également quelque chose de logique.

- Réaliser, à l'aide de conditions, des boucles qui comptent de 0 à 10 et qui n'affichent que les nombres pairs. Une avec une boucle while et une avec une boucle for.

- Créez une variable chaine de caractères vide (contenant ""). Á l'aide d'une boucle, remplissez la de dix fois le caractère étoile (*) et affichez-la à l'aide d'un alert().

Options

- Trouvez maintenant une solution pour réaliser une boucle comptant de 0 à 10 dans une boucle while(true){} (utiliser un while(true) dans un vrai code est une mauvaise pratique dans la réalité ;)).

Point cours : Pour représenter un saut de ligne utiliser la chaîne de caractères "\n".

- Á partir de l'exercice de la ligne d'étoiles et à l'aide de deux boucles imbriquées, entrez maintenant dans la variable une chaîne de caractères représentant un rectangle de 10 X 10 étoiles (**********\n**********\n*****…).

Sur le même principe que l'exercice précédent réaliser un triangle rectangle contenant une étoile à la première ligne, deux étoiles à la seconde jusqu'à dix à la dixième. Réalisez maintenant ce triangle dans l'autre sens (10 étoiles sur la première ligne et une sur la dernière). Enfin, cherchez la manière la plus élégante et concise possible pour afficher un losange.

1.3 Fonctions et tableaux

- Créez la fonction soustraction (nombre1, nombre2), prenant donc deux nombres en argument et qui retourne la soustraction de ces 2 nombres. Testez votre fonction en affichant son résultat.

- Créez un tableau qui contient quatre éléments de votre choix. Afficher le second élément du tableau.

- Utilisez maintenant une boucle pour afficher tout le contenu du tableau.

- Créez une fonction lectureTableau(tableau) qui prend un tableau en argument et qui en affiche tout le contenu à l'aide d'une boucle.

1.4 Exercices complets

Point cours : Obtenir une lettre de l'alphabet aléatoirement.
Pour générer une lettre aléatoire, il faut :

- Avoir la liste de toutes les lettres
- En sélectionner une au hasard

Pour avoir la liste des lettres :

- Créez une variable contenant toutes les lettres (alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";)
- On peut utiliser une variable contenant une chaîne de caractères de manière similaire à un tableau : alphabet[x] renverra la (x+1)ième lettre de l'alphabet !

Pour choisir une lettre aléatoire :

- Il suffit d'utiliser la ligne de code alphabet[Math.floor(Math.random()*X)]. Math.floor(Math.random()*X qui renvoyant un nombre entre 0 et X-1.

Explication : Math.random() renvoie un nombre aléatoire entre 0 et 1 (par exemple 0.7861447615), si on le multiplie par X, on obtient un nombre décimal aléatoire entre 0 et X. Et on injecte ce nombre dans Math.floor() qui enlève la partie décimale des nombres. On obtient donc un nombre aléatoire entre 0 et X-1.

Pour ceux qui aiment être précis :

- Vu qu'il s'agit d'un nombre aléatoire obtenu par algorithme, il ne s'agit pas réellement d'un nombre aléatoire, mais d'un nombre pseudo aléatoire. Dans notre cas, ça n'a aucune importance.

- Réalisez une fonction motAleatoire() génératrice de mots de 7 lettres aléatoires. Par exemple "ADGRTP", "HYETRPO" ou encore "CHIANTI"

- Créez une fonction listeAleatoire() retournant un tableau contenant 25 mots de 7 lettres aléatoires.

Options

- Á partir du générateur de mots aléatoires créé précédemment, Créez une fonction prenant un tableau de mots en argument et retournant le nombre de ces mots dont la 4e lettre est une voyelle.

Point maths : factoriel(nombre) vaut nombre * nombre-1 * nombre -2 ... 3 * 2 * 1

- Créez la fonction Factoriel(nombre) qui renvoie la factorielle du nombre donné en argument.

- Créez une fonction nombresPremiers(nombre) qui renvoie un tableau contenant tous les nombres premiers plus petits ou égaux à nombre.

Optimisez votre fonction pour qu'elle puisse afficher le résultat de nombresPremiers(1 000 000 000) en moins de une minute. N'hésitez pas à chercher sur Internet des algorithmes pour optimiser la recherche des nombres premiers.
Au passage, plutôt que de chronométrer, utilisez le code suivant pour mesurer le temps d'exécution de votre script :

```
var startTime = new Date().getTime();
var elapsedTime = 0;
// votre code à mesurer ici
elapsedTime = new Date().getTime() - startTime;
```

- Si vous avez terminé vous pouvez passer à la partie concernant les timers.

