

TP1 : prise en main de JavaScript et jQuery

Exercice 1 : exécution d'un programme JavaScript

1. Télécharger le fichier *index.html* depuis l'espace cours contenant le squelette HTML5 minimal et enregistrez-le dans un dossier *exo1/*.
2. Créez un sous-dossier *js/* dans *exo1/*.
3. Dans Notepad++, créer un nouveau fichier contenant la ligne de code `alert ("Hello world!") ;` et enregistrez-le dans le dossier *js/* sous le nom *script.js*
4. Ajouter au fichier *index.html* le code nécessaire pour exécuter le fichier JavaScript créé et testez en ouvrant *index.html* dans un navigateur.
5. Rajouter la ligne de code `var nom = prompt ("Entrez votre nom :") ;` dans le fichier JavaScript et actualiser *index.html* dans le navigateur.
6. Ouvrez la console du navigateur pour visualiser le message d'erreur renvoyé par le programme et repérez où se trouve l'information indiquant la ligne du programme où une erreur a été rencontrée. Corrigez cette erreur et re-testez votre code
7. Ajoutez une ligne de code pour afficher dans la console le message "Bonjour" suivi du nom entré par l'utilisateur.

Exercice 2 : import et utilisation d'une bibliothèque externe

1. Créez un dossier *exo2/* et placez-y le fichier *index.html* minimal, ainsi qu'un sous-dossier *js/* contenant un fichier JavaScript *script.js*.
2. En vous aidant du début du document ressource *p5js_intro.pdf* importez la librairie JavaScript p5 pour pouvoir l'utiliser dans votre programme *script.js*
3. Pour tester, dans *script.js* placez le code suivant :

```
function setup() {
    createCanvas(800, 800);
}

function draw() {
    background('black');
    fill('red');
    stroke('green');
    strokeWeight(5);
    ellipse(200, 200, 60, 60);
}
```
4. Changer les valeurs des paramètres des fonctions `background()`, `fill()`, `stroke()`, `strokeWeight()` et `ellipse()` pour voir à quoi ils correspondent (vous pouvez rajouter d'autres types de formes en vous aidant du document *p5js_intro.pdf*).
5. Remplacez la ligne `ellipse(200, 200, 60, 60);` par `ellipse(mouseX, mouseY, 60, 60);` et observez l'interaction créée avec la souris.

Exercice 3 : import de jQuery et sélection d'éléments

1. Créez un dossier *exo3_4* / et placez-y le contenu de l'archive *siteWeb_TP1.zip*.
2. A la racine du répertoire du site, créer un sous-dossier *js*/ contenant un fichier JavaScript *script.js*.
3. Ajouter le nécessaire aux fichiers pour pouvoir utiliser jQuery
4. Avec jQuery, sélectionner l'élément de type image de la partie navigation du site et récupérer cet élément dans une variable nommée *selection*. Afficher cette variable dans la console.

Remarque : vous récupérez dans la console l'affichage d'un objet (Object) représentant la sélection des éléments par jQuery. Vous pouvez visualiser l'intérieur de l'objet en le déployant grâce au petit triangle à gauche de Object. Notamment, sur les premières lignes vous pouvez visualiser le ou les éléments sélectionnés, ce qui vous permet de vérifier le bon fonctionnement de votre sélection.

5. Récupérer dans la variable *selection* l'élément image de la dernière division de la partie d'identifiant *reserver* et afficher cette variable dans la console.
6. Récupérer dans la variable *selection* le titre de niveau 2 de la 3ème division de la partie *reserver* et afficher cette variable dans la console (on vérifiera la propriété *innerText* de l'élément récupérer dans l'objet affiché).
7. Récupérer dans la variable *selection* le dernier paragraphe du premier article de la section *programmation* et afficher cette variable dans la console (vérifier la propriété *innerText*).
8. Récupérer dans la variable *selection* les éléments de classe 'encadrer' et afficher cette variable dans la console.
9. Récupérer dans la variable *selection* les éléments enfants de la division de la partie *programmation* et afficher cette variable dans la console.
10. Récupérer dans la variable *selection* les éléments de classe 'encadrer' qui sont enfants directs de la partie *reserver* et afficher cette variable dans la console.
11. Récupérer dans la variable *selection* l'élément parent de l'élément dont l'attribut *src* vaut 'videos/videoHeader.mp4' et afficher cette variable dans la console.

