

## **TP1**

### **Exercice 1 : Calcul TTC**

Complétez le fichier **exo1.js** de façon que la page HTML **exo1.html** se comporte de la façon suivante :

- L'utilisateur saisit un montant hors taxe et un taux de TVA dans les champs prévus à cet effet, et à l'aide d'un clic sur le bouton, affiche ce montant augmenté de la TVA.
- L'affichage du résultat est réalisé en modifiant le contenu d'un élément HTML particulier et en le faisant apparaître alors qu'il était initialement invisible.
- Si les chaînes de caractères saisies par l'utilisateur ne sont pas des nombres, une erreur est signalée dans une alerte.

### **Indication :**

- 1) Les fonctions utiles pour réaliser cet exercice : alert, getElementById, isNaN, toFixed, Number.
- 2) Les attributs utiles à manipuler pour cet exercice : value, innerHTML, style.visibility

### **Exercice 2 : Appréciation**

Complétez le fichier **exo2.js** de façon que la page HTML **exo2.html** se comporte de la façon suivante :

- L'utilisateur saisit trois notes dans les champs prévus à cet effet et, à l'aide d'un clic sur le bouton, affiche la moyenne de ces notes ainsi que l'appréciation qui convient.
- Comme dans l'exercice précédent, le résultat apparaît dans un élément HTML prévu à cet effet.
- Si l'une des chaînes de caractères saisies par l'utilisateur n'est pas un nombre, une erreur est signalée dans une alerte.

L'appréciation est :

- "Très Faible" si la moyenne est strictement inférieure à 6.
- "Faible" si la moyenne est supérieure ou égale à 6 et strictement inférieure à 10.
- "Moyen" si la moyenne est supérieure ou égale à 10 et strictement inférieure à 13.
- "Bien" si la moyenne est supérieure ou égale à 13 et strictement inférieure à 16.
- "Très Bien" si la moyenne est supérieure ou égale à 16 et strictement inférieure à 19.
- "Excellent" si la moyenne est supérieure ou égale à 19.

**Indication :**

1. Les fonctions utiles pour réaliser cet exercice : alert, getElementById, isNaN, toFixed, Number.
2. Les attributs utiles à manipuler pour cet exercice : value, innerHTML, style.visibility

**Exercice 3 : Répétiteur de Multiplications**

On veut réaliser une page HTML qui serve de répétiteur de multiplications. Les multiplications successives sont proposées sur la même page HTML.

- Lors du chargement de la page, une multiplication est proposée en tirant deux nombres au hasard compris entre 1 et 10.
- L'utilisateur doit alors proposer un résultat en saisissant un nombre dans le champ adéquat. Lorsqu'il clique sur le bouton, le programme vérifie sa proposition et affiche le message adéquat dans l'élément prévu à cet effet.
- Le bouton change également de label et passe de vérifier à continuer. Lors d'un nouveau clic sur ce bouton, le message disparaît et une nouvelle multiplication est proposée.
- Dans le même temps, le label du bouton change à nouveau, et passe de continuer à vérifier.

Remarquez que pour réaliser ce programme, il est nécessaire que la fonction **valider** (appelée lors du clic sur le bouton) fasse alternativement deux choses différentes.

Cette alternance peut être implémentée à l'aide d'une variable globale qui prend successivement et alternativement deux valeurs distinctes, permettant à chaque appel de la fonction de savoir quelle est la partie à exécuter...

**Indication :**

1. Les fonctions utiles pour réaliser cet exercice : alert, getElementById, isNaN, random, floor.
2. Les attributs utiles à manipuler pour cet exercice : value, innerHTML, style.visibility, style.color

**Exercice 4 : Vérification Formulaire**

Le principe de la vérification de formulaire en JavaScript est le suivant :

- l'utilisateur remplit les champs du formulaire et soumet les valeurs saisies en cliquant sur le bouton de type *submit*.
- Lors d'un clic sur ce bouton, l'évènement **onsubmit** est déclenché sur l'élément HTML **FORM** et la fonction JavaScript associée à cet évènement (ici, la fonction **checkform**) est appelée.

- Si cet appel de fonction retourne la valeur **true** alors les données du formulaire sont bien envoyées au script placé dans l'attribut action du formulaire (ici, le script **exo4-action.php**, fourni) sur le serveur.
- Sinon, si l'appel à la fonction retourne la valeur **false**, les données du formulaire ne sont pas envoyées au serveur.

Complétez le fichier **exo4.js** de façon que la page HTML **exo4-formulaire.html** ait le comportement décrit précédemment.

Si les données du formulaire sont incorrectes, la fonction **checkform** doit faire apparaître le message d'erreur adéquat dans l'élément prévu à cet effet (et bien sûr, retourner la valeur **false**).

**Indication :**

1. Les fonctions utiles pour réaliser cet exercice : alert, getElementById, test.
2. Les attributs utiles à manipuler pour cet exercice : value, innerHTML, style.visibility

**Exercice 5 : Deviner un nombre**

On veut réaliser une page HTML qui permette de jouer au jeu "deviner un nombre". Le jeu s'effectue dans la page HTML :

- L'utilisateur saisit ses propositions dans un élément INPUT et le résultat est affiché sur la page elle-même.
- Il est possible de jouer plusieurs parties de suite et d'afficher les statistiques après chaque partie :
  1. Le nombre total de parties jouées.
  2. Le nombre moyen d'essais sur toutes les parties jouées.
  3. Le nombre d'essais de la meilleure partie (de la partie dans laquelle l'utilisateur a deviné le nombre avec le moins d'essais).

Ces statistiques doivent être mise à jour :

- Dans la page HTML elle-même
- Après chaque partie

Après chaque partie, une fenêtre de confirmation (affichée par la fonction confirm) demande à l'utilisateur s'il veut jouer une nouvelle partie.

**Indication :**

1. Les fonctions utiles pour réaliser cet exercice : alert, confirm, getElementById, isNaN, toFixed, Number, random, floor.
2. Les attributs utiles à manipuler pour cet exercice : value, innerHTML, style.visibility, style.color