

# TP - Javascript

## 1. Événements

Testez les différents événements disponibles en javascript (onmouseover, onmouseout, onload, onclick, onfocus, onselect...) en effectuant différentes actions lors des événements (alert, console.log, document.write...).

## 2. Conditions et boucles

Ecrire une fonction permettant de jouer au jeu plus petit plus grand : un entier entre 0 et 1000 est tiré au sort par le navigateur (Math.random) que l'utilisateur doit deviner. A chaque proposition de l'utilisateur, vous devez indiquer si le nombre est plus petit, plus grand ou égal au nombre choisi.

```
var solution = null;

// generate a random number between 1 and 1000
function start () {...}

// check if the value in the #arg field is smaller, larger or equal to the solution
// display the result in the #result field
function check(){...}
```

## 3. Moteur de recherche

Construire un tableau en Javascript contenant des chaînes de caractères quelconques. Puis, à chaque fois que l'utilisateur tape une touche, la liste des réponses potentielles s'affiche dans une liste.

```
var answers = ['aa', 'ab', 'abc', 'abd', 'b']; // a set of potential answers

// return the subset of answers that start with query
function getAnswers(query) {...}

// display the answers that match the query in the #answers list
function displayAnswers(query) {...}
```

## 4. Champ éditable

Créez une fonction javascript telle que si l'on clique (onclick) sur le span#editable, cette balise se transforme en champ input dans lequel on peut modifier le contenu. Quand on clique en dehors de la balise (onblur) elle redevient un span avec le nouveau texte.

## 5. Fonctions d'ordre supérieur

Sur le modèle de la fonction map vue en cours (reprise ci-dessous dans une version plus correcte) :

```
function map (a,f) {
  var res = new Array();
  for (var i=0 ; i<a.length ; i++) {
    var v = a[i];
    res.push(f(v));
  }
  return res;
}
```

Ecrire une fonction filter qui retourne un tableau ne contenant que les éléments qui vérifient un prédicat, par exemple :

```
function filter(a, f) {...}
console.log(filter([1,2,3,4,5], function(a){return a%2==1;}));
```

Doit afficher dans la console le tableau contenant uniquement 1, 3 et 5.

## 6. Tableaux + JSON

Créer un format de stockage de tables en JSON, par exemple avec le format ci-dessous :

```
{
  "header": {
    "id": "Identifiant",
    "mail": "E-mail",
    "pass": "Mdp",
  },
  "content": [
    {"id": "1", "mail": "ab@t.com", "pass": "jhjf567Gjf"},
    {"id": "2", "mail": "cd@u.fr", "pass": "d567hgjf"},
    {"id": "3", "mail": "ef@v.de", "pass": "kf87GHJG"}
  ]
}
```

Générer automatiquement le code HTML du tableau en fonction du contenu du fichier JSON. Attention, le nombre de ligne et de colonnes est variable !

## 7. Validation de formulaire

Créez des fonctions de validation de formulaire simples (checkRequired, checkEmail, checkInteger, checkRegexp, etc.)

Appelez ces fonctions lors de la soumission (onsubmit) d'un formulaire que vous aurez créé.

Ecrivez une ou plusieurs fonctions javascript pour générer un formulaire ainsi qu'une validation de tous les champs lors de la soumission directement à partir d'un fichier json de description du formulaire, par exemple dans le format suivant :

```
{
  "id": {"name": "Identifiant", "props": ["numeric", "required"]},
  "mail": {"name": "E-mail", "props": ["email", "required"]},
  "pass": {"name": "Mdp", "props": ["required", "/^[A-Za-z0-9!@#$$%^&*()_]{6,20}$/"]}
}
```