



## INTRODUCTION

Cet exercice consiste à créer un formulaire pour tester l'ordre dans lequel les événements sont transmis au système. Pour réaliser cet exercice, vous devrez reproduire le plus fidèlement possible le formulaire de la dernière page et créer le code JavaScript nécessaire pour afficher correctement les événements. La présentation du formulaire doit être assurée par une feuille de style CSS.

## COMPÉTENCES NÉCESSAIRES

- Connaître les bases du langage JavaScript;
- Connaître les attributs HTML pour les événements;
- Connaître les balises HTML pour la création de formulaires et de contrôles.

## COMPÉTENCES VISÉES

- Après cet exercice, vous serez capable de:
  - Comprendre l'ordre d'interprétation des événements;
  - Respecter les normes de codage imposé dans le cours pour le code JavaScript.

## SOURCES

- Le fichier compressé « 09.10\_Exercice\_06\_evenements\_html\_js.zip », disponible sur LÉA, comprennent les fichiers suivants :
  - Le présent document;
  - Une capture d'écran du formulaire " 09.11\_Exercice\_06\_CaptureEcran.gif";
  - Le PowerPoint du cours "09.20\_PPT\_JavaScript\_Evenements.pdf".

## PRODUCTION

1. Créez un nouveau document nommé « evenement\_html\_js.html ».
2. Conseil : Créez les éléments visuels du document et assurez la présentation CSS avant de commencer à créer le code JavaScript.
3. Note : Les éléments de base du document affiché ne sont pas générés avec du code JavaScript, ils sont codés directement dans la partie <body> du document (n'oubliez pas de respecter les normes de codage XHTML du cours). Le JavaScript sert uniquement à la gestion des événements.
4. Le document est composé d'une zone de texte (textarea) pour afficher les événements et d'un contrôle texte (input) qui servira de source pour la capture des événements. C'est donc sur ce dernier que vous devez ajouter les attributs pour capturer les événements.



- Créer une fonction « Affiche » qui recevra en paramètre le texte à afficher. Ce texte sera fourni par le contrôle de texte lorsque vous provoquerez des événements sur ce dernier à l'aide de la souris ou du clavier.
- Contraintes pour l'affichage dans la zone de texte. Lorsque vous avancerez dans le codage, vous allez vous rendre compte que certains événements provoquent plusieurs inscriptions, par exemple le survol du contrôle. Cet effet est indésirable et vous devez trouver un moyen d'éviter que le nom d'un événement ne soit affiché sur plusieurs lignes consécutives. De plus, pour ajouter à la lisibilité, vous allez insérer une ligne séparatrice après un groupe d'événement. Pour déterminer si un événement est seul ou fait partie d'un groupe, nous utiliserons la règle suivante. Un séparateur est inséré si l'affichage de l'événement précédent s'est fait il y a plus d'une demi-seconde. Enfin, l'ajout d'un séparateur provoque une nouvelle situation que vous devez gérer efficacement. Par exemple, si vous faites un déplacement de la souris au dessus du contrôle texte, qu'un délai d'une demi-seconde s'écoule et que vous déplacez à nouveau la souris, l'affichage devra afficher qu'il y a eu un déplacement de souris, une ligne séparatrice et une autre inscription qu'il y a eu un déplacement de souris. Si ce n'est pas clair pour vous, étudiez la capture d'écran fournie.

### Saisies d'écrans

La couleur d'arrière-plan de la page du navigateur est #4D6185

Les bordures des contrôles ont une largeur de 1px, une couleur #999999 et un effet de sillon.

fieldset

Ce contrôle affiche 25 lignes

Tout le texte est de type « Verdana », de taille 14px et de couleur #FFFFFF sauf indication contraire.

Les contrôles ont une largeur de 190px. Le texte affiché dans ceux-ci a la couleur #7C9FD3 et une taille de 12 px.

### REMISE

- Cet exercice terminé est à présenter à l'enseignant en classe (Date de remise sur LÉA).
- Votre code source doit être parfaitement présenté (règles imposées dans le cours).
- Votre résultat final doit correspondre **exactement** au résultat présenté ci-dessus.