Exercice 4 : prise en main de la gestion d'événements avec jQuery

1. Sur l'élément *header*, attacher un écouteur d'événement "**click de souris**" déclenchant une fonction nommée *afficherPopupClickHeader* qui affichera une popup d'alerte avec le message "Vous avez cliqué sur le header !".
2. Sur l'élément *footer*, attacher un écouteur d'événement "**click de souris**" déclenchant **une seule fois** une fonction nommée *afficherPopupClickFooterUneFois* qui affichera une popup d'alerte avec le message "Vous avez cliqué sur le footer !".
3. Sur l'ensemble des éléments de classe *encadrer*, enfants directs de la partie d'identifiant *reserver*, attacher 2 événements :
 - o un événement "**entrée de souris**" dans l'élément, déclenchant une fonction *afficherEntreeSourisEncadrer* qui affichera dans la console le message "Entrée" ;

- un événement "**sortie de souris**" de l'élément, déclenchant une fonction `afficherSortieSourisEncadrer` qui affichera dans la console le message "Sortie".
4. Sur l'élément d'identifiant `programmation`, attacher un écouteur d'événement "**déplacement de souris**" déclenchant une fonction `afficherPositionSouris` qui affichera dans la console la position de la souris dans la page, avec un message du type : "Position : (432, 871)".
- 5.
- 5.1. Sur l'objet `document` du DOM (utiliser cet objet à la place du sélecteur CSS de jQuery, sans guillemets), attacher un événement "**touche appuyée**" déclenchant une fonction nommée `afficherToucheConsole` qui affichera dans la console le code de la touche enfoncée (ex : "Code de la touche : 54").
 - 5.2. Tester avec les touches "flèche vers le haut" et "flèche vers le bas". Quel est l'effet par défaut de ces touches sur la page ? Ajouter le nécessaire pour que les effets par défaut des touches n'aient plus lieu.
 - 5.3. En utilisant le code de touche correspondant, ajouter un affichage dans la console indiquant par les valeurs `true` ou `false` si c'est la touche "flèche vers le haut" qui est enfoncée (ex : "Flèche vers le haut : true" si c'est le cas, "Flèche vers le haut : false" sinon).
- Indication : il faut utiliser un opérateur de comparaison dans un test pour obtenir ces valeurs true et false.*
- 5.4. Modifier votre code pour indiquer maintenant si c'est l'une ou l'autre des flèches vers le haut ou vers le bas qui est enfoncée (ex : "Flèche haut/bas : true" si c'est le cas, "Flèche haut/bas : false" sinon)

Exercice 5 : interactions HTML/CSS avec jQuery

1. Mise en place de Javascript et jQuery
 - 1.1. Créez un dossier `exo5` / et placez-y le contenu de l'archive `siteWeb_TP1_Exo5.zip`.
 - 1.2. A la racine du répertoire du site, créer un sous-dossier `js`/ contenant un fichier JavaScript `script.js`.
 - 1.3. Ajouter le nécessaire aux fichiers pour pouvoir utiliser jQuery
2. On voudrait créer une animation sur les divisions (balises `<div>`) de la partie '`reserver`' : au survol par la souris d'une de ces divisions, faire pivoter l'image qu'elle contient sur elle-même de 360° suivant l'axe Y (durée de l'animation : 1 seconde), surélever le cadre contenant le mot "`RESERVER`" et augmenter son ombre (valeur '`4px 4px 10px black`') pour le mettre en valeur, et atténuer les autres divisions avec une légère transparence (durée de l'animation : 1 seconde).
 - 2.1. Sur l'ensemble des divisions de la partie '`reserver`' : attacher un écouteur d'entrée de souris déclenchant une fonction `animDivReserver`.
 - 2.2. Dans la fonction déclenchée `animDivReserver`, appliquer avec jQuery les propriétés CSS pour effectuer les animations souhaitées

Indications :

 - *dans la fonction déclenchée, `\$(this)` représente l'élément qui a capté l'événement*

- la méthode `.find(' sélecteur2 ')` utilisée sur une sélection jQuery , permet de rechercher à l'intérieur des éléments sélectionnés d'autres éléments correspondant à '`sélecteur2`'.
- la propriété CSS `transform` avec la valeur `rotateY(360deg)` permet de faire une rotation autour de l'axe Y
- la propriété CSS `transform` avec la valeur '`perspective(200px) translateZ(40px)`' permet de faire une translation vers l'avant.
- la propriété CSS `transition` permet de faire une transition progressive vers une nouvelle valeur de propriété CSS en lui précisant la propriété CSS concernée et le temps de transition (ex : '`transform 1s`')

