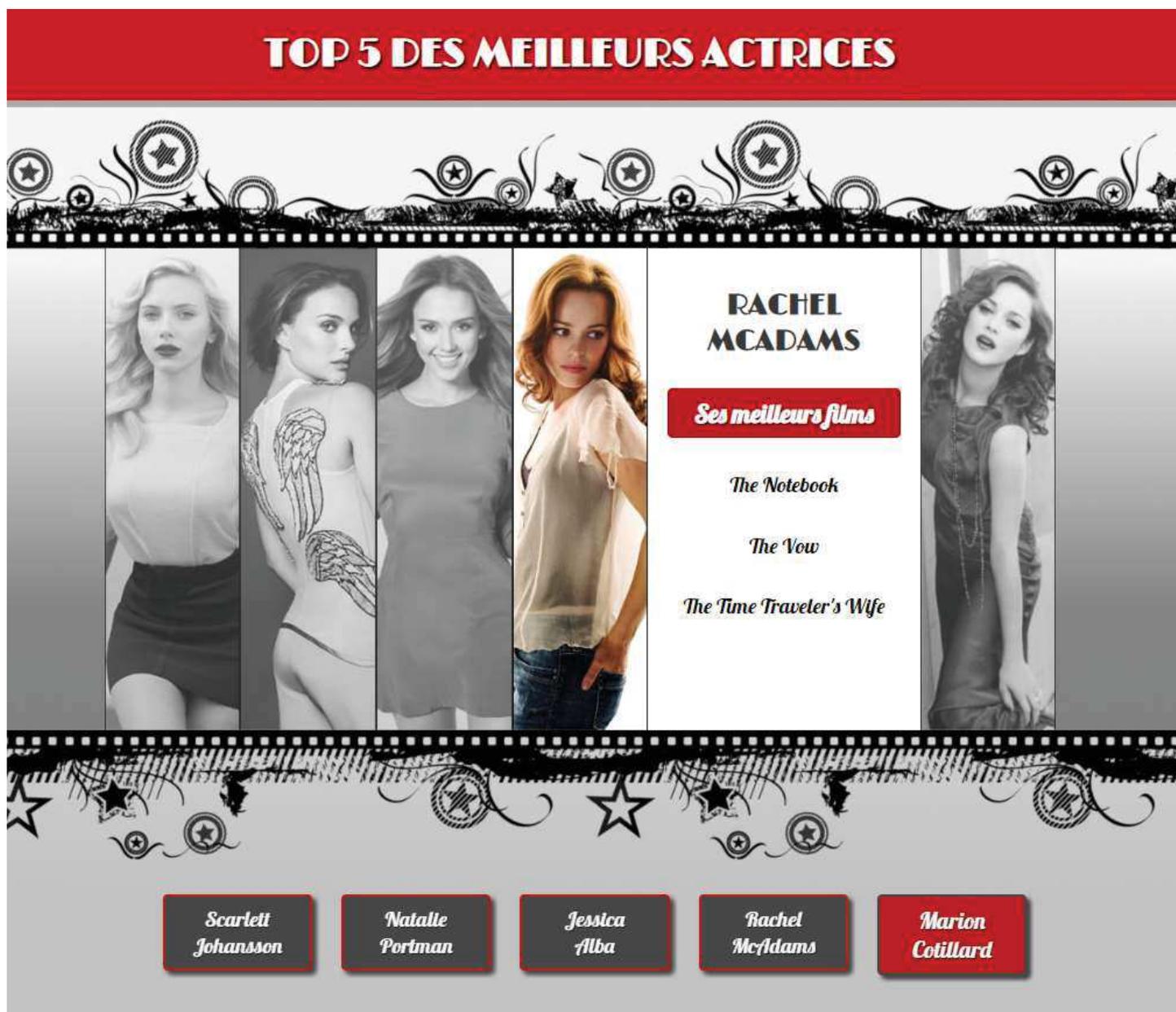


Développement :

Il s'agit d'une page web interactive : en cliquant sur l'une des photo des actrices, un volet s'ouvre avec son nom et 3 de ses films les plus marquants. A tester ici : <http://meilleuresactrices.lencodage.fr/>



Cadre :

Il s'agit d'un TP proposé dans le cadre du cours " Apprendre jQuery et créer une page web interactive", pdans le cadre de la formation " Formation Complète Développeur Web " proposée par Udemy.

Support :

HTML/CSS, Javascript avec la bibliothèque jQuery, sur l'IDE NetBeans.

