

TP d'initiation au langage javascript

Objectifs :

Cette séquence a pour objectif d'écrire des programmes en javascript

- Écrire des programmes javascript
- Structures de base du langage

Moyens disponibles :

Un éditeur de texte pour la réalisation de la page HTML + javascript

Un navigateur (*Firefox* par exemple) pour visualiser des pages HTML réalisées

Des notes de cours sur le javascript.

Exercice 1 : calcul de maximums

objectifs de l'exercice:

- Utilisation structures conditionnelles
- Écrire une fonction
- Utilisation de la fonction de conversion de type `parseInt()`

Réaliser une fonction `max` qui renvoie le plus grand des 2 nombres qui lui sont passés en paramètre.

Réaliser une fonction `max3` qui renvoie le plus grand des 3 nombres qui lui sont passés en paramètre.

On pourra écrire deux versions, l'une en utilisant `max`, l'autre sans.

`parseInt(String Chaîne, [Integer base])` : convertit une chaîne en entier. Par défaut la base est 10.

Exemples :

`parseInt("100.5")` → 100

`parseInt("FF", 16)` → 255

<http://www.toutjavascript.com/reference/reference.php?iref=215>

Écrire une page HTML :

1. Demander la saisie de 2 nombres
Pour cela, on va ouvrir une boîte de dialogue.
`var nombre1 = prompt("Donnez un premier nombre", "");`
2. Afficher le max de ces 2 nombres
`document.writeln(" max de "+ nombre1 + " et "+ nombre2 + " = "+ max(nombre1, nombre2) + "
");`
3. Demander la saisie d'un troisième nombre
4. Afficher le max des 3 nombres calculé selon les 2 versions programmées

Résultat

Calcul de maximums

max de 5 et 9 = 9
max de 5, 9 et -1 = 9

Exercice 2 : Conversion de température

objectifs de l'exercice:

- Créer des fonctions
- Utilisation des opérateurs dynamiques
- Utilisation de la fonction de conversion de type `parseFloat()`

On souhaite créer une page qui permet la conversion d'une température exprimée en Fahrenheit en Celsius puis l'inverse.

Réaliser (= spécifier, commenter, coder et tester) une fonction qui prend en paramètre un nombre représentant une température en Fahrenheit et a pour résultat la température exprimée en °C équivalente.

Réaliser une fonction qui prend en paramètre un nombre représentant une température en Celsius et a pour résultat la température exprimée en Fahrenheit équivalente.

La relation de conversion $\text{TempFahrenheit} = 9/5 * \text{TempCelsius} + 32$;

Écrire une page html :

1. Demander la saisie d'une température en Fahrenheit et affiche la température Celsius équivalente
2. Demander la saisie d'une température en Celsius et affiche la température Fahrenheit équivalente

Résultat

Conversion Fahrenheit vers Celsius

la température 50 F vaut 10 °C

Conversion Celsius vers Fahrenheit

la température -5 °C vaut 122 F

Exercice 3 : Le jeu du nombre mystérieux

objectifs de l'exercice:

- Utilisation de la structure conditionnelle (if)
- Tests de comparaison
- Utilisation de l'objet primitif Math

Le jeu du nombre mystérieux consiste à deviner un nombre compris entre 0 et 100 en faisant des propositions successives. Après chaque proposition erronée, le joueur est informée si celle-ci était plus petite ou plus grande que le nombre à découvrir.

On souhaite créer une page qui permet de jouer à ce jeu. Pour la génération d'un nombre aléatoire voir **Math.random()**.

Math.random() : retourne un nombre aléatoire entre 0 et 1

Exemple pour générer un nombre entier compris entre 0 et 10:

```
var nombre = Math.floor((Math.random() * 10) + 1) ;
```

Réaliser une fonction qui :

1. Demande au joueur de proposer un nombre compris entre 0 et 100, la demande est répétée jusqu'à ce que la réponse respecte cette contrainte
2. Si la proposition n'est pas le nombre mystère affiche si elle est trop petite ou trop grande
3. Le résultat **true** si la proposition est le nombre mystère, **false** sinon

→ Réaliser une procédure jeu qui permet de jouer au jeu du nombre mystérieux et affiche le nombre de tentatives qui ont été nécessaires pour trouver le nombre mystérieux.

→ Écrire la page html qui permet de jouer à ce jeu et affiche combien de tentatives ont été nécessaires pour parvenir à trouver le nombre mystérieux.

Nombre mystère

50 est trop petit

Proposez un nombre entre 0 et 100

☐ Empêcher cette page d'ouvrir des dialogues supplémentaires

OK Annuler

Nombre mystère

50 est trop grand
 25 est trop petit
 37 est trop petit
 44 est trop petit
 47 est trop petit

Nombre mystère 49 trouvé en 6 tentatives

Exercice 4 : Affichage du code ASCII

objectifs de l'exercice:

- Utilisation des boucles itératives
- Utilisation de l'objet primitif String
- Utilisation de la fonction fromCharCode(i)
- Création d'un tableau HTML dynamique

On souhaite créer une page qui affiche la table des 128 premiers caractères de la table ASCII sous forme brute et dans une table HTML.

