Université Grenoble Alpes - IUT2 - Département informatique
Module 4103C

TP 3 : JavaScript, Evénements & DOM - Partie 2 (2H)

Objectifs du TP :

- Développer sa connaissance des événements pouvant potentiellement survenir sur des éléments HTML
- Savoir gérer les cookies en JavaScript
- Savoir utiliser l'objet Regexp
- Savoir gérer le Drag&Drop

Ce document est consultable sur l'intranet du département.

Les pages suivantes : http://www.w3schools.com/js/js_events.asp et http://www.w3schools.com/jsref/dom_obj_event.asp peuvent vous être utiles dans cet apprentissage.

1. Sauvegarde des fonctions ajoutées avec des cookies

Dans le TP précédent nous avons donné à l'utilisateur la possibilité d'éditer les boutons de la calculatrice. Le problème est que si l'utilisateur recharge la page, les boutons édités ont repris leur état initial et tout est à refaire. Il faut donc trouver un moyen de sauvegarder ces informations. Avec JavaScript dans le contexte d'une application Web, il est impossible d'écrire des fichiers chez le client.

Une solution (*que l'on verra plus tard*) consiste à transmettre les données à sauvegardées au serveur Web (par des appels AJAX).

Une autre solution, utilisable lorsque les données ne sont pas trop volumineuses, consiste à utiliser des **cookies**.

C'est cette solution que nous nous proposons d'expérimenter ici.

Pour pouvoir utiliser les cookies, vous devrez travailler dans votre **`public_html`** et accéder à votre calculatrice via l'URL avec votre **`~login`**. Pour cela, commencez par :

1. Créer un répertoire `calculatrice` dans votre `public_html`
2. Placez-y vos fichiers `html, css et js` correspondant à la réalisation du TP précédent (La calculatrice)
3. Faites un `setup-public-html`
4. Vérifiez que votre application calculatrice est accessible à l'adresse `http://www-etu-info.iut2.upmf-grenoble.fr/~`**`login`**`/calculatrice/calculatrice.html` et que celle-ci fonctionne

Une fois votre calculatrice fonctionnelle via votre page perso, suivez les instructions suivantes :

1. La sauvegarde sera réalisée au moyen d'une fonction `save` prenant en paramètre le bouton édité. Nous choisissons de déclencher la sauvegarde de la chaîne de caractères saisie dans le champ texte dans deux cas :
   - Lorsque l'utilisateur double-clique sur le champ texte pour le retransformer en bouton (il suffira dans ce cas d'appeler `save` dans la fonction `fix`)
   - Lorsque le champ texte perd le focus (on peut considérer que la saisie est terminée dans ce cas). Cet événement correspond à l'évènement onblur. Dans la fonction `edit`, associez l'appel de `save(this)` à l'événement `onblur` de l'input text d'édition. Cette association doit être réalisée par programmation. Supprimer la gestion de l'événement `onblur` dans `mode_calcul` et `fix` (avec RemoveAttribute)
2. En vous appuyant sur http://www.w3schools.com/js/js_cookies.asp (copier-coller les fonctions `setcookie` et `getCookie`), écrivez le code de `save` pour ajouter un cookie dont le nom correspond à l'identité de l'élément édité (libre1 ou libre2 ou ... ou libre6) et dont la valeur correspond au contenu de l'élément. Pour vérifier que les cookies ont bien été déposés cliquez sur l'onglet "Stockage" du web developper tool.
3. En vous appuyant sur le même document, dans la fonction `init` définie lors du précédent TP récupérer la valeur des cookies et

