

TP Javascript

Pour ce TP, deux activités vous sont proposées ci-dessous, et pour vous aider vous aurez sur la deuxième feuille un résumé des codes HTML / Js les plus utiles.

Activité 1

1. Récupérer le dossier TPJs du TP sur studserver.llb.school - réseaux
2. Télécharger puis dézipper le dossier dans votre répertoire de travail sur votre ordinateur.
3. Ouvrez le fichier **activité11.html** dans le navigateur de votre choix, puis avec un éditeur de texte (type geany ou notepad++) afficher son code source
4. Ouvrez aussi sur geany ou notepad++ le code du fichier javascript **activite-1.js**.
5. Réaliser les 3 missions ci-dessous - **3 pages HTML différentes** - et un unique fichier javascript afin d'implémenter les pages web dynamiques demandées.

Vous trouverez de l'aide sur la feuille jointe à cet énoncé, vous aurez aussi des indices donnés dans le menu de votre TP.

Vous trouverez aussi les corrections de chacune des missions dans le menu, mais vous ne devez regarder la correction que pour vérifier votre travail. Pensez à appeler votre professeur pour valider votre travail mais aussi pour vous aider si tous les indices fournis ne vous permettent pas de réaliser votre mission.

- **Mission 1 : Boite de dialogue**

Vous aurez à compléter uniquement le code source du fichier Javascript **activite-1.js** pour cette mission mais vous aurez besoin du code source du fichier **activité11.html**.

Puis effectuer les 2 étapes demandées.

Aucune modification à faire sur le fichier **activite11.html**

- **Mission 2 : Survol de la souris – modifier la taille**

Effectuer de même en ouvrant **activité12.html** avec un éditeur de texte (type geany ou notepad++) afficher son code source ainsi que celui du même fichier javascript **activite-1.js**.

Puis effectuer le travail demandé.

Aucune modification à faire sur le fichier **activite12.html**

- **Mission 3 : Couleur de fond de la page**

Effectuer de même en ouvrant **activité13.html** avec un éditeur de texte (type geany ou notepad++) afficher son code source ainsi que celui du même fichier javascript **activite-1.js**.

Puis effectuer les 2 étapes demandées.

Attention, il vous faudra **modifier les deux fichiers : activite-1.js et activite13.htm**.

Activité 2 - : La synthèse RGB

Un professeur veut créer une page web qui illustre la synthèse additive des couleurs. Pour cela, il voudrait changer la couleur du fond d'écran de la page en agissant sur trois glissières. Chacune doit modifier l'intensité d'une couleur (rouge, vert et bleu) de 0 à 255.

Il n'arrive pas à venir à bout de son travail, pouvez-vous l'aider à le terminer ?

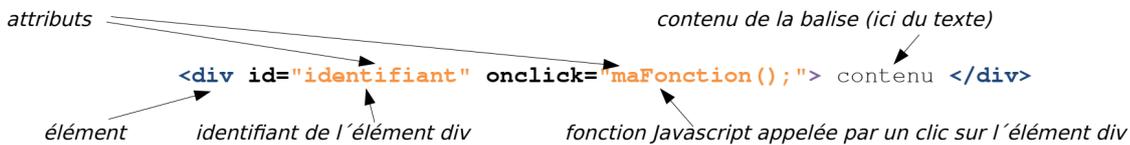
Il vous a laissé les fichiers non terminés **activite2.html** et **activite2.js** dans le dossier.

Pour une explication de la synthèse additive des couleurs :

<http://webphysique.fr/synthese-additive/>

Rappels utiles sur HTML/Javascript :

- **Fonctionnement de chaque attribut d'une balise** d'un élément HTML :



- **Accès en Javascript** à un élément HTML : `getElementById()`

Par exemple, dans la variable x on récupère l'élément HTML dont l'Id est "identifiant" :

```
let x =document.getElementById("identifiant");
```

- **Modifier le style** d'un élément HTML :

Par exemple, pour modifier la **couleur du texte** de l'élément x récupéré précédemment :

```
x.style.color="#0000FF";//modifie la couleur du texte en bleu (codage hexadécimal)
```

```
x.style.color="rgb(0,0,255)"; // décimal
```

Par exemple, pour modifier la **couleur de fond** de l'élément x récupéré précédemment :

```
x.style.background="#FFFF00";// modifie la couleur du fond en jaune
```

Par exemple, **change la largeur** de l'élément à 100 pixels :

```
x.style.width=100+"px";// fixe la largeur de l'élément à 100 pixels
```

- Modifie le contenu (le texte) de l'élément HTML dont l'id est x :

```
x.innerHTML="nouveau contenu";
```

- **Quelques évènements HTML** utiles sur les éléments :

`onclick` <!-- clic sur l'élément-->

`onmouseover` <!-- survol de l'élément par le pointeur de la souris -->

`onmouseout` <!-- le pointeur de la souris sort de l'élément -->

`oninput` <!-- se produit immédiatement après que la valeur d'un élément a changé -->

- Exemple d'utilisation de `oninput` :

On utilise ce dernier, par exemple lors de la création d'une glissière `type="range"` dans le fichier HTML, comme ci-dessous :

```
<input id="entree" type="range" oninput="maFonction();" min="0" max="100">
```

La fonction javascript `maFonction()` est appelée lorsque l'utilisateur déplace le curseur, c'est-à-dire dès que la valeur est modifiée (`oninput`). Dans le code javascript, on accède ensuite à la valeur de la position du curseur (ici entre 0 et 100) en écrivant :

```
let x=document.getElementById("entree")
```

```
x.value //attention : cette valeur est de type string.
```

- **D'autres fonctions et méthodes javascript** utiles :

```
prompt("saisie de texte"); //Ouvre une boîte de dialogue avec une zone de saisie utilisateur.
```

```
typeof(a); //affiche type de la variable a
```

```
Math.random(); //Renvoie un flottant pseudo-aléatoire compris dans l'intervalle [0,1[
```

```
chaine.length //longueur de la chaîne chaine
```

```
console.log("texte") // affiche un texte dans la console , très pratique pour débiter
```