2. Timers

Pour attendre une durée précise avant de lancer un évènement, on utilise le plus souvent en JavaScript la fonction setTimeout(fonctionToLaunch,duree);
Où durée est une durée en millisecondes et fonctionToLaunch la fonction à appeler à la fin du timer.
FonctionToLaunch peut être écrit de deux manières :

- Si cette fonction ne prend pas d'argument, indiquez uniquement le nom de la fonction sans parenthèses.
- Si cette fonction nécessite des arguments il faut utiliser une fonction anonyme. Par exemple remplacer fonctionToLaunch par : function() { fonctionALancer(argument) };

- Créez une page contenant uniquement un nombre qui compte les secondes qui passent.

- Faites en sorte que ce nombre s'affiche en lettres (pour les 10 premières secondes).

Option (après interactions avec HTML)

- Remplacez les nombres par des images (pack de chiffres fournit) pour toutes les valeurs.

3. Agir sur HTML avec JavaScript

3.1 Interactions directes

Ces exercices sont à effectuer sur la page d'exemple fournie en utilisant du code JavaScript

- Affichez le Titre de la page (balise avec ID=Titre)

- Modifiez le Titre de la page

Point cours : Pour modifier l'image d'une balise img, utiliser document.getElementById pour la capturer dans varBalise et modifier la valeur varBalise.src

- Faites-en sorte qu'au lieu de l'image chaton.jpg ce soit l'image chiot.jpg (fournie dans le même dossier) qui s'affiche.

Point cours : Pour changer le surlignement d'un élément, donnez au paramètre de style "backgroundColor" la valeur "red" (varBalise.style.backgroundColor)

- Faîtes en sorte que le Paragraphe1 soit surligné en rouge

3.2 Les évènements

Exercices à réaliser sur la page d'exemple :

- Faites-en sorte que cliquer sur le titre le modifie.

- Faites- en sorte que cliquer sur le chaton le change en chiot.

Exercices à réaliser sur la page couleurs.html :
La page Couleurs.html contient 3 affichant une couleur et un contenant le texte "Rien !"

- Créez une fonction changeSpan(texte) qui modifie le contenu du pour qu'il affiche la valeur de l'argument "texte".

- Faites-en sorte que le indique la couleur du dernier carré cliqué.

Options

Sur la page d'exemple

- Faire en sorte que cliquer sur le chaton le change en chiot et que cliquer sur le chiot nouvellement apparu le change en chaton et ce indéfiniment.

- Faire en sorte que passer le curseur sur le chaton le change en chiot et que sortir le curseur du chiot le rechange en chaton et ce indéfiniment

Point cours :
document.getElementsByTagName() prend en argument un type de balise (div, p, li …) et renvoie un tableau contenant toutes les occurrences de la balise

- Utilisez document.getElementsByTagName() pour créer une variable tableau contenant tous les éléments <p> de la page. Créez une boucle qui permette d'afficher le texte de chacun de ces éléments (pour plus de lisibilité, utiliser textContent plutôt que innerHTML).

Point cours :
document.querySelectorAll() renvoie un tableau contenant tous les éléments HTML correspondant à un sélecteur CSS. Par exemple, document.querySelectorAll(".menu div") enverrait un tableau contenant toutes les balises contenues dans une classe "menu". En sélectionnant la balise <a> (et pas la balise <li>) on peut accéder au et modifier le lien avec variableBaliseLien.href

- Sur la page Contact, utilisez document.querySelectorAll() pour créer une variable tableau contenant tous les liens à partir de la classe ".navbar" . Créez une boucle affichant les urls vers lesquels pointent les liens et les modifie pour qu'ils pointent tous vers google.fr.

- Sur la page Couleurs.html faites-en sorte que cliquer sur deux couleurs à la suite indique le mélange des deux couleurs (cliquer sur rouge indique "rouge" puis cliquer sur bleu indique "violet").

4. CSS

Sur la page d'exemple

- Faites-en sorte que les lettres du titre de la page soient maintenant rouges

Point cours : Pour assigner une taille en pixels il faut écrire varBalise.style.elementCSS = "TAILLEpx" où TAILLE est un nombre.

- Rendez-le chaton immense (d'une taille de 1000 par 1000 pixels).