initialisez les boutons éditables avec ces valeurs

2. Contrôle de la validité d'une chaîne de caractères avec RegExp

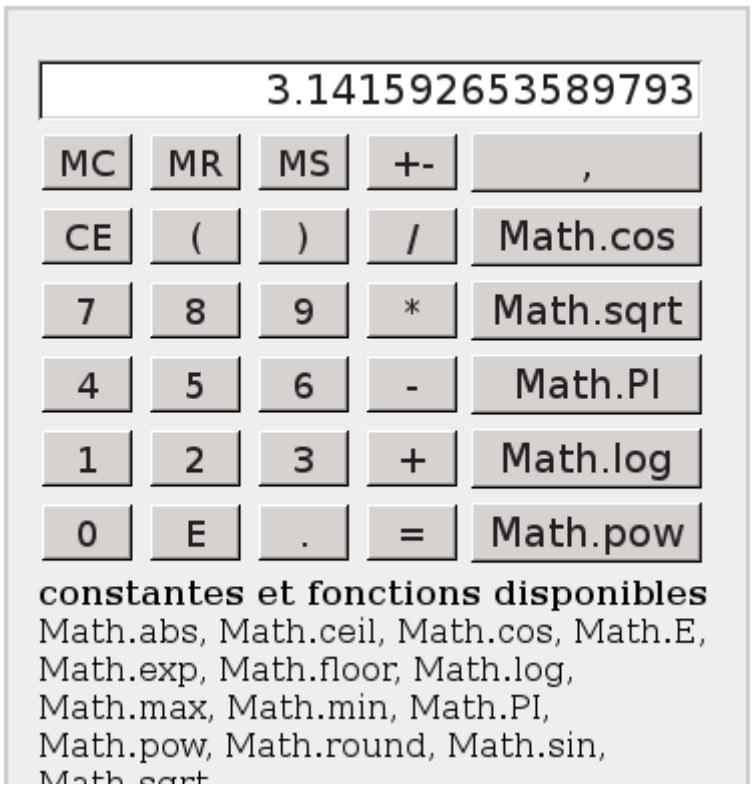
Afin d'ajouter un peu de contrôle sur ce que fait l'utilisateur, on souhaite maintenant vérifier avant de ranger une valeur en mémoire (lorsque l'utilisateur clique sur le bouton `MS`) que le contenu de la zone d'affichage à stocker **représente bien un nombre**.

Pour cela, vous allez vérifier via une Expression Régulière (RegExp) que la chaîne de caractères soit bien composée de chiffres et éventuellement d'un caractère '-' au début et d'un caractère '.' à l'intérieur.

En considérant le pattern suivant pour un nombre valide : `/^-?\d+\.?\d*$/` et en vous appuyant sur MDN - RegExp (Fr) et w3schools.com - JavaScript RegExp Reference, modifiez la fonction `range_memory` définie lors du précédent TP pour que la sauvegarde dans la variable globale memory ne se fasse qu'à la condition que le pattern soit respecté.
Un message d'erreur sera affiché (alert) si ce n'est pas le cas.

3. Edition des boutons par Glisser-Déposer (Drag&Drop)

Au moment de l'édition, l'utilisateur ne connaît pas nécessairement les fonctions disponibles. C'est pour cette raison que nous proposons d'afficher une liste de fonctions disponibles dans la partie inférieure de la calculatrice (voir l'image ci-contre et le code html correspondant ci-dessous).



De plus, le fait d'écrire en toutes lettres les fonctions

est un peu fastidieux.
Nous proposons d'éditer les boutons éditables par glisser-déposer des noms de fonctions sur les boutons.

En vous basant sur l'exemple suivant w3school.com - Demonstration of drag and drop events (Try it Yourself) (que vous devez analyser et comprendre), faites en sorte que le glisser-déposer d'un nom de fonction sur un bouton déclenche la modification de l'attribut `value` du bouton éditable (on pourra se limiter à la gestion des événements `dragstart`, `dragend`, `dragover` et `drop`).
Attention, seul le contenu des **boutons éditables** doit pouvoir être modifié de cette façon.
Enfin, pensez à déposer un cookie sur l'événement `drop` (appel à `save`)

```
<strong>Constantes et fonctions disponibles :</strong>
<span draggable="true" id="Math.abs"> Math.abs </span>,
<span draggable="true" id="Math.ceil"> Math.ceil </span>,
<span draggable="true" id="Math.cos"> Math.cos </span>,
<span draggable="true" id="Math.E"> Math.E </span>,
<span draggable="true" id="Math.exp"> Math.exp </span>,
<span draggable="true" id="Math.floor"> Math.floor </span>,
<span draggable="true" id="Math.log"> Math.log </span>,
<span draggable="true" id="Math.max"> Math.max </span>,
<span draggable="true" id="Math.min"> Math.min </span>,
<span draggable="true" id="Math.PI"> Math.PI </span>,
<span draggable="true" id="Math.pow"> Math.pow </span>,
<span draggable="true" id="Math.round"> Math.round </span>,
<span draggable="true" id="Math.sin"> Math.sin </span>,
<span draggable="true" id="Math.sqrt"> Math.sqrt </span>
```

4. S'il vous reste du temps...

Nous n'avons pu passer en revue qu'un faible nombre d'événements au regard de ce qui existe.

Nous vous invitons à tester la majorité des exemples donnés sur la page http://www.w3schools.com/jsref/dom_obj_event.asp. Regardez aussi du côté du MouseEvent Object afin de prendre connaissance des informations qu'il est possible de récupérer lors d'un événement souris.