

TD/TP4 – Introduction au jQuery

A. Exercices sur jQuery (TD de 3h)

Résultat d'apprentissage visé : En répondant à chaque question, les élèves doivent comprendre d'utiliser des fonctions jQuery selon les exigences demandées pour rendre la page web plus dynamique

1. Comment invoquer jQuery (la version hébergée par Google code) dans votre page Web ?

- Dans la partie `<head></head>`, ajouter cette déclaration :
`<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js">
</script>`

2. Contrôle sur la souris : jQuery contient des fonctionnalités pour contrôler la souris

- Avec `click()`, faire disparaître un paragraphe s'il est cliqué.
- Avec `dblclick()`, faire disparaître un paragraphe s'il est double-cliqué.
- Avec `mouseenter()`, faire disparaître lentement un paragraphe sil la souris est entré dans ce paragraphe.
- Avec `mouseleave()`, faire disparaître lentement un paragraphe sil la souris est sortie de paragraphe.
- Toutes ces fonctionnalités sont applicables aussi pour les autres éléments de votre page web. Par exemple, remplacer le paragraphe par une `image` et appliquer et tester `click()`, `dblclick()`, `mouseenter()` et `mouseleave()` pour cet élément.

NB : En jQuery, quand on modifie des éléments de la page, cela marchera seulement s'ils sont déjà chargés. La fonction `ready()` nous permet d'attendre que ce soit le cas avant d'exécuter le code.

```
$(document).ready(function() {  
    // code lu uniquement une fois le document chargé  
});
```

On peut également l'écrire de cette manière :

```
$(function() {  
    // code lu uniquement une fois le document chargé  
});
```

- **Étapes à suivre :**
- Dans la partie `<body></body>` de votre page HTML, ajouter le code suivant :

```
<!-- Clic -->  
<p class="clic">Si vous cliquez sur moi, je vais disparaître.</p>  
<p class="clic">Cliquez sur moi !</p>  
<p class="clic">Cliquez-moi aussi !</p>
```
- Qu'est-ce que c'est l'attribut `"class"` dans le paragraphe ?
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :

```
$(document).ready(function() {  
    // votre code ici !  
});
```
- En sélectionnant la classe « clic » de votre code parce qu'elle a été cliquée, on va sélectionner l'élément concerné et on le cache. Pour le faire, ajouter ce code :

```
$(".clic").click(function() {  
    $(this).hide();  
});
```
- Voilà ! Quand vous cliquez sur n'importe quel paragraphe de la classe « clic », il va disparaître grâce à votre code jQuery.
- Répétez le même processus. Ajouter des codes pertinents dans la partie `<body></body>` et `<script></script>` pour implémenter la fonction `dblclick()`. Cette fois-ci, c'est quand on double clique sur un élément (un paragraphe, dans cet exemple), cet élément disparaît.
- Cette fois, dans la partie `<body></body>`, ajouter ce code :

```
<!-- Mouse enter -->  
<p id="souris1">Bonjour Paris ! </p>
```
- Quelle est la différence entre « classe » et « id » dans le contexte de jQuery ?

- Dans la partie `<script></script>`, ajouter un code pour quand on bouge la souris vers ce paragraphe, ce texte disparaît doucement, en quelques millisecondes. Pour le faire, ajouter ce code :

```
$("#souris1").mouseenter(function() {
    $("#souris1").fadeOut(2000);
});
```
- Répétez le même processus. Ajouter des codes pertinents dans la partie `<body></body>` et `<script></script>` pour implémenter la fonction `mouseleave()`. Cette fois-ci, c'est quand on bouge la souris hors de cet élément (un paragraphe, dans cet exemple), cet élément disparaît doucement, en quelques millisecondes.

3. Toujours avec le contrôle de la souris, afficher sept images dans votre page web. Ensuite, faire disparaître une image à la fois en utilisant `click()`, `dblclick()`, `mouseenter()`, `mouseleave()`, `mousedown()`, `mouseup()` et `hover()`. Notez bien que `hover()` est la combinaison de `mousedown()` et `mouseup()`. Par exemple, appliquer les contrôles de la souris aux images des oiseaux :

Contrôle de la souris



- **Étapes à suivre :**
- Dans la partie `<body></body>` de votre page HTML, ajouter le code suivant :

```
<!-- images -->







```
- Ces images sont disponibles sur Boostcamp et à télécharger pour que votre code marche bien.
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :

```
$(document).ready(function() {
    // votre code ici !
});
```
- Répétez le même processus qu'on a fait dans le dernier exercice. Implémenter la fonction `click()` pour faire disparaître la première image, `dblclick()` pour faire disparaître la deuxième image, `mouseenter()` pour faire disparaître doucement en quelques millisecondes la troisième image, `mouseleave()` pour faire disparaître doucement en quelques millisecondes la quatrième image, `mousedown()` pour faire disparaître doucement en quelques millisecondes la cinquième image, `mouseup()` pour faire disparaître doucement en quelques millisecondes la sixième image, et `hover()` pour faire disparaître doucement en quelques millisecondes la dernière image.

