

TP 7 - jQuery

Exercice 1 : prise en main de jQuery

1. Récupérer le fichier cv.html ci-dessous :

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>mon cv</title>
  </head>
  <body>
    <div class="nom">Jérémie Dupont</div>
    <p id="datenaiss">01/01/1992</p>
    <p>Courriel :
      <a href="mailto:michel.martin@gmail.com">michel.dupont@gmail.com</a>
    </p>
    <h1 id="titre" class="titre">Développeur web </h1>
    <section>
      <h2 class="titre">Expérience professionnelle</ h2>
      <p class="annee" data-annee="experience">2015-2017</p>
      <p>développeur web pour l'agence triangulaire</p>
    </section>
    <section>
      <h2 class="titre">formation</ h2>
      <p class="annee" data-annee="formation">2013-2014</p>
      <p> L1 informatique - Université JF Champollion</p>
      <p class="annee" data-annee="formation">2014-2015</p>
      <p> L2 informatique - Université JF Champollion </p>
      <h2 class="titre" id="divers">divers</h2>
      <ul>
        <li>lecture de romans policiers</li>
        <li>amateur de cinéma </li>
      </ul>
    </section>
  </body>
</html>
```

2. Insérez le lien vers la bibliothèque jQuery hébergée chez Google :

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"> </script>
```

3. Positionnez un écouteur d'événement sur l'objet document pour que s'affiche le message « document chargé » dans la console lorsque le document est chargé.

Documentation: <http://api.jquery.com/ready/>

4. A l'aide de commandes jQuery, afficher dans la console :

- a. l'élément body
- b. la valeur de l'élément d'identifiant datenaiss
- c. l'élément de classe titrecv
- d. la valeur des éléments de classe annee

NB : pour appliquer un traitement à chaque élément on utilise la méthode `.each()`

```
$("#td").each( ... $(this)... ) avec $(this) est l'objet courant
```

- e. le nombre d'éléments p fils de body
- f. le deuxième élément p fils de body
- g. l'élément script ayant un attribut src
- d. l'élément d'attribut data-annee égal à formation

5. Modifiez le premier titre H1 du CV avec jQuery en insérant "M. " avant le prénom de la personne.

6. Ajoutez une commande jQuery permettant de cacher tous les liens de la page.

7. Ajoutez une commande jQuery mettant les années en gras

8. A l'aide de jQuery, faites suivre la date de naissance par l'âge de la personne, de telle sorte que l'âge soit mis à jour automatiquement en fonction de la date de naissance (1^{er} janvier) et de la date actuelle.

9. Ajoutez "Horaires souples (soirées, week-ends, etc.)" comme premier élément de la liste dans la section "Divers" si le visiteur du CV le regarde le week-end ou après 20h.

NB : pour ajouter un noeud dans l'arborescence

- insertion/modification du texte `.text(...)`
- insertion/modification d'un sous arbre `.html(...)`
- insertion d'enfants `.append(...)` à la fin de la sélection, `.prepend()` au début de la sélection.
- insertion de frères `.after(...)` `.before(...)`
- insertion d'un père `.wrap()`

exemple : `$('#datenaiss').wrap('<h2>'); // c'est le < et > qui indique qu'il s'agit d'une création`

Exercice 2 : diaporama en jQuery

- Créer un nouveau fichier html contenant le code suivant :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Diaporama avec JS</title>
  <link rel="stylesheet" type="text/css" href="stylejs.css">
</head>
<body>
  <div id="diaporama">
    <!-- Les images -->
    <div id="contenu" >
      
    </div>
    <!-- Boutons de navigation -->
    <div id="btnNavigation">
      <button id="prem">Première</button>
      <button id="prec">Précédent</button>
      <button id="lect">Lecture</button>
      <button id="suiv">Suivant</button>
      <button id="dern">Dernière</button>
    </div>
  </div>
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
  <script>
    $.ready(function() {
    })
  </script>
</body>
</html>
```

- et un fichier stylejs.css contenant le code suivant :

