

TP 6

JavaScript et Web

Ce TP est un mélange de JavaScript et HTML. Aucun code CSS n'est demandé au cours de ce TP.



EXERCICE

Exercice 6.1 (Calcul de l'âge). Créer une page HTML minimale `calculeAge.html` (c'est-à-dire contenant le minimum de contenu pour être un code HTML valide), liée à un fichier JavaScript externe `calculeAge.js`. Le code HTML ne doit rien contenir, à part l'inclusion du code JavaScript. L'appel à la fonction JavaScript décrite ci-dessous devra se faire directement depuis le corps du fichier JavaScript, et non depuis le HTML.



Question 1 : Écrire une fonction `calculeAge(annee)` qui prend une année en paramètre et, en fonction de l'année courante, rend l'âge d'une personne née cette année-là.



Question 2 : Écrire une fonction `demandeEtCalculeAge()` qui va demander à l'internaute son année de naissance (à l'aide de `prompt`), et va afficher son âge au sein d'un paragraphe dans le document HTML (utiliser `document.write`). Penser à réutiliser la fonction `calculeAge()`.



Question 3 : Améliorer la fonction `demandeEtCalculeAge()` pour que, tant que l'internaute n'a pas entré une année correcte (c'est-à-dire un nombre entier positif inférieur ou égal à l'année courante), la fonction affiche une erreur (avec `alert`) et demande de nouveau d'entrer une année.





Question (optionnelle) 4 : Refaire l'exercice avec cette fois-ci 3 arguments : le jour de naissance, le mois de naissance et l'année de naissance.



EXERCICE

Exercice 6.2 (Calcul d'une moyenne de nombres). Créer une page HTML minimale `calculeMoyenne.html` (c'est-à-dire contenant le minimum de contenu pour être un code HTML valide), liée à un fichier JavaScript externe `Moyenne.js`. Le code HTML ne doit rien contenir, à part l'inclusion du code JavaScript. L'appel à la fonction JavaScript décrite ci-dessous devra se faire directement depuis le corps du fichier JavaScript, et non depuis le HTML.

 **Question 1 :** Écrire une fonction `saisirNombre(message)` qui demande à l'internaute de saisir un nombre, avec `prompt(message)` (où `message` est donc le message par défaut de la fenêtre du `prompt`). Tant que l'internaute n'a pas entré un nombre positif ou nul, la fonction affiche une erreur (avec `alert`) et demande de nouveau d'entrer un nombre.

 **Question 2 :** Écrire une fonction `calculeMoyenne()` qui demande à l'internaute d'entrer un nombre d'entiers `n`; puis demande à l'internaute d'entrer `n` entiers. Bien utiliser la fonction `saisirNombre(message)` ci-dessus.

Ensuite, pour chacun des `n` entiers, la fonction va l'afficher dans un paragraphe HTML. Enfin, la fonction va afficher la moyenne des `n` nombres en gras.

Exemple de rendu HTML après que l'internaute a entré 4, 2, 5, 4, 1 :


```
Entier 1 = 2
Entier 2 = 5
Entier 3 = 4
Entier 4 = 1
Moyenne = 3
```

Conseil : stocker les entiers dans un tableau, et utiliser un parcours de tableau avec `forEach`. Penser à réutiliser la fonction du calcul de moyenne du TP précédent.



EXERCICE

Exercice 6.3 (Vérification d'un code postal (optionnel)). Créer une page HTML minimale `verifCodePostal.html` (c'est-à-dire contenant le minimum de contenu pour être un code HTML valide), liée à un fichier JavaScript externe `verifCodePostal.js`. Le code HTML ne doit rien contenir, à part l'inclusion du code JavaScript. L'appel à la fonction JavaScript décrite ci-dessous devra se faire directement depuis le corps du fichier JavaScript, et non depuis le HTML.

 **Question (optionnelle) 1 :** Écrire une fonction `verifieCodePostal()` qui demande à l'internaute de saisir un code postal français, avec `prompt()`. Tant que l'internaute n'a pas entré un code postal valide, la fonction doit afficher une erreur (avec `alert`) et demander de nouveau d'entrer un nombre. Si le code postal est valide, celui-ci doit s'afficher dans la page HTML.

Tester avec au minimum avec les codes postaux valides suivants : 93430, 35000, 09110; et avec les codes invalides 9110, 9340x, -93430, 93 430, -9110.