4. La fonction `focus()` est exécutée lorsque le champ de formulaire obtient le focus (lorsqu'il est sélectionné par un clic de souris ou en "naviguant par tabulation" jusqu'à celui-ci). La fonction `blur()` est exécutée quand le champ de formulaire perd le focus. Appliquer `focus()` et `blur()` dans notre formulaire ci-dessous :

Dynamisme avec jQuery	
Nom	<input type="text"/>
Age	<input type="text"/>
Téléphone	<input type="text"/>
Date de naissance	<input type="text" value="jj/mm/aaaa"/> <input type="button" value="📅"/>
<input type="button" value="Soumettre"/>	

- **Étapes à suivre :**
- Le code pour réaliser ce formulaire est déjà disponible dans votre TD/TP 3 – « Dynamisme avec JavaScript ». Retapez le code ou implémentez-le à nouveau maintenant.
- Dans la partie `<body></body>` de votre page HTML, ajouter le code suivant :


```
<!-- images -->







```
- Ces images sont disponibles sur Boostcamp et à télécharger pour que votre code marche bien.
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :


```
$(document).ready(function() {
  // votre code ici !
});
```
- Répétez le même processus qu'on a fait dans le dernier exercice. Implémenter la fonction `click()` pour faire disparaître la première image, `dblclick()` pour faire disparaître la deuxième image, `mouseenter()` pour faire disparaître doucement en quelques millisecondes la troisième image, `mouseleave()` pour faire disparaître doucement en quelques millisecondes la quatrième image, `mousedown()` pour faire disparaître doucement en quelques millisecondes la cinquième image, `mouseup()` pour faire disparaître doucement en quelques millisecondes la sixième image, et `hover()` pour faire disparaître doucement en quelques millisecondes la dernière image.

5. La fonction `hide()` masque un élément dans la page web, tandis que `show()` affiche cet élément. La fonction `toggle()` bascule entre `hide()` et `show()`. Appliquer `hide()`, `show()` et `toggle()` dans une image dans votre page web.

- **Étapes à suivre :**
- Dans la partie `<body></body>` de votre page HTML, ajouter le code suivant :


```
 <br>
<button id="hide">Masquer</button>
<button id="show">Afficher</button>
<button id="toggle">Basculer</button>
```
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :


```
$(document).ready(function() {
  // votre code ici !
});
```
- Implémenter la fonction `click()` pour faire disparaître l'image quand le bouton « Masquer » a été cliqué. Dans ce cas, l'image disparaît en invoquant la fonction `hide()`.
- Implémenter la fonction `click()` pour faire apparaître l'image quand le bouton « Afficher » a été cliqué. Dans ce cas, l'image apparaît en invoquant la fonction `show()`.
- Implémenter la fonction `click()` pour faire apparaître ou disparaître l'image quand le bouton « Basculer » a été cliqué. Dans ce cas, l'image apparaît ou disparaît en invoquant la fonction `toggle()`.

6. La fonction `fadeIn()` fait apparaître un élément dans la page web, tandis que `fadeOut()` fait disparaître cet élément. La fonction `fadeToggle()` bascule entre `fadeIn()` et `fadeOut()`. La fonction `fadeTo()` est comme `fadeIn()` avec une valeur d'opacité. Appliquer `fadeIn()`, `fadeOut()`, `fadeTo()` et `fadeToggle()` avec variations de paramètres dans trois carrés dans votre page web.

- **Étapes à suivre :**
- Dans la partie `<body></body>` de votre page HTML, ajouter le code suivant :


```
<h2>Exemples de fadeIn()</h2>
<button id="fadeIn1">Fondu entrant 1</button>
<button id="fadeIn2">Fondu entrant 2</button>
<button id="fadeIn3">Fondu entrant 3</button><br><br>
```
- On ajoute également les codes suivants :


```
<div id="div1"></div><br>
<div id="div2"></div><br>
<div id="div3"></div><br>
```
- Nous voulons créer trois carrés. Dans la partie `css` de votre page web, ajouter le code suivant :


```
div {
  width: 100px;
  height: 100px;
}
#div1 {
  background-color: red;
  display: none;
```

```

    }
    #div2 {
        background-color: green;
        display: none;
    }
    #div3 {
        background-color: blue;
        display: none;
    }
    }
    > Au début, les trois carrés ne sont pas affichés dans votre page web (display: none;)
    > Dans la partie <script></script>, ajouter cette déclaration d'invocation de jQuery :
        $(document).ready(function() {
            // votre code ici !
        });
    > Quand on clique sur le texte, avec fadeIn(), un carré va apparaître. Voici le code pour le faire :
    //fadeIn
    $("#fadeIn1").click(function(){
        $("#div1").fadeIn();
    });

    $("#fadeIn2").click(function(){
        $("#div2").fadeIn("slow");
    });

    $("#fadeIn3").click(function(){
        $("#div3").fadeIn(5000);
    });
    > Etudier les codes et voir ses résultats.
    > Utiliser le même principe pour implémenter les autres fonctions souhaitées : fadeOut(), fadeTo() et fadeToggle()