```
#diaporama {
  position: relative;
  width: 700px;
  height: 300px;
  margin: 20px auto;
  overflow: hidden;
  text-align:center;
}
#contenu {
  position: absolute;
  margin:0;
  top: 0;
  left: 0;
  width: 700px;
  height: 250px;
  padding: 0;
}
#btnNavigation {
  margin:auto;
  margin-top: 250px;
}
```

- Tester (seule la première image est affichée au centre du navigateur avec en dessous 5 boutons)

- **Etape 1** : les boutons *première* et *dernière*
 - ajouter un écouteur d'événement pour le bouton *dernière*:

<i>type d'événement</i>	click
<i>cible / target</i>	le bouton <i>dernière</i> ayant pour id "dern"
<i>action associée</i> (sous forme de fonction)	afficher la dernière diapo (à l'aide de la fonction <i>derniere</i>)

- soit en jQuery :

```
$("#dern").click(derniere);
```

- il nous faut maintenant trouver comment changer d'image ; pour simplifier l'écriture du code on partira du postulat que les images à afficher sont toutes dans le dossier images et qu'elles sont nommées de manière continue : 1.jpg, 2.jpg, 3.jpg, ...
Le nombre d'images total sera définie par une constante javascript nommée NOMBRE_IMAGES
Donc la première image s'appellera toujours 1.jpg et la dernière NOMBRE_IMAGES.jpg.

Dans notre cas, on aura donc le code JS suivant :

```
const NOMBRE_IMAGES = 5;

$("#dern").click(derniere);

function derniere(event) {
    $("#image").attr("src", "images/" + NOMBRE_IMAGES + ".jpg");
}
```

- Tester que le bouton *derniere* fonctionne.
- Ecrire un code similaire pour faire fonctionner le bouton *premiere*

- **Etape 2** : les boutons *suivante* et *précédente*
 - Définir un gestionnaire d'événement pour le bouton *suivante* :

<i>type d'événement</i>	click
<i>cible / target</i>	le bouton <i>suivante</i> ayant pour id "suiv"
<i>action associée</i>	afficher la diapo suivante ; si c'est la dernière diapo, on ne fait rien (à l'aide de la fonction <i>avancer</i>)

- Ecrire la fonction *avancer()*
- Faire de même pour le bouton *précédente*

- **Etape 3 : le bouton lecture**

- Tout d'abord, on va s'intéresser à la valeur du bouton ; en effet si on clique sur le bouton *lecture*, il faut qu'il affiche *pause*. Et si on clique sur *pause*, il faut qu'il affiche *lecture* :

○

type d'événement	click
cible / target	le bouton lecture ayant pour id "lect"
action associée	changer de manière alternative la valeur du bouton lecture/pause/lecture/pause/... (à l'aide de la fonction <i>lecture</i>)

- Là aussi, pour savoir dans quel état est le lecteur de diapo (pause ou lecture), il nous faut une variable javascript de type booléen par exemple :

```
var lectureOk = false; // diapo en lecture (true) ou en pause (false)
```

- Ecrire maintenant la fonction *lecture()* qui va donc changer la valeur du bouton (méthode html) en fonction de la valeur de la variable *lectureOk*.
- Pour terminer, il reste maintenant à faire fonctionner le changement automatique d'image lorsque le lecteur de diapo est en mode lecture ; pour cela il faut écrire une fonction *diapoSuivante()* qui passe à la diapo suivante (avec retour automatique à la diapo 1 quand on est en fin de diapo).

Puis pour que cette fonction soit exécutée toutes les 2 secondes par exemple, on va utiliser la fonction *setInterval*(nomdefonction, 2000) qui va exécuter de manière automatique la fonction passée en paramètre avec un délai (en nombre de millisecondes) entre chaque exécution.

Ecrire la fonction *diapoSuivante()*

Modifier le code de la fonction *lecture()* pour utiliser la fonction *setInterval* qui exécutera automatiquement la fonction *diapoSuivante* avec le délai que vous aurez choisi.

- Pour le mode pause, il faut utiliser la fonction *clearInterval()* [voir la doc en ligne]