

TP JavaScript – Initiation

Gestion des événements souris et clavier, Accès au DOM.

Pré-requis : Le cours JavaScript

Activité 0 :

Suivez les tutoriels et les exercices :

http://10.69.88.1/opale/2-%20Premiers%20Pas%20JavaScript/co/0-module_Javascript.html

http://10.69.88.1/opale/3-%20JavaScript%20-%20Les%20fonctions/co/0-module_Javascript-Fonctions.html

(Accès extérieur :

http://lycvauv.ydns.eu/accueil/opale/2-%20Premiers%20Pas%20JavaScript/co/0-module_Javascript.html

http://lycvauv.ydns.eu/accueil/opale/3-%20JavaScript%20-%20Les%20fonctions/co/0-module_Javascript-Fonctions.html)

Activité 1 :

On vous donne les fichiers d'un mini site web :

activite1.html La page HTML

activite1.css Le CSS associé

activite1.js Le code JavaScript que vous allez modifier.

et un dossier «images » avec les images à afficher.

TRAVAIL 1 : Etude de l'existant

a) **Copier** le site dans votre espace personnel et testez-le. Pour l'instant, le JavaScript n'est pas actif.

b) **Associez** le JavaScript au document HTML en ajoutant à la fin :

```
<script type="text/javascript" src="activite1.js"></script>
```

Pourquoi à la fin ?

Parce que le script utilise des éléments du DOM qui doivent exister au moment de son exécution.

Si vous avez fait correctement le travail, le titre doit apparaître en haut de l'écran.

c) **Observation** à propos du CSS :

Vous noterez qu'on utilise des fonctionnalités récentes du CSS :

- le positionnement des <div> se fait par la méthode *display :flex* (et non avec *position :absolute* que l'on a déjà étudié).
- On utilise *:hover* pour modifier le CSS quand la souris passe sur l'élément.
- On utilise les codes couleurs *rgba*. (Rouge, Vert, Bleu, Transparence).

d) **Observation** à propos du HTML :

Pour le html, on utilise les balises <figure> et <figcaption> pour les images avec légende.

TRAVAIL 2 : Analysez le code JavaScript de la fonction qui permet d'afficher le titre :

- Modifiez le texte du titre
- Quel est la classe CSS qui gère l'apparence du titre ? Modifiez le CSS pour ajouter au titre le réglage : `text-shadow: 6px 6px 6px black;`

TRAVAIL 3 :

- Observez :** Dans le code JavaScript, ligne 5, la variable « mesImages » est un tableau contenant des noms de fichiers images. Ces images sont fournies avec le TP. Vous devez les avoir à disposition dans un sous dossier appelé « images »
- En vous inspirant des « fonctions » déjà créées, vous allez créer une FONCTION CLASSIQUE vide nommée « creerBoutons () » qui ne prendra aucun argument (les parenthèses resteront vides). **Les explications pour y arriver** sont les suivantes :

Présentation du travail de la fonction (à lire) :

Cette fonction va directement modifier le HTML :

Elle va créer des balises <a> (lien hypertexte) dans la zone HTML repérée par l'attribut « id = 'cible' ».

Les balises <a> sont réglées par le CSS pour avoir un style « bouton »

Il y aura 1 lien pour chaque nom contenu dans le tableau des images.

Chaque lien est associé à un événement « click » de la souris qui exécute la fonction « afficherImage »

Quand on appuiera sur un lien, la photo correspondante s'affichera.