```

7. La fonction `animate()` est utilisée pour personnaliser une animation. Appliquer `animate()` pour modifier les paramètres d'un élément div dans votre page web.

```

    > Étapes à suivre :
    > Dans la partie <body></body> de votre page HTML, ajouter le code suivant :
    <button>Commencer animation</button>
    > On ajoute également le code suivant :
    <div id="div1"></div>
    > Nous voulons créer un carré. Dans la partie css de votre page web, ajouter le code suivant :
    div {
        background-color: #98bf21;
        height: 100px;
        width: 100px;
        position: absolute;
    }
    > Dans la partie <script></script>, ajouter cette déclaration d'invocation de jQuery :
        $(document).ready(function() {
            // votre code ici !
        });
    > Quand on clique sur le bouton, on fait une petite animation. Pour le faire, ajouter ce code :
    $("#button").click(function(){
        $("#div1").animate({height: "300px", opacity: "0.5"}, "slow");
        $("#div1").animate({width: "1000px", opacity: "0.8"}, "slow");
        $("#div1").animate({height: "600px", opacity: "0.2"}, "slow");
        $("#div1").animate({width: "500px", opacity: "0.7"}, "slow");
    });
    > Ajouter vos codes d'animation avec la fonction animate().

```

8. Avec jQuery, vous pouvez créer un effet de glissement sur les éléments. Appliquer `slideUp()`, `slideDown()`, `slideToggle()`, et la combinaison de `slideUp()` et `slideDown()` en 2000 ms dans une image pour animer votre page web.

```

    > Étapes à suivre :
    > Dans la partie <body></body> de votre page HTML, ajouter le code suivant :
    <p>jQuery est amusant !</p>
     <br>
    <button id="button1">slideUp</button>
    <button id="button2">slideDown</button>
    <button id="button3">slideToggle</button>

```

- `<button id="button4">slideUp et slideDown</button>`
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :


```
$(document).ready(function() {
    // votre code ici !
});
```
- Si on clique sur le premier bouton (`slideUp`), l'image apparaît avec un glissement vers le haut. Pour le faire, on ajoute le code suivant :


```
$("#button1").click(function(){
    $("#img").slideUp(2000);
});
```
- Utiliser le même principe et implémenter les autres fonctions souhaitées : `slideDown()`, `slideToggle()` et tous les deux, `slideUp()` puis `slideDown()`.

9. On va maintenant utiliser la fonction `css()` qui permet de changer les propriétés CSS. Comment mettre (i) *La taille des `<h1>` à 50px*, (ii) *L'id `#un_id` en rouge*, et (iii) *cachez la classe `.une_classe`*.

- **Étapes à suivre :**
- Dans la partie `<body></body>` de votre page HTML, ajouter le code suivant :


```
<h1>Exercice sur jQuery</h1>
<div id="un_id"> Bonjour Paris! </div>
<p class="une_classe">Jeux olympiques Paris 2024</p>
<button id="button1">Cliquez pour changer la taille de h1</button> <br> <br>
<button id="button2">Cliquez pour changer la couleur de l'id</button> <br> <br>
<button id="button3">Cliquez pour cacher la classe</button>
```
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :


```
$(document).ready(function() {
    // votre code ici !
});
```
- Quand on clique sur le premier bouton, la taille de la police de h1 change à 50px. Pour le faire, ajouter ce code :


```
$("#button1").click(function(){
    $('h1').css('fontSize', '50px');
});
```
- En utilisant le même principe et toujours avec la fonction `css()`, modifier la *couleur* d'un *id* quand on clique sur le *deuxième bouton*. Toujours avec la fonction `css()` de jQuery, faire disparaître une *class* quand le *troisième bouton* est cliqué.

10. Comment ajouter un élément dans une liste d'élément non ordonnée ? Par exemple, si la liste présente les éléments "Python" et "PHP", en cliquant sur le bouton, on ajoute l'élément "Java" dans la liste.

- **Étapes à suivre :**
- Dans la partie `<body></body>` de votre page HTML, ajouter le code suivant :


```
<div id="header">
    <ul>
        <li><a href="http://www.w3resource.com/python/python-tutorial.php"> Python</a></li>
        <li><a href="http://www.w3resource.com/php/php-home.php">PHP</a></li>
    </ul>
</div>
```
- On ajoute également un *bouton* cliquable par ce code :


```
<input type="button" value="Cliquez pour ajouter un nouvel élément li"
    onclick="newElement()" />
```
- Dans la partie `<script></script>`, on doit créer la fonction demandée : `function newElement() {}`
- Dans cette fonction, on utilise la méthode `append()` pour ajouter l'élément demandé. Voici le code :


```
$("#header ul").append('<li><a href="http://www.w3resource.com/java/java-tutorial/">
    Java</a></li>');
```

11. Un utilisant jQuery, comment définir la couleur de fond (valeur : #add080) pour les éléments "p", "span" et "textarea" ?