Contraintes / Organisation :

Les images nous étaient fournies, ainsi que les instructions pas à pas pour le développement.

Difficultés rencontrées :

Il ne faut pas se perdre : il y a des fonction() {} à l'intérieur desquelles on trouve d'autres fonctions, à l'intérieur desquelles on en trouve d'autres encore. Oubliez un ";" quelque part, et ça ne fonctionne pas, fatalement.

Améliorations proposées :

Quand on clique sur une actrice, les titres de ses films apparaissent les uns après les autres.

Description résumée du développement

J'ai suivi les étapes suivantes pour le développement :

- Création des éléments
- Changer les polices
- Ouverture et fermeture
- Ouverture de départ
- Ajout des boutons
- Animation des boutons
- Animation des titres de films

1. Création des éléments :

La partie Html est facile à résumer : pour chaque actrice on met une division couleur et une division noir et blanc, pour les photos :

```
<ul>
  <li>
    <div id="part1" class="images">
      <div class="bw"></div>
      <div class="color"></div>
    </div>
    <div class="description">
      <h2>Scarlett<br>Johansson</h2>
      <h3>Ses meilleurs films</h3>
      <p>Lucy</p>
      <p>The Island</p>
      <p>Lost In Translation</p>
    </div>
  </li>
</ul>
```

En réalité, il n'y a qu'une photo des actrices en noir et blanc, et une photo en couleurs :



Alors comment ça marche ?

Chaque actrice est dans un rectangle de 140 pixels de largeur, et 500 pixels de hauteur.

Donc en réalité, on superpose les 2 images, et c'est avec le Css qu'on affiche seulement l'actrice choisie : on voit bien que l'image couleur commence à 140px à partir de la gauche, alors la noir et blanc commence à 0 :

```

.bw {
  position: absolute;
  left: 0px;
  width: 140px;
  height: 500px;
  cursor: pointer;
  background: url("images/bw-image.jpg") no-repeat;
  opacity: 0.7;
}

.color {
  position: absolute;
  left: 140px;
  width: 140px;
  height: 500px;
  cursor: pointer;
  background: url("images/color-image.jpg") no-repeat;
}

```

Pour avoir l'actrice suivante, il suffit de décaler de 140 px :

```

#part2 div{
  background-position: -140px 0px;
}

```

Et ainsi de suite pour les suivantes.

2. Changer les polices :

Pour trouver de jolies polices, Google est notre ami : <https://fonts.google.com/>

A l'heure où j'écris, il y a 988 polices disponibles.

Le plus long est donc de choisir.....ensuite, google vous fournit : le lien à insérer dans l'Html, ainsi que la façon de l'utiliser en Css :

Il ne reste plus qu'à copier/coller le lien dans le <head> du document Html :

```

<link href="https://fonts.googleapis.com/css?family=Limelight&display=swap"
      rel="stylesheet">
<link href="https://fonts.googleapis.com/css?family=Lobster&display=swap"
      rel="stylesheet">

```

Ici nous avons choisi la police LimeLight pour le titre de la page et le nom des actrices, et Lobster pour les titres des films.

3. Ouverture et fermeture

C'est à ce moment-là qu'on entre (enfin !) dans le vif du sujet : les animations.

Il faut donc : connaître le n° de l'index sur lequel l'utilisateur a cliqué, puis l'image couleur de l'actrice remplace l'image noir et blanc, et le cadre de description apparaît à sa droite.