3. On voudrait maintenant annuler les effets de l'animation précédente lorsque la souris ressort d'une division de la partie '`reserver`'.
 - 3.1. Sur l'ensemble des divisions de la partie '`reserver`' : attacher un écouteur de sortie de souris déclenchant une fonction `annuleAnimDivReserver`.
 - 3.2. Dans la fonction déclenchée `annuleAnimDivReserver` ajouter avec jQuery les propriétés CSS pour annuler l'animation du survol de souris :
 - Retour à une rotation de 0° en 0 seconde.
 - Retour à une translation en Z de 0px et d'une ombre '`2px 2px 2px black`' pour le cadre contenant le mot "RESERVER"
 - Retour à une opacité de 1 en 1 seconde, pour toutes les divisions de la partie '`reserver`'.
4. On voudrait pouvoir filtrer l'affichage des groupes de la partie programmation en utilisant les boutons qui ont été rajoutés en haut de cette partie et des classes qui ont été rajoutées dans les articles des groupes (voir le code HTML).
 - 4.1. Sur l'ensemble de ces boutons, ajouter un écouteur de click de souris déclenchant une fonction `filtrerGroupe`.
 - 4.2. Dans cette fonction déclenchée `filtrerGroupe`, ajouter le code jQuery permettant d'afficher les groupes correspondant au bouton cliqué, et de masquer les autres (avec des transitions de 1 seconde pour l'affichage et le masquage)

Indications :

 - on utilisera les méthodes `.hide()` et `.show()` pour masquer/afficher les éléments. On peut passer un temps de transition en millisecondes entre les parenthèses de ces méthodes.
 - On pourra utiliser une variable `filtre` qui contiendra le texte du bouton cliqué, avec le comportement suivant :
 - si `filtre` vaut "`TOUS`" : afficher tous les articles
 - sinon, afficher les articles correspondant à la valeur `filtre` et cacher les autres
 - la méthode `.not(' sélecteur2 ')` appliquée à une première sélection permet d'exclure les éléments ciblés par `sélecteur2` de la première sélection.
5. On voudrait que lorsqu'on clique sur une des images de groupe, celle-ci s'affiche en grand (600x600) au centre de l'écran sur un fond noir transparent (laissant voir la page en transparence derrière), avec le nom du groupe affiché au-dessus et la scène et le jour affichés en dessous (sur fond noir). Une fois la grande image et le fond affichés, un click supplémentaire dessus doit permettre de les faire disparaître .

- 5.1. Mettre en place un écouteur d'événement 'click' sur l'ensemble des images de la section "programmation", déclenchant une fonction `afficherGrandeImage`.
- 5.2. Dans cette fonction déclenchée, créer avec jQuery un nouvel élément de type `<div>` et mémorisez-le dans une variable `divFond`.
- 5.3. Sur cet élément `divFond`, appliquer des propriétés CSS pour qu'il soit positionné de façon fixe, à 0 pixel du haut et 0 pixel de la gauche, de hauteur et de largeur 100 %, et avec une couleur de fond noire légèrement transparente (on utilisera un code couleur `rgba()`).
- 5.4. Ajouter cet élément `divFond` au `body` à l'aide de jQuery et vérifier son affichage lors d'un click sur une image.
- 5.5. Toujours dans la fonction déclenchée, créer avec jQuery un nouvel élément de type `<div>` et mémorisez-le dans une variable `divGrandeImage`. Appliquez-lui une couleur de fond noire et une ombre '`0px 0px 80px white`' et ajoutez-la à l'élément de fond `divFond`.
- 5.6. Créer un nouvel élément de type `<h2>` et mémorisez-le dans une variable `h2Nom`.
Ajouter le nom du groupe (récupéré dans l'attribut `title` de l'image cliquée) comme texte à l'élément `h2Nom`, et appliquez-lui les propriétés CSS pour obtenir une police blanche et un texte centré. Ajouter l'élément `h2Nom` à la division `divGrandeImage`.
- 5.7. Dupliquer l'image cliquée en la clonant avec jQuery et mémoriser l'objet récupéré dans une variable `grandeImage`. Appliquez-lui une largeur de 600 pixels et ajoutez-la à la division `divGrandeImage`.
Rappel : dans une fonction déclenchée par un événement, `$(this)` permet d'accéder à l'élément qui a capté l'événement.
- 5.8. Créer un nouvel élément de type `<h2>` et mémorisez-le dans une variable `h2Scene`.
Ajouter le nom de la scène et le jour comme texte à l'élément `h2Scene`, et appliquez-lui les propriétés CSS pour obtenir une police blanche et un texte centré. Ajouter l'élément `h2Scene` à la division `divGrandeImage`.
Indication : la méthode `.next()` de jQuery appliquée à un élément sélectionné permet de récupérer l'élément frère suivant.
- 5.9. Transformer la division `divFond` en boite flexible et ajouter les propriétés CSS pour que l' élément qu'elle contient (`divGrandeImage`) soit centré horizontalement et verticalement.
- 5.10. Sur l'élément `divFond` ajouter un écouteur d'événement 'click' avec une fonction déclenchée permettant de le retirer du DOM HTML.