Options (nécessite d'avoir fait les exercices sur les Timers)

La page DHTML.html affiche une petite animation de 2 rectangles qui se déplacent deux fois.

- Étudiez le code de la page pour qu'il y est maintenant 3 rectangles qui se déplacent 3 fois.
- Faites-en sorte que les éléments changent de couleur lorsque l'on clique dessus.
- Faites-en sorte que les éléments changent de couleurs lorsque l'on clique dessus dans un panel de 3 couleurs aléatoires (vous aurez besoin de créer au moins une fonction).
- Faites-en sorte que les éléments bougent au hasard dans une limite de 1000*1000 pixels et que leur taille (hauteur et largeur) varie au hasard entre 20 et 200.
- (Après le cours sur les objets) Réécrire ce code en créant des objets Carre et Rectangle contenant les propriétés stockElementHTML (contenant l'élément récupéré par getElementById), hauteur (, largeur pour les rectangles), couleur, position à gauche, position en haut et la méthode replacer() qui mettra à jour les données entrée sur la page.

5. JavaScript objet

5.1 Le garage de motos

- Créez un objet Moto avec comme propriétés : marque, modèle, couleur, prix. Ajoutez la méthode description() qui affiche un petit texte indiquant ces propriétés. Créez deux instances de Moto et testez que vous arrivez bien à avoir accès à leurs propriétés et à la méthode.

Point cours : Pour écrire une méthode recevant un argument, la grammaire est celle-ci : this.nomMethode = function(argument)

- Créez la méthode repeindre(nouvelleCouleur) qui change la couleur actuelle de la Moto par nouvelleCouleur. Assurez-vous grâce à description() que la couleur a correctement changé.

Options

- Créez maintenant le nouvel objet GarageMotos avec comme propriété un tableau de Motos. Ajoutez lui la fonction inventaire() qui appelle la méthode description() de chacune des Motos.

- Ajoutez à l'objet GarageMotos la propriété valeurStock initialisée à 0 et la méthode calculerValeurStock() qui calcule la somme cumulée des prix des Motos et qui renseigne cette valeur dans la propriété valeurStock. Appelez calculerValeurStock dans le constructeur. Testez.

6. DOM

Ces exercices sont à effectuer sur la page PageDOM.html

- En partant du "Corpsdepage", accédez au noeud "Paragraphe1". Vérifiez que vous êtes sur le bon nœud à l'aide de .prop('nodeName') et de .html()

Options

- Á l'aide d'une boucle listez le type et le contenu de tous les enfants de "Corpsdepage".
- Ajoutez avant l'image un paragraphe indiquant "c'est un dragon".

7. Le jeu du fermier

7.1 Concept du jeu

Le joueur est un fermier possédant initialement un champ de carottes. Son champ produisant une carotte par seconde qui est directement stockée. Quand le joueur le décide il peut vendre ses légumes. Grâce à l'argent cumulé il peut acheter d'autres champs pour augmenter sa productivité. Le joueur gagne lorsqu'il atteint une certaine somme d'argent.

Vous sont fournis :

- Une page index.html qu'il ne sera pas nécessaire de modifier.
- Un fichier style.css pour améliorer un peu l'affichage.
- Un fichier main.js contenant les principales variables à utiliser et des fonctions vides pour vous aider à démarrer.

Mais sentez-vous libre de partir from scratch (à partir de rien) ou sans main.js

7.2 Évolution de la partie

En début de partie, le joueur ne voit que l'affichage fournit dont un unique bouton "Vendre ses légumes". Cliquer sur ce bouton vend tous les légumes (mets leur nombre à 0 et incrémente l'argent total) et indique sur le "messageBoard" la valeur des légumes vendus.
Lorsque le joueur a accumulé suffisamment d'argent pour acheter un autre champ de carottes, un nouveau bouton "Acheter un champ de carottes" apparaît et permet de faire la transaction.
Une fois que le joueur a assez d'argent pour acheter un champ d'avocats, un nouveau bouton "Acheter un champ d'avocats" apparaît, similaire au précédent. Si le joueur achète un champ d'avocats, les données "nombre avocats" et "nombre de champs d'avocats" apparaissent dans l'affichage.
Le jeu doit détecter automatiquement si le joueur a gagné et l'afficher.