J'ai beaucoup documenté mon code, je vous mets donc les extraits sans autre commentaire :

```
/*
 * tous les li qui sont dans des ul de main-menu
 */
var $mainMenuItems = $("#main-menu ul").children("li");
/*
 * le nbre total de "li"
 */
totalMainMenuItems = $mainMenuItems.length,

/*
 * fonction qui vérifie que l'index est compris entre 0 et le nbre d'index
 */
validIndex = function(indexToCheck) {
    return (indexToCheck >= 0 && indexToCheck < totalMainMenuItems);
},

/*
 * animation des volets
 * @param {$item} c'est l'index du volet à animer
 * @param {toOpen} booléen, true si on veut ouvrir le volet, false sinon
 * @param {speed} vitesse de l'animation, en millisecondes
 */
animateItem = function($item, toOpen, speed) {
    // on sélectionne l'image couleurs
    var $colorImage = $item.find(".color");
    // si toOpen est vrai, alors la description s'affiche, sinon sa taille
    // est réduite pour disparaître sous l'image suivante
    itemParam = toOpen ? {width: "420px"} : {width: "140px"},
    // si on ouvre l'image, alors la couleur est par-dessus le n&b
    colorImageParam = toOpen ? {left: "0px"} : {left: "140px"};

    // on anime l'image couleur avec les paramètres ci-dessus
    $colorImage.animate(colorImageParam, speed);
    // et on anime la division "description" avec les paramètres ci-dessus

/*
 * Fonction qui regroupe les fonctions validIndex et animateItem
 * @param {indexToCheckAndAnimate} int, l'index concerné
 */
checkAndAnimateItem = function(indexToCheckAndAnimate) {
    // si l'index à fermer = l'index à vérifier
    if (openedIndex === indexToCheckAndAnimate) {
        // alors on ferme l'item
        animateItem($mainMenuItems.eq(indexToCheckAndAnimate), false, 250);
        // et on remet l'index à -1
        openedIndex = -1;
    } else {
        // sinon si l'index à vérifier est valide
        if (validIndex(indexToCheckAndAnimate)) {
            // on ferme l'index en cours
            animateItem($mainMenuItems.eq(openedIndex), false, 250);
            // on attribue le bon index
            openedIndex = indexToCheckAndAnimate;
            // et on l'ouvre
            animateItem($mainMenuItems.eq(openedIndex), true, 250);
        }
    }
}
}
```

4. Ouverture de départ

Au chargement de la page, je veux que ce soit Jessica Alba, au milieu, qui soit “ ouverte ”. Pour cela, il suffit d’indiquer un `openedIndex = 2` à l’initialisation,

```
openedIndex = 2,  
  
init = function() {  
    bindEvents();  
  
    if(validIndex(openedIndex)) {  
        // eq() sélectionne le n° du li dans l'ul du main-menu  
        animateItem($mainMenuItems.eq(openedIndex), true, 700);  
        fadeTitles();  
    }  
},
```

5. Ajout des boutons

Côté Html, ils ont été simplement créés à partir d’une liste :

```
<ul>  
    <li class='button'>Scarlett<br>Johansson</li>  
    <li class='button'>Natalie<br>Portman</li>  
    <li class='button'>Jessica<br>Alba</li>  
    <li class='button'>Rachel<br>McAdams</li>  
    <li class='button'>Marion<br>Cotillard</li>  
</ul>
```

et côté Css, on a créé une classe `.hovered` pour qu’ils changent de couleur quand le pointeur de la souris passe dessus. En réalité, on peut, directement en Css, gérer le comportement de l’affichage dans ce cas. Mais puisque c’est un TP jQuery, la classe créée a été ajoutée ainsi :

```
$(".button").hover(  
    function() {  
        // on lui ajoute une classe  
        $(this).addClass("hovered");  
    },  
    function() {  
        // ou on la retire  
        $(this).removeClass("hovered");  
    });
```

6. Animation des boutons

En cas de clic sur le bouton bien sûr, c’est le fonction `checkAndAnimate()` qui est appelée, afin que le comportement soit similaire à celui d’un clic sur la photo d’une actrice.

7. Animation des titres de films

Je voulais donc que les titres de films apparaissent progressivement, les uns après les autres :

TOP 5 DES MEILLEURS ACTRICES



Des fonctions call-back ont été utilisées : une première fonction, `hide()`, avec en 1^{er} argument la vitesse, et en 2^{ème} argument une fonction. Dans cette fonction, une fonction `fadeIn()` qui fait apparaître le titre, avec la vitesse en argument 1 et ... une fonction donc, en argument 2. Et ainsi de suite pour faire apparaître les 3 titres. De cette manière, les effets se font bien en cascade comme je le souhaitais.

```
/*
 * Fonction qui fait disparaître les titres, puis les affiche 1 par 1,
 * en cascade : opacité passe de 0 à 1 en 1 seconde
 */
fadeTitles = function() {
    // cache tous les titres (dans les balises <p>)
    $("p").hide("fast",function() {
        // affiche le 1er titre : c'est le 3ème élément de la <div>
        $("p:nth-child(3)").fadeIn(1000, function() {
            // affiche ensuite le 2ème titre : 4ème élt de la <div>
            $("p:nth-child(4)").fadeIn(1000, function() {
                // et enfin le 3ème titre
                $("p:nth-child(5)").fadeIn(1000);
            });
        });
    });
};
```