- **Étapes à suivre :**
- Dans la partie `<body></body>` de votre page HTML, mettre des éléments `<p>`, `` et `<textarea>`. Ajouter également un bouton. Pour le faire, ajouter le code suivant :


```
<textarea>Exercice sur jQuery</textarea>
<p>jQuery</p>
<span>Exercices</span>
<button id="button1">Cliquez pour voir l'effet</button>
```

- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :

```
$(document).ready(function() {
    // votre code ici !
});
```
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :

```
$(document).ready(function() {
    // votre code ici !
});
```
- Dans cette fonction, on utilise les méthodes `add()` pour considérer plusieurs éléments en même temps, puis la méthode `css()` pour contrôler leurs propriétés. Ajouter le code suivant :

```
$('#button1').click(function(){
    $('p').add('span').add('textarea').css('background-color', '#add080');
});
```

12. En cliquant sur un div, comment afficher un texte disant la couleur de ce div ?

- **Étapes à suivre :**
- Dans la partie `<body></body>` de votre page HTML, mettre plusieurs éléments de type `<div>`. Ajouter également un paragraphe. Pour le faire, voici le code :

```
<div style="background-color:red;">Div 1</div>
<div style="background-color:rgb(16,99,30);">Div 2</div>
<div style="background-color:#523456;">Div 3</div>
<div style="background-color:#a11;">Div 4</div>
<p></p>
```
- Dans la partie `css` de votre page web, ajouter des styles pour `<div>` et `<p>`. Voici le code :

```
div {
    width: 60px;
    height: 60px;
    margin: 5px;
}
p {
    margin-top:20px;
    padding: 5px;
    border: 2px solid #666;
}
```
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :

```
$(document).ready(function() {
    // votre code ici !
});
```
- Dans cette fonction, on va utiliser la méthode `css()` de jQuery pour récupérer la couleur d'un div cliqué et la méthode `html()` de jQuery pour afficher un message dans le paragraphe. Pour le faire, ajouter le code suivant :

```
$('#div').click(function() {
    var color = $(this).css('background-color');
    $('p').html("That div is " + color + ".");
});
```

13. Comment désactiver/activer le bouton de soumission d'un formulaire ? Par exemple, désactivez le bouton "Soumettre" jusqu'à ce que le visiteur ait cliqué sur une case à cocher.

- **Étapes à suivre :**
- Dans la partie `<body></body>` de votre page HTML, nous voulons mettre un **checkbox** ☐ (case à cocher en français) à côté d'un texte (accepter les conditions). Ajouter un bouton également. Pour le faire, voici le code :

```
<input id="accept" name="accept" type="checkbox" value="y"/>J'accepte les conditions.<br>
<input id="submitbtn" disabled="disabled" name="Submit" type="submit" value="Submit" />
```
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :

```
$(document).ready(function() {
    // votre code ici !
});
```
- Dans cette fonction, nous souhaitons que quand le **checkbox** est activé, le bouton (Soumettre) est actif également. Sinon, le bouton est désactivé. Pour le faire, ajouter le code suivant :

```
$('#accept').click(function() {
    if ($('#submitbtn').is(':disabled')) {
        $('#submitbtn').removeAttr('disabled');
    } else {
        $('#submitbtn').attr('disabled', 'disabled');
    }
});
```

14. Comment créer un élément de paragraphe avec du texte et l'ajoutez à la fin du corps du document à l'aide de jQuery.

➤ **Étapes à suivre :**

- Dans la partie `<body></body>` de votre page HTML, ajouter un texte dans un paragraphe. Par exemple, considérer ce code :
`<p>Ceci est le texte original de ce document.</p>`
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :
`$(document).ready(function() {
 // votre code ici !
});`
- Dans cette fonction, nous voulons que quand le paragraphe est cliqué, on va ajouter un autre paragraphe en utilisant la méthode `appendTo()`. Pour le faire, ajouter le code suivant :
`$("p").click(function(){
 $("<p>Ceci est le texte ajouté à la fin du document.</p>").appendTo(document.body);
});`

15. Comment associer un événement clic et double-clic à l'élément paragraphe `<p>`. Exemple de données: `<body><p> Cliquez-moi! </p></body>`. Pour un simple clic, ajoutez le paragraphe suivant: Ceci est un événement clic. Pour double-cliquer, ajoutez le paragraphe suivant: Ceci est un événement double-clic.