String.fromCharCode() : crée une chaîne à partir d'une série de code ASCII

Exemple pour générer ne chaîne de caractères :

```
document.write(String.fromCharCode(104, 101, 108, 108, 111))
```

Affiche -> Hello

Attention : String comme Math sont des classes **static**. Il faut préciser le nom de la classe avant la méthode. Ex String.charCodeAt('A') → retourne le code ASCII de 'A'.

→ Réaliser une procédure qui affiche les 128 premiers caractères de la table ASCII sous forme brute à raison de 16 lignes de 8 colonnes.

→ Réaliser une procédure qui affiche les 128 premiers caractères de la table ASCII dans une TABLE html à raison de 16 lignes de 8 colonnes.

→ Réaliser une procédure qui affiche les 128 premiers caractères de la table ASCII et le code en Hexa dans une TABLE html à raison de 32 lignes de 8 colonnes.

Vous pouvez mettre en place une CSS pour rendre l'affichage de la table plus "esthétique".
Écrire la page html qui contient les affichages de la table des 128 premiers caractères ASCII dans les 3 formats.

Exercice 5 : Le tri à bulle

objectifs de l'exercice:

- Utilisation des boucles itératives
- Mettre en œuvre l'algorithme de tri à bulle
- manipulation de tableaux

Le tri bulle est un algorithme de tri dont le principe consiste pour trier un tableau de n éléments, à parcourir dans l'ordre les cases du tableau en échangeant deux éléments successifs si le premier est plus grand que le second (soit échanger $t[i]$ et $t[i+1]$ si $t[i] > t[i+1]$), ce qui fait remonter "comme une bulle" le plus grand élément contenu dans ces n cases vers la dernière d'entre elles, où il est alors bien placé. Puis on recommence avec les $n-1$ premiers éléments du tableau (non triés) et ainsi de suite. De nombreuses vidéos illustrent le fonctionnement de ce tri, en [voici une](#), [une autre avec des légos](#) et il en existe de [très flokloriques](#) (tri effectif après 52"...).

Réaliser une fonction qui prend en paramètre un tableau d'entiers et modifie ce tableau pour que ses éléments soient triés par ordre croissant à la fin de l'exécution de la fonction.

Réutilisez la fonction qui permet de créer un tableau de n nombres entiers entre 0 et 100.

Écrire une page html qui :

1. demande le nombre d'éléments du tableau à créer puis remplit ce tableau avec des entiers choisis aléatoirement (`Math.random ()`)
2. affiche le tableau
3. affiche le tableau après l'avoir trié par l'algorithme du tri à bulles

Exercice 6 : opérations sur les notes d'étudiant

Recopiez dans votre fichier JavaScript les deux lignes suivantes

```
NomEleves = new Array("Dupont", "Durand", "Petit", "Martin", "Legrand",  
"Lacroix", "Legros", "Lenaim");
```

```
Notes = new Array([10,12,18,5,9,13,16,19],[13,11,14,7,14,16,12,10],  
[9,14,14,12,8,20,13,13]);
```

afin de disposer :

- d'un tableau listant les élèves d'une classe.
- d'un second tableau à 2 dimensions regroupant les notes des élèves.

1. Écrire une fonction **Afficher(TabNom, TabNotes)** qui affiche la liste d'élèves et leurs notes. Les paramètres de cette fonction sont respectivement le tableau de noms et le tableau de notes. Il faut utiliser des boucles imbriquées.

<http://www.toutjavascript.com/savoir/savoir09.php3>

Affichage du tableau de notes

```
Dupont - 10 - 13 - 9
Durand - 12 - 11 - 14
Petit - 18 - 14 - 14
Martin - 5 - 7 - 12
Legrand - 9 - 14 - 8
Lacroix - 13 - 16 - 20
Legros - 16 - 12 - 13
Lenaim - 19 - 10 - 13
```

Remarque : pour la mise en forme, cette fonction peut générer un tableau HTML.

Dupont	10	13	9
Durand	12	11	14
Petit	18	14	14
Martin	5	7	12
Legrand	9	14	8
Lacroix	13	16	20
Legros	16	12	13
Lenaim	19	10	13

2. Écrire une fonction **CalculMoyenneEleve(TabNom, TabNotes)** qui calcule la moyenne des notes pour chaque élève. Cette dernière affiche les résultats sous la forme d'un tableau.

Noms	Moyennes
Dupont	10.67
Durand	12.33
Petit	15.33
Martin	8
Legrand	10.33
Lacroix	16.33
Legros	13.67
Lenaim	14

3. Écrire une fonction **NoteMin(TabNotes)** qui recherche et retourne la valeur min du tableau de notes passé en paramètre.

Moyenne de la classe = 12.58

4. Écrire enfin une fonction **NoteMax(TabNotes)** qui recherche et retourne la valeur min du tableau de notes passé en paramètre.

La note maximale = 20

Écrire une page html qui :

- Appeler la fonction Afficher(TabNom, TabNotes)
- > Appeler la fonction CalculMoyenneEleve(TabNom,TabNotes)(TabNom, TabNotes)
- Appeler la fonction NoteMin () et afficher le résultat
- Appeler la fonction NoteMax () et afficher le résultat

Remarque : pour les deux dernières questions, on peut aussi afficher le nom de l'élève ayant la note min et la note max.