

Programmation web en js

TD n°2 : tableaux, objets

Exercice 1 : tableaux

1. écrire la fonction `range(a,b)` qui reçoit 2 paramètres entiers et retourne un tableau contenant tous les entiers de a à b (si $a \leq b$) ou de b à a (si $a > b$),
2. écrire une fonction `sum()` qui reçoit un tableau d'entiers et retourne la somme des éléments ; écrire une version utilisant une boucle "for", une version utilisant la fonction `forEach()` et une version utilisant la fonction `reduce()`,
3. écrire une fonction qui reçoit un tableau d'entier et retourne la moyenne des éléments. écrire une version utilisant une boucle "for", une version utilisant la fonction `forEach()` et une version utilisant la fonction `reduce()`,
4. écrire une fonction qui reçoit un tableau de chaînes de caractères (`t`) et une chaîne de caractères (`pattern`), et qui retourne un tableau composé des chaînes de `t` qui comportent le motif `pattern`, mises en majuscule. On utilisera `forEach()` .
5. généraliser la fonction précédente : écrire une fonction qui reçoit un tableau (`t`), une fonction de test `testFnct` et une fonction de transformation `transFnct` et retourne un tableau contenant les éléments de `t` pour lesquels `testFnct` retourne `true`, transformés par `transFnct`.

Utiliser cette fonction généralisée pour ré-écrire le fonction précédente.

6. ré-écrire la fonction (4) précédente en utilisant `filter()` et `map()`,
7. on considère un tableau contenant des références de produits dans un catalogue en ligne. Ecrire une fonction qui reçoit ce tableau en paramètre et retourne un tableau contenant les urls vers ces produits, chaque url étant de la forme :
http://www.cata.log/products/<reference_produit>.
8. Compléter la fonction précédente pour qu'elle retourne une chaîne de caractères correspondant à une fraction de code html affichant un lien `<a>` pour chaque produit du tableau reçu en paramètre :

```
"<div class="cata"><ul>
  <li> <a href="url"> ref_prod</a><li>
  <li> <a href="url"> ref_prod</a><li>
  ...
</ul> </div>"
```

Exercice 2 : objets

1. écrire une fonction qui reçoit un tableau d'entier et retourne un *objet* contenant 3 propriétés : le nombre d'éléments, la somme des éléments et la moyenne des éléments,
2. créer sous forme littérale un objet représentant un étudiant. Il contient les propriétés `numero`, `nom`, `prenom`, `dateNaiss`, `mail`, `notes`. La valeur de la propriété `dateNaiss` est construite avec le constructeur de dates. La valeur de la propriété `notes` est un tableau vide.
3. ajouter une méthode qui calcule l'âge de l'étudiant. Ajouter une méthode qui affiche les nom, prénom et date de naissance de l'étudiant. Le nom est en majuscule, la date est sous la forme `dd/mm/YYYY`.
4. ajouter une méthode qui ajoute une note dans une matière à l'étudiant. la méthode reçoit 2 paramètres `m` (matière) et `n` (note), en fait un objet contenant 2 propriétés, et ajoute cet objet dans le tableau de notes de l'étudiant ; on suppose que l'on a toujours 1 seule note par matière.
5. ajouter une méthode qui calcule la moyenne de l'étudiant (toutes les matières ont le coeff. 1),

6. créer un constructeur d'étudiants : il doit produire des objets ayant la même structure que celui ci-dessus. Les méthodes ci-dessus doivent être définies dans le prototype.
7. créer une fonction qui reçoit un tableau d'étudiants et un n° de mois et retourne un tableau contenant la liste des étudiants dont l'anniversaire a lieu durant le mois indiqué.
8. Créer une fonction qui retourne la liste des étudiants ayant plus d'un age donné.
9. Créer un constructeur d'objets correspondant à un groupe d'étudiants. Un groupe d'étudiants contient les propriétés nomgpe, formation, liste, annee.
10. Prévoir les méthodes pour : ajouter un étudiant au groupe, compter les étudiants d'un groupe, calculer la moyenne générale de chaque étudiant du groupe.
11. Souhaitez un bon anniversaire à tous les étudiants du groupe nés un mois donné en leur ajoutant 2 pts sur la note d'une matière donnée.