➤ **Étapes à suivre :**

- Dans la partie `<body></body>` de votre page HTML, comme indiqué dans le sujet, ajouter un texte dans un paragraphe. On ajoute également un élément `<div>`. Voici le code :
`<p>Cliquez-moi!</p>
<div id="resultat"></div>`
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :
`$(document).ready(function() {
 // votre code ici !
});`
- Dans cette fonction, nous observons comment le paragraphe est cliqué. *S'il est cliqué*, on affiche un message approprié dans `<div>` de notre code, partie `<body>`. *S'il est double-cliqué*, on affiche un message qu'il est double-cliqué dans `<div>` de notre code. On va utiliser la méthode `bind()` de jQuery. Voici le code :
`$("p").bind("click", function(){
 $("<p>Ceci est un événement clic.</p>").appendTo("#resultat");
});
$("p").bind("dblclick", function(){
 $("<p>Ceci est un événement double-clic.</p>").appendTo("#resultat");
});`

16. Comment changer la couleur de fond d'un paragraphe s'il y a un événement double-clic.

➤ **Étapes à suivre :**

- Dans la partie `<body></body>` de votre page HTML, ajouter un texte dans un paragraphe. Voici un exemple :
`<p>Double-cliquez-moi pour changer la couleur de fond</p>`
- Dans la partie CSS, ajouter les propriétés `background` et `color` pour le paragraphe et
`p { background: blue;
 color: white;
}
p.dbl { background: red;
 color: white;
}`
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :
`$(document).ready(function() {
 // votre code ici !
});`
- Dans cette fonction, nous voulons modifier la couleur de fond de notre paragraphe selon la manière qu'il est cliqué ou double-cliqué. Voici le code :
`var pdbl = $("p");
pdbl.dblclick(function() {
 pdbl.toggleClass("dbl");
});`
- Analyser le code. Dans ce code, on a utilisé la méthode `dblclick()` et `toggleClass()`. Tester le code pour voir le résultat.

17. Comment obtenir les styles (*largeur* (*width*), *hauteur* (*height*), *couleur* du texte (*color*) et *couleur* de fond (*background-color*)) d'une division `<div>` cliquée

- **Étapes à suivre :**
- Dans la partie `<body></body>` de votre page HTML, nous voulons ajouter trois `<div>` et un paragraphe `<p>`. Voici le code :


```
<div id="box1">A</div>
<div id="box2">B</div>
<div id="box3">C</div>
<p id="resultat">&nbsp;</p>
```
- Dans la partie `css` de votre code, nous voulons configurer les trois `<div>` pour avoir trois carrés avec couleurs différents. Nous voulons également ajouter de style pour notre paragraphe. Voici le code de style :


```
div {
  height: 50px;
  margin: 5px;
  padding: 5px;
  float: left;
}
#box1 {
  width: 50px;
  color: yellow;
  background-color: red;
}
#box2 {
  width: 50px;
  color: rgb(255, 255, 255);
  background-color: rgb(15, 99, 30);
}
#box3 {
  width: 50px;
  color: #fcc;
  background-color: #123456;
}
p {
  clear: both;
  color: blue;
}
```
- Étudier le code de votre style css pour bien comprendre ses résultats.
- Dans la partie `<script></script>`, ajouter cette déclaration d'invocation de jQuery :


```
$(document).ready(function() {
  // votre code ici !
});
```
- Dans cette fonction, quand un carré est cliqué, on récupère ses propriétés (*largeur*, *hauteur*, *couleur* et *couleur de fond*), et puis, on les affiche dans notre paragraphe. Voici le code :


```
$( "div" ).click(function() {
  var html = [ "Le div cliqué a les styles suivants:" ];
  var styleProps = $(this).css([
    "width", "height", "color", "background-color"
  ]);
  $.each( styleProps, function( prop, value ) {
    html.push( prop + ": " + value );
  });
  $( "#resultat" ).html( html.join("<br>") );
});
```
- Étudier le code de votre style css pour bien comprendre ses résultats. Notez bien qu'on utilise le tableau dont chaque élément contient une propriété et sa valeur.

B. Travail Pratique sur jQuery : Le Carrousel (1h)

Résultat d'apprentissage visé : En suivant les étapes mentionnées dans cet exercice, les élèves doivent apprendre comment formuler un carrousel en utilisant la programmation JavaScript et les fonctionnalités jQuery

- **Ce TP est un devoir individuel (solo) et à soumettre sur Boostcamp.**

Cet exercice est consacré aux animations et à la manipulation du DOM. Votre travail consiste à réaliser un carrousel grâce à jQuery.

Explications et prérequis

Un carrousel, c'est tout simplement un défilement d'images, le plus souvent présent sur l'index d'un site web. Cela permet de gagner de la place, tout en restant très esthétique, et faire une présentation originale

Objectif

Votre travail est de réaliser un carrousel. Ce carrousel, devra comporter quelques fonctions basiques :

- Il devra posséder deux boutons, un pour afficher l'image précédente, l'autre pour afficher la suivante ;
- Il saura défiler seul, c'est-à-dire que les images défileront d'elles-mêmes sans que l'on ait à intervenir ;
- Il pourra être très flexible ; il sera possible de mettre autant d'images que l'on souhaite.

