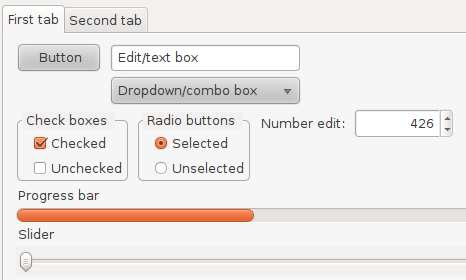
**TD : Interaction Homme Machine sur le web**

Lors de la navigation sur le Web, les internautes interagissent avec leur machine par le biais des pages Web. L’Interface Homme-Machine (IHM) repose sur la gestion d’événements associés à des éléments graphiques (textes, images, boutons ...) munis de méthodes algorithmiques.

a. Les évènements

Un évènement en informatique est une action reconnue et identifiée par un programme.

Les évènements peuvent être déclenchés par un utilisateur ou par le système lui-même.

Un grand nombre d’évènements sont les interactions avec les utilisateurs.

On peut par exemple citer :

• les interruptions clavier,

• les différentes actions possibles de la souris (survol, clic gauche,clic droit, ... )

• les différentes actions avec les doigts pour une interface tactile (tap, pinch, swipe).

On utilise les évènements en programmation dite « événementielle » pour construire des interfaces graphiques (en anglais GUI graphical user interface).

Jusqu’ici nous avons fait de la programmation séquentielle, le programme exécute les commandes dans l’ordre où elles apparaissent comme une séquence (à la suite l’une de l’autre). Ce n’est pas

le cas en programmation événementielle, l’utilisateur ou un évènement extérieur va provoquer le déclenchement d’une action via une fonction.

b. Interface web

Dans une page web on est très souvent amené à interagir avec l’utilisateur. La programmation événementielle y est particulièrement adaptée.

Les réactions à prévoir suite aux événements déclenchés par l’utilisateur peuvent être faites du « côté client » ou côté serveur.

**Exercice 1 Premier exemple : Un formulaire traité côté client en javascript.**

1. Considérer le fichier source formulaire.html (normalement récupéré à côté de ce présent document)

2. Ouvrir la page avec un navigateur et le code source avec un éditeur. Quels sont les différents langages présents dans cette page ?

2. En lisant attentivement le code source, on peut y trouver les bonnes réponses. Préciser les lignes ou se trouvent ces informations.

3. Que ce passe t-il lorsqu’on sélectionne les trois bonnes réponses ?

4. Qu’est-ce qui déclenche la fonction maFonction() ?

5. Modifier l’opération et vérifier que le bon résultat est saisi.

6. Ajouter une proposition pour les radiobuttons et le selecteur.

7. On voudrait signaler à l’utilisateur non seulement son score mais aussi là ou il s’est trompé. Comment peut-on faire ?

Modifier la page et le script pour indiquer en rouge une éventuelle erreur dans l’addition.

**Exercice 2 spécialités de première**

En s’inspirant de l’interface de l’exercice 1, réaliser une interface graphique web ou un élève de première pourra saisir :

son nom, son prénom , sa classe dans un sélecteur, ses trois spécialités parmi les dix du lycée (checkbox) et enfin choisir celle qu’il veut abandonner en terminale (radio button).

c. Listeners ou écouteurs

Un listener est un moyen d’associer à un élément de l’interface une fonction qui se sera appelée si un événement d’un certain type est déclenché.

C’est comme si on ajoutait à l’élément la possibilité d’écouter si l’événement est déclenché et de réagir en conséquence.

En javascript, on utilisera addEventListener dont une doc se trouve à :

https://developer.mozilla.org/fr/docs/Web/API/EventTarget/addEventListener

Voici un exemple basique :

Le code HTML

<ul id="outside ">

< li id="item1"> premier choix</li>

< li id="item2">second choix</li>

</ul>

En JavaScript on place écouter sur l’élément identifié par “outside”.

// ajout d’un écouteur d’evenement

var element = document.getElementById("outside");

element.addEventListener("click " , modifyText , false );

Lors d’un clic sur un élément de la liste “outside”, l’écouteur déclenchera la fonction modifyText, qu’il reste à écrire.

Réaliser l’ensemble en écrivant une fonction modifyText qui change la couleur de l’item de la liste cliquée.