

## TP n° 01 – JavaScript

### Module 2105

Ce TP va être l'occasion de s'initier à la programmation JavaScript coté « client » (sur le navigateur).

## 1 Création dynamique de balises HTML et événements :

Soit la page WEB très sommaire suivante :

```
...
<body>
  <button id="ok">OK</button>
  <span id="msg">Messages</span>
  <div class="contenu"></div>
</body>
</html>
```

1. Ajouter la balise `script` chargeant le fichier vide `tp1-1.js`, en utilisant l'attribut `src`.

2. Editer le fichier JavaScript en ajoutant un message de trace dans la console :

```
console.log("Début du programme JS...");
```

Charger ensuite le fichier HTML dans un navigateur et ouvrir la console du développeur WEB (Ctrl+Shift+K avec Firefox) pour faire apparaître le message.

3. On veut faire apparaître une alerte quand on clique sur le bouton. Ajouter alors l'attribut `onclick` à la balise `button` et ajouter le code JavaScript directement dedans :

```
alert('Vous avez cliqué sur le bouton !');
```

Tester ensuite le fonctionnement du bouton.

4. Il est cependant préférable d'utiliser une fonction JavaScript plutôt que d'écrire le code directement dans le HTML pour l'évolutivité et la lisibilité du code. Créer donc la fonction `click1()` dans le fichier `tp1-14.js`, qui va permettre de changer la couleur de fond de la balise `span`. On utilisera pour cela la méthode `getElementById()` sur la variable `document`, pour obtenir l'objet représentant le `span`, puis ensuite, on lui appliquera la méthode `setAttribute()` pour changer la valeur de la classe CSS. Utiliser les 2 classes `ok0` et `ok1` alternativement.
5. Afin de modifier le contenu HTML d'une balise, on peut se servir de la propriété `innerHTML` d'un élément DOM (Document Object Model). Ajouter un second bouton (« Message ») et la fonction `click2()`, pour afficher la valeur du code HTML actuel de la balise `span` dans les traces de la console, puis ajouter à cette valeur, le code HTML suivant :

```
<br/>Ajout du message numéro x
```

On utilisera une variable pour que `x` corresponde au nombre de clics sur le bouton.

6. Ajouter enfin un 3<sup>ème</sup> bouton (« Effacer ») pour supprimer tous les messages existants dans la balise `span` (fixer à "" la valeur). Mais cette fois-ci, on n'utilisera pas l'attribut `onclick`, mais plutôt la méthode `addEventListener()` pour enregistrer l'association du clic à une méthode :

```
...
btn.addEventListener('click',click3);
```

7. On veut maintenant ajouter des balises `canvas` dans la première balise possédant la classe `contenu`. Créer la fonction `addCanvas()` - que l'on appellera à la fin de `click1()` - et utiliser la méthode `querySelector()` de l'objet `document`, pour sélectionner la cible désirée comme en CSS. Créer ensuite avec la méthode `createElement()` de l'objet `document`, la balise `canvas` et ajouter celle-ci à la fin du « contenu ». Fixer enfin la couleur de fond du `canvas` avec la couleur complémentaire de celle du `span` (Si le `span` est rose, le dessin sera vert et vice versa - voir § 1.4).
8. Comment afficher le nombre de dessins du « contenu » sur le bouton OK en utilisant la propriété `childNodes` qui renvoi la liste des enfants d'un élément DOM (voir `NodeList`) ?
9. Comment effacer aussi le premier dessin lors du clic sur le bouton « Effacer » ? Utiliser la méthode `removeChild()` de tout élément DOM et la méthode `item(0)` pour récupérer l'élément 0 d'une liste d'éléments. On pourra ajouter un identifiant unique au `canvas` (attribut `id`) et visualiser le code HTML dans l'inspecteur WEB du navigateur pour vérifier que l'on supprime bien le premier élément du « contenu ». Attention à bien mettre à jour le nombre du bouton OK.
10. Finir le TD n° 01.