Méthodes requises

Ce TP est complet dans le sens qu'il utilise tous les domaines de jQuery : les événements, les sélecteurs, les fonctions de parcours, la manipulation du DOM.

Vous devrez donc partir d'un squelette en HTML. En effet, l'astuce est de définir une taille fixe à votre carrousel, les images étant empilées les unes sur les autres grâce à une position absolue.

Une fois terminé, tout se passe ensuite du côté de jQuery. L'astuce est de jouer sur la présence des images, grâce à la propriété `display` de CSS. N'hésitez pas non plus à manipuler les index, car c'est comme cela que vous pourrez déterminer quelle image doit être affichée. Enfin, les boutons « **précédent** » et « **suivant** » devront être ajoutés à la volée.

setTimeout(), ou répéter une fonction régulièrement

Une méthode native de JavaScript est utile pour faire défiler automatiquement les images. Il s'agit de `setTimeout()`, qui prend deux arguments : le nom de la fonction à exécuter, et l'intervalle de temps (en millisecondes) à attendre avant de le faire. L'astuce, c'est de relancer la fonction à l'intérieur d'elle-même pour réaliser une boucle infinie :

```
function maBoucle(){
  setTimeout(function(){
    alert('Bonjour ECE Paris!'); // affichera "Bonjour ECE Paris!" toutes les secondes
    maBoucle(); // relance la fonction
  }, 1000);
}
maBoucle(); // on n'oublie pas de lancer la fonction une première fois
```

Si vous souhaitez partir sur de bonnes bases, nous vous suggérons de prendre ce structure HTML et le style CSS minimal. On va afficher sept images de France (disponible sur Boostcamp avec noms de "france1.jpg", "france2.jpg", ..., "france7.jpg").

```
<div id="carrousel">
<ul>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
```

```
</ul>
</div>
```

Les images utilisées ici proviennent des images gratuites disponibles sur le net. Si vous voulez, vous pouvez changer ces images par celles que vous aimez.

```
#carrousel {
position: relative;
height: 200px;
width: 700px;
margin: auto;
}
#carrousel ul li{
position: absolute;
top: 0;
left: 0;
}
h1 {text-align: center;
color: red;
background-color: #ADD8E6;
}
```

Vous avez maintenant toutes les clés pour réaliser ce carrousel !

Correction

Votre carrousel fonctionne bien ? Ou en est-il toujours à la phase théorique ? Quoi qu'il en soit, nous vous proposons comme de juste une correction. Suivez-la bien pas à pas, et vous saurez quelles ont été vos erreurs si vous en avez commis. Nous vous avons fourni le code HTML minimal permettant de constituer une base de travail :

```
<h1> La belle France</h1>
<div id="carrousel">
<ul>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
</ul>
</div>
```

Notre carrousel possède donc une structure bien simple : une liste non-ordonnée d'images. La première chose à faire était évidemment d'inclure jQuery.

```
<head>
<!-- on inclut la bibliothèque depuis les serveurs de Google -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
</head>
```

Il fallait également inclure correctement le style CSS minimal du carrousel, qui consistait à définir les positions absolues des images et la position relative du carrousel.

Explications du code jQuery

De même que pour l'inclusion de la librairie, il fallait tout d'abord s'assurer que le DOM était correctement chargé, grâce à l'évènement `ready()` lancé sur le document :

```
$(document).ready(function(){  
    // notre code ici  
});
```

Plusieurs variables devaient ensuite être définies. Ici, il suffisait de cibler seulement deux éléments : le bloc du carrousel, et les images contenues par celui-ci.

Nous vous avons mis sur une petite piste : les index des images. Grâce à un système de compteur, vous pouviez contrôler l'affichage des images par leur index. Il était donc d'usage de définir le nombre d'images qu'il y avait dans la liste, puis de soustraire 1 pour obtenir l'index de la dernière diapositive (rappel : on commence de compter les index à partir de 0).

Une fois ceci fait, il fallait obtenir l'image courante, c'est-à-dire la première image sur laquelle notre *slider* devait commencer. Grâce à la fonction `eq()`, vous pouviez la cibler très facilement.

```
var $carrousel = $('#carrousel'), // on cible le bloc du carrousel  
    $img = $('#carrousel img'), // on cible les images contenues dans le carrousel  
    indexImg = $img.length - 1, // on définit l'index du dernier élément  
    i = 0, // on initialise un compteur  
    $currentImg = $img.eq(i); // enfin, on cible l'image courante, qui possède l'index i (0 pour l'instant)
```

Une fois les variables créées, il fallait s'assurer que toutes les images sauf une étaient invisibles. La propriété CSS `display : none` vous permettait de faire cela : on cache toutes les images, puis on affiche seulement l'image courante.

```
$img.css('display', 'none'); // on cache les images  
$currentImg.css('display', 'block'); // on affiche seulement l'image courante
```

Créer les contrôles : image précédente et image suivante

Employons-nous à présent à travailler sur les contrôles des images, c'est-à-dire les fonctions d'affichage des images précédentes ou suivantes. Nous vous suggérons `append()`, bien que vous pouviez tout aussi bien employer d'autres fonctions.

