

## TD Informatique – Technologies du Web - le langage Javascript

### Programmation événementielle

#### Exercice : formulaire de conversion

On se propose de réaliser l'outil de conversion suivant ; pour un prix saisi, l'utilisateur peut choisir de le convertir en francs ou en euros.

12	Euros >> Francs	Francs >> Euros	1.83 euros
----	-----------------	-----------------	------------

12	Euros >> Francs	Francs >> Euros	78.6 francs
----	-----------------	-----------------	-------------

```
<form>
  <input type="text" id="valeur" size="5" placeholder="prix">
  <input type="button" id="versfrancs" value="Euros >> Francs">
  <input type="button" id="verseuros" value="Francs >> Euros">
  <input type="text" id="resultat" size="15" readonly="true" placeholder="résultat">
</form>
```

Ecrire le script permettant de faire ces conversions.

#### Exercice : vérification saisie

L'objectif est d'afficher un message d'information au fur et à mesure de la saisie d'un mot de passe :

mot de passe	
...	longueur faible
.....	longueur moyenne
.....	longueur bonne

Du coup, dans le code HTML il faut prévoir une zone pour afficher les messages :

```
<form id="inscription">
  <input type="password" name="passwd" id="passwd" placeholder="mot de passe" required/>
  <span id="msgPasswd"></span>
</form>
```

L'événement permettant de réagir à chaque saisie clavier dans une zone texte est input.

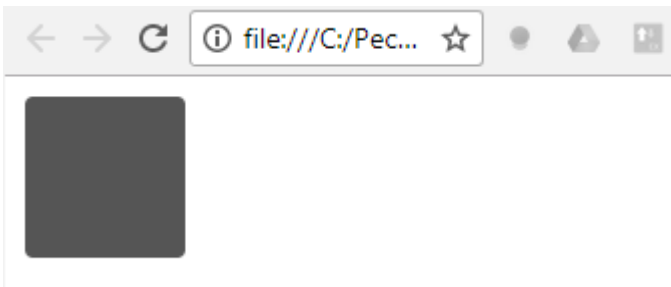
#### Exercice : déplacer un objet

Créer un fichier html contenant le code ci-dessous ; lui associer un fichier .js.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
```

```
<title>TP Javascript</title>
<style>
  div {
    background: #555555;
    border-radius: 4px;
    width: 80px; height: 80px;
    position: absolute;
    top: 10px; left: 10px; margin:0px;
  }
</style>
</head>
<body>
  <div id="c1"></div>
</body>
</html>
```

Ouvert dans le navigateur, le document devrait avoir cette apparence.



- a) Le but de l'exercice est déplacer la carte en utilisant les touches de direction du clavier. Les codes des touches de direction sont (propriété key de l'objet event) :

ArrowDown flèche bas,  
ArrowUp flèche haut,  
ArrowRight flèche droite,  
ArrowLeft flèche gauche

*Attention le type d'événement keypress ne fonctionne pas pour toutes les touches du clavier.*

Les propriétés `offsetLeft` et `offsetTop` permettent de récupérer les positions actuelles absolues de l'objet (en nombre de pixels).

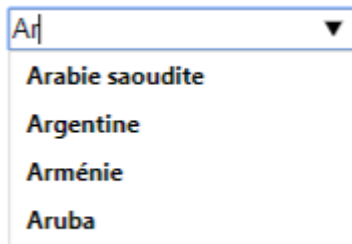
Les propriétés `style.left` et `style.top` permettent de définir les nouvelles positions d'un objet ; attention il faut écrire la valeur sous la forme « 100px »

- b) Ajouter la gestion des bords pour éviter que la carte disparaisse. Les propriétés `innerHeight` et `innerWidth` de l'objet `window` permettent de récupérer la largeur et la hauteur de la fenêtre.
- c) Maintenant il s'agit de déplacer la carte à chaque clic de la souris.
- d) Enfin Ajouter un effet de zoom (+10px sur la largeur et la hauteur) sur la carte si le clic a lieu sur la carte. (attention si on clique sur la carte elle ne doit pas bouger cf. Bubbling phase)  
*Remarque : tester 3 versions -> avec une référence à la carte, avec la cible de l'événement `event.target`, avec le mot clé `this`*

## Exercice : autocomplétion

Il s'agit de mettre en œuvre le système d'autocomplétion que l'on rencontre très fréquemment sur les sites et qui permet de faire des propositions de saisie à l'utilisateur.

Nous allons essayer de le mettre en œuvre pour une zone de saisie d'un pays.



A screenshot of a web form with an autocomplete feature. The input field contains the text 'Ar'. Below the input field, a dropdown menu is open, displaying a list of suggestions: 'Arabie saoudite', 'Argentine', 'Arménie', and 'Aruba'. The suggestions are listed in a simple, sans-serif font, with the first suggestion 'Arabie saoudite' appearing slightly more prominent than the others.

L'élément HTML permettant de faire cela est `datalist` ; le lien avec la zone de saisie se fait avec l'attribut `list`. Concrètement l'attribut `list` de l'input doit contenir l'id de la `datalist`.

```
<input type="text" name="pays" id="pays" list="proposition" placeholder="nom de pays"/>
<datalist id="proposition"></datalist>
```

Les propositions de la `datalist` devront apparaitre comme des éléments `<option>` (identique à `<select>`)

Mettre en œuvre l'autocomplétion en utilisant cette liste de pays pour tester :

```
const listePays = [ "Afghanistan", "Afrique du Sud", "Albanie", "Algérie", "Allemagne", "Andorre",
"Angola", "Anguilla", "Antarctique", "Antigua-et-Barbuda", "Antilles néerlandaises", "Arabie saoudite",
"Argentine", "Arménie", "Aruba", "Australie", "Autriche", "Azerbaïdjan"];
```