7.3 Pour commencer le développement

En l'état, au lancement de index.html, les graphismes s'affichent, la fonction main() est appelée et l'exécution s'arrête (main() étant encore une fonction vide).
Je vous recommande de faire en sorte que la fonction main() appelle la fonction pousse() qui fera pousser une carotte et qui se rappellera elle-même toutes les secondes.
Mettre alors à jour l'affichage en remplissant la fonction miseAJourAffichage().
Vous devriez alors avoir une carotte qui pousse par seconde. Il vous reste à coder la vente des légumes, puis l'achat des champs...

Point cours :
Pour faire apparaitre les éléments concernant l'achat de champs et les avocats, il suffit de retirer l'élément de style ="display:none;".

7.4 Détails chiffrés

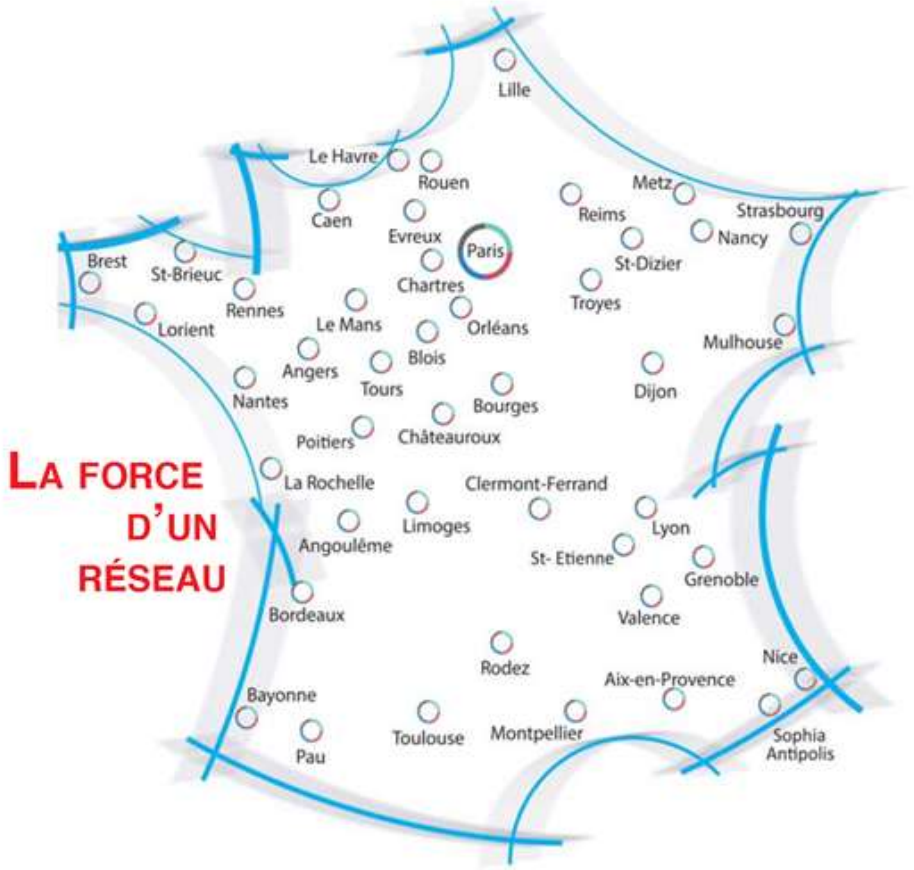
- Le joueur commence avec un champ de carottes, faisant pousser une carotte par seconde qu'il peut revendre 0.5$ pièce.
- Il peut acheter d'autres champs de carottes à 10$. Chaque champ de carotte faisant pousser une nouvelle carotte par seconde.
- Il peut également acheter des champs d'avocats à 20$, faisant pousser un avocat toutes les 3 secondes et qu'il peut vendre 4$.
- Le joueur gagne lorsqu'il a accumulé 1000$.

Options

- Ajoutez l'option "produire bio" coutant un total de 5$ par champs faisant que ceux-ci produisent deux fois plus lentement mais se vendent trois fois plus chers. Activer cette option vend tous les légumes actuellement possédés. Cette option fait également que le fond blanc du jeu sera maintenant composé de fleurs.

- Ajoutez à l'affichage une ligne indiquant la rentabilité à la seconde de tous les champs du joueur.

- Utilisez la balise <video> pour afficher la vidéo fournie au moment de la victoire.



Formation

N°Azur 0 810 007 689

PRIX D'UN APPEL LOCAL DEPUIS UN POSTE FIXE

Découvrez également l'ensemble des stages à votre disposition sur notre site

Site web

http://www.m2iformation.fr