Nous avons choisi de créer un bloc `div` contenant deux `span`, ayant respectivement la classe `.prev` et `.next`.

```
$carrousel.append('<div class="controls"> <span class="prev">Précédent</span>  
<span class="next">Suivant</span> </div>');
```

En écoutant le clic sur ces éléments, il était possible de déterminer s'il fallait passer à l'image suivante, ou l'image précédente. Dans les deux cas, il suffisait de jouer avec l'index des diapositives : si l'on va à l'image suivante, l'index s'incrémente de 1, si on va à l'image précédente, il se décrémente de 1.

```
$('.next').click(function(){ // image suivante  
    i++; // on incrémente le compteur  
    $img.css('display', 'none'); // on cache les images  
    $currentImg = $img.eq(i); // on définit la nouvelle image  
    $currentImg.css('display', 'block'); // puis on l'affiche  
});  
  
$('.prev').click(function(){ // image précédente  
    i--;  
    // on décrémente le compteur, puis on réalise la même chose que pour la fonction "suivante"  
    $img.css('display', 'none');
```

```
$currentImg = $img.eq(i);  
$currentImg.css('display', 'block');  
});
```

On se heurte alors à un nouveau problème : si l'on clique trop de fois sur une des deux fonctions, alors le compteur ne suit plus les index des images. On peut alors se retrouver avec une image courante qui n'existe pas, et qui ne peut donc pas s'afficher. Pour remédier à ce problème, il suffit de s'assurer grâce à une condition que le compteur ne dépasse pas le dernier index, et ne puisse pas aller en dessous de 0:

```
$('.next').click(function(){ // image suivante  
    i++; // on incrémente le compteur  
    if( i <= indexImg ){  
        $img.css('display', 'none'); // on cache les images  
        $currentImg = $img.eq(i); // on définit la nouvelle image  
        $currentImg.css('display', 'block'); // puis on l'affiche  
    }  
    else{  
        i = indexImg;  
    }  
});  
  
$('.prev').click(function(){ // image précédente  
    i--; // on décrémente le compteur  
    //puis on réalise la même chose que pour la fonction "suivante"  
    if( i >= 0 ){  
        $img.css('display', 'none');  
        $currentImg = $img.eq(i);  
        $currentImg.css('display', 'block');  
    }  
    else{  
        i = 0;  
    }  
});
```

Créer le défilement d'images automatique

Enfin, pour terminer, vous deviez créer un défilement d'images automatique. Nous vous conseillons d'utiliser la fonction `setTimeout()` couplée à une autre fonction pour créer une boucle infinie, et répéter le défilement. Celui-ci ne va que dans un sens, ce qui est logique : il défile vers la droite, c'est-à-dire qu'il affiche à chaque fois l'image suivante. Rien de bien compliqué : il suffit d'incrémenter le compteur. Seulement, il convenait de faire attention à ce qu'il ne dépasse pas l'index de la dernière image, en le remettant à 0 le cas échéant.

```
function slideImg(){  
    setTimeout(function(){ // on utilise une fonction anonyme  
        if( i < indexImg ){ // si le compteur est inférieur au dernier index  
            i++; // on l'incrémente  
        }  
        else { // sinon, on le remet à 0 (première image)  
            i = 0;  
        }  
        $img.css('display', 'none');  
        $currentImg = $img.eq(i);  
        $currentImg.css('display', 'block');  
        slideImg(); // on oublie pas de relancer la fonction à la fin  
    }, 2000);  
}
```

```

    }, 4000); // on définit l'intervalle à 4000 millisecondes (7s)
  }

  slideImg(); // enfin, on lance la fonction une première fois