Voici un exemple d'utilisation de *createElement* et *appendChild* pour créer une balise :

Exemple : Modifier une page HTML en ajoutant une balise <p> à une balise <div> :

Situation initiale :

```
<div id="div1">
<p id="p1"> Ceci est un paragraphe </p>
<p id="p2"> et là un nouveau paragraphe </p>
</div>
```

Exécution du Javascript suivant :

```
var para = document.createElement("p"); // Créé une balise <p>
para.innerHTML = "Je suis la nouvelle balise"; // Ecrit dans la zone texte

var cible = document.querySelector("#div1"); // Recherche de la cible
cible.appendChild(para); // Ajoute la balise <p> à l'endroit voulu dans la page html
```

Situation après l'exécution du script :

```
<div id="div1">
<p id="p1"> Ceci est un paragraphe </p>
<p id="p2"> et là un nouveau paragraphe </p>
<p> Je suis la nouvelle balise</p>
</div>
```

Comment appliquer au contenu de la fonction que vous devez créer :

1. Recherche de la <div> qui recevra les boutons :

```
var cible = document.querySelector("#cible");
```

2. Ecrire cette boucle FOR qui s'exécutera autant de fois qu'il y a de nom dans le tableau « mesImages » :

```
for ( let i = 0; i < mesImages.length; i++)  
{  
    //  
    // Vous allez bientôt ajouter du code ici  
    //  
}
```

NB : La variable « i » dans la boucle sera bientôt utilisée comme un **index** pour accéder à chaque élément du tableau.

3. Code de la boucle FOR, à traduire en langage JavaScript :
 - Avec la méthode « createElement », créez un élément « lien hypertexte » (balise <a>)
 - Modifier sa *propriété* « innerHTML » : elle doit contenir le nom de l'image correspondant à l'index dans le tableau, soit `mesImages[i]`
 - Avec la *méthode* « addEventListener », associez à l'élément un événement « click » de souris, avec comme action, la fonction « afficherImage »

NB : Vous avez un exemple d'utilisation de « addEventListener » dans le TP JS0)

 - Avec la méthode « appendChild », ajoutez l'élément créé à la cible.
4. Votre fonction est prête mais ... encore faut-il l'exécuter, car une fonction ne s'exécute que si on « l'appelle ». C'est ce qu'on fait à la ligne 8 pour « appeler » la fonction `afficher_titre()`.
Ajoutez à la ligne 9 l'appel à votre fonction.

TRAVAIL 4 :

Certaines photos du dossier « images » ne sont pas affichées. Corrigez cela !

Activité 2 : Pour consolider des acquis ...

TRAVAIL 1 :

Créez une fonction « `creer_automatic()` » qui ajoutera dans la partie droite de l'interface 2 liens :

- Lien « Suivant » qui exécute la fonction « `suiwant()` »
- Lien « Précédent » qui exécute la fonction « `precedent()` »

Ces 2 fonctions existent déjà. Elles modifient la variable globale « `i` » et appellent la fonction « `automatic()` »

Testez : Les boutons créés permettent de faire défiler les photos.

TRAVAIL 2 :

Modifiez le code de la fonction « `automatic()` » pour gérer les dépassements :

Suivant : Quand on arrive à la dernière diapo, on revient à la première.

Précédent : Quand on arrive à la première diapo, on passe à la dernière.

Activité 3 :

Modifiez la fonction « `creer_automatic()` » qui ajoutera dans la partie droite de l'interface 2 autres liens :

- lien « Départ »
- lien « Arrêt »

Créez le code : Le bouton « Départ » exécute une fonction qui utilise la méthode JavaScript « `setInterval` »

Objectif : lancer le diaporama en automatique, à raison d'une diapo par seconde.

Site pour la fonction JavaScript `setInterval()` :

voir https://www.w3schools.com/jsref/met_win_setinterval.asp

Prévoir le code pour le bouton « arrêter ».

Pour les curieux : le CSS3 apporte les « animations » aux objets. C'est une partie de ce qu'on appelle le WebKit. Au début, chaque navigateur avait son propre jeu d'instruction Webkit. A présent, les navigateurs « à jour » comprennent ces instructions évoluées sans qu'il y ait besoin de préciser qu'elles appartiennent au Webkit.

Exemple de tourniquet de photos utilisant les animations :

<http://ambrosialaure.free.fr/Animation%20css3%20%20%20tourniquet.htm>