```

Et voilà, le carrousel est terminé !

```

$(document).ready(function(){
  var $carrousel = $('#carrousel') ; // on cible le bloc du carrousel
  $img = $('#carrousel img') ; // on cible les images contenues dans le carrousel
  indexImg = $img.length - 1 ; // on définit l'index du dernier élément
  i = 0 ; // on initialise un compteur
  $currentImg = $img.eq(i); // enfin, on cible l'image courante, qui possède l'index i (0 pour l'instant)
  $img.css('display', 'none'); // on cache les images
  $currentImg.css('display', 'block'); // on affiche seulement l'image courante
  $carrousel.append('<div class="controls"> <span class="prev">Précédent</span>
    <span class="next">Suivant</span> </div>');

  $('.next').click(function(){ // image suivante
    i++; // on incrémente le compteur
    if (i <= indexImg){
      $img.css('display', 'none'); // on cache les images
      $currentImg = $img.eq(i); // on définit la nouvelle image
      $currentImg.css('display', 'block'); // puis on l'affiche
    }
    else{
      i = indexImg;
    }
  });

  $('.prev').click(function(){ // image précédente
    i--; // on décrémente le compteur, puis on réalise la même chose que pour la fonction "suivante"
    if (i >= 0){
      $img.css('display', 'none');
      $currentImg = $img.eq(i);
      $currentImg.css('display', 'block');
    }
    else{
      i = 0;
    }
  });

  function slideImg(){
    setTimeout(function(){ // on utilise une fonction anonyme
      if (i < indexImg){ // si le compteur est inférieur au dernier index
        i++; // on l'incrémente
      }
      else { // sinon, on le remet à 0 (première image)
        i = 0;
      }
      $img.css('display', 'none');
      $currentImg = $img.eq(i);
      $currentImg.css('display', 'block');
      slideImg(); // on oublie pas de relancer la fonction à la fin
    }, 4000); // on définit l'intervalle à 4000 millisecondes (4s)
  }

```

```
slideImg(); // enfin, on lance la fonction une première fois  
});
```

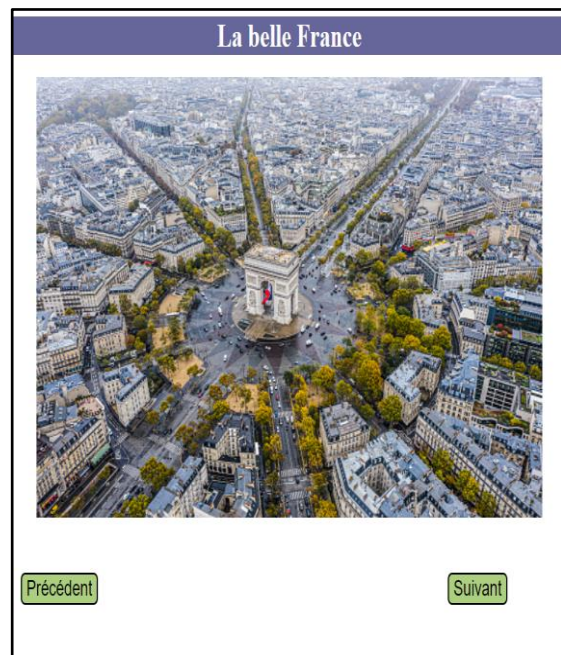
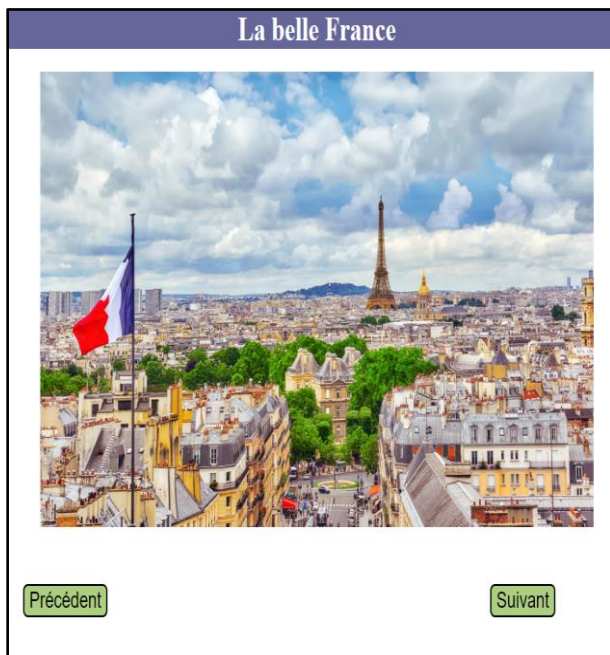
Améliorations

Nous venons de vous présenter un carrousel très basique. La plupart de ceux que vous rencontrerez sur la toile sont beaucoup plus sophistiqués. Voici quelques propositions :

- Le défilement d'images n'est pas très esthétique : à la place du système de display, essayez de mettre en place des effets d'animation sympathiques !
- Pour le moment, les contrôles sont très basiques : il n'est pas possible d'aller à une image précise en cliquant une seule fois. Que diriez-vous de réaliser une liste de boutons représentant chacun une diapositive ?
- De même, il est possible de faire des miniatures des images et de les afficher en dessous du carrousel pour faire une bien meilleure navigation !

Les possibilités sont infinies, on peut toujours trouver de nouvelles idées, qu'elles soient bonnes ou non. N'hésitez pas à regarder comment certains carrousel¹ sont fait, vous risquez d'avoir de fabuleuses surprises !

Notre résultat souhaité:



¹ <https://openclassrooms.com/fr/courses/1567926-un-site-web-dynamique-avec-jquery/1569419-tp-le-carrousel>