

M2202 – Algorithmique

TD 5 : Une bannière animée interactive

Nous verrons dans ce TD comment :

- enchaîner une suite d'images en fondu dans une bannière ;
- afficher une image spécifique de cette bannière par survol d'un lien.

Une bannière est un élément souvent utilisé dans les sites. Il permet d'afficher une série d'images dans une zone donnée, généralement située dans l'en-tête du site. Nous commencerons par créer une structure de site minimaliste permettant d'accueillir :

- un wrapper (div) qui permettra de centrer le site
- l'en-tête qui contiendra un logo et la bannière (les images seront créées en JS)
- une section avec article et liens/texte pour l'affichage d'une image donnée.

html – css

Cette page contiendra comme on l'a vu les éléments utiles à la structure prévue. Le style permettra (`site.css`) quant à lui de mettre en forme les éléments. On y insère également l'accès à la librairie jQuery (ici en relatif, mais on peut pour l'exercice utiliser celle hébergée sur le site de Google) ainsi qu'à notre propre page js (`site.js`).

[Créez la page html5..](#)

```
<!DOCTYPE html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Bannière animée</title>
<link rel="stylesheet" href="site.css" />
<script type="text/javascript" src="jquery-1.11.0.min.js"></script>
<script type="text/javascript" src="site.js"></script>
</head>
```

```

<body>
  <div id='wrapper'>
    <header id='entete'>
      <div id='logo' class='floatleft'>
        <img width='285px' src='duck.png' alt='logo'>
      </div>
      <div id='banniere' class='absolu'></div>
    </header>
    <section id='contenu'>
      <h1>Une banni&egrave;re anim&eacute;e et interactive JS - CSS2</h1>
      <article id='navi'>
        <a href="#" id='1'>Une cigogne.</a>
        <a href="#" id='2'>Des flamands roses.</a>
        <a href="#" id='3'>Un tigre.</a>
        <a href="#" id='4'>Un paon.</a>
        <a href="#" id='5'>Un zoziau.</a>
      </article>
    </section>
  </div>
</body>
</html>

```

[et la page CSS..](#)

```

@charset "utf-8";
/* CSS Document */

```

```

body{
  margin:0;
  font-family:Arial, Helvetica, sans-serif;
}

```

```
a{
    text-decoration:none;
    color:black;
}

#wrapper{/* centré larg. Fixe */
    width:960px;
    margin:auto;
}

#banniere{
    margin-left:285px;/* décalage / logo flottant à gauche */
    width:675px;
    height:285px;
}

#logo{/* hauteur imposée */
    height:285px;
}

#contenu{
    clear:left;/* le contenu ne doit pas glisser sous les flottants */
}

.absolu{
    position:absolute
}

.etire{/* l'élément occupe toute la place du contenant (??Déformation)
    width:100%;
    height:100%;
}
```

```
.floatleft{//Classe des flottants à gauche
    float:left;
}
```

Et jQuery

Nous reprenons la fonction `charge_image()` du TD précédent, les fonctions `change_image` et `$(document).ready()` sont modifiées pour prendre en compte les nouvelles fonctionnalités. Nous créons deux éléments `imgs` pour pouvoir réaliser le fondu de l'un par dessus l'autre. Les deux `img` ont donc un z-index de 1 et 2 respectivement. Lors du changement d'image :

1. on affecte la nouvelle source à l'`img` du dessous ;
2. on fait disparaître celle du dessus en fondu, ce qui révèle celle d'en-dessous dans une sorte de fondu enchaîné ;
3. Le fondu terminé, on inverse les z-index ;
4. Enfin on ramène l'opacité de l'image 'fondue'..

```
// JavaScript Document
```

```
//Déclaration de variables
var nb = 6;//nb d'images
var actuel = 0;//image courante
var imgsrep = 'images/';
var imgs = new Array();//tableau des images pour préchargement
var t; // objet timer
var img1, img2;// objets jQuery pour les images
var tempo = 5; // intervalle de changement d'image - en sec
var fondu = 1 ; // temps de fondu - en sec
```

```

//Création du tableau des sources
//Les images sont nommées image0.jpg, image1.jpg, ..
for (i=0;i<nb;i++){
    imgs.push(imgsrep + 'image' + i + '.jpg');
}

function charge_images(){
    //on créé un img jQuery virtuel dont on affecte la source -> provoque le chargement dans le
    cache
    $(imgs).each(function(){//pour chaque élément du tableau d'images -> pour chaque image
        $("<img/>").src = imgsrep + this; //on affecte à la source d'un img virtuel la valeur de
        l'élément(this)
    });
}

```

Dans la mesure où les méthodes `fadeIn`, `fadeOut` (comme `slideDown` et `slideUp`) s'exécutent en mode asynchrone (le code continue à s'exécuter), on utilise le paramètre *complete* de la méthode `fadeOut`, qui s'exécute lorsque le fondu s'est terminé !!

```

//fonction changement d'image
function change_image(a){// a sera l'id du lien survolé
    if (a>0){//si a positif
        actuel=a-1;// a-1 car incrémenté ensuite
    }
    actuel++; //on incrémente actuel(
    if(actuel===nb){actuel=0;} //Test de dépassement

    //Affectation de la nouvelle source à l'image du dessous
    if(img1.css('z-index')=='2'){//variables locales i1 et i2 pour plus de clareté
        i1 = img1;
        i2 = img2;
    }
}

```

```

else{
    i2 = img1;
    i1 = img2;
}
i2.attr('src', imgs[actuel]); // changement de source
i1.fadeOut(fondu*1000, function() { //Disparition en fondu PUIS
    i2.css('z-index', '2'); //Echange des z-index
    i1.css('z-index', '1');
    i1.fadeIn(100); //On peut réafficher l'image
}
);
}

```

Nous créons les deux objets images dans la méthode `$(document).ready()` en leur affectant les attributs nécessaires à leur manipulation, les deux éléments sont ensuite insérés dans la div 'banniere' au même endroit mais avec un empilement spécifique.

```

//installation de la bannière
$(document).ready(function() { //lorsque la page est chargée
    //préchargement des images
    charge_images();

    //création de 2 objets img
    //img1
    img1 = $('<img />', {
        src : imgs[1],
        class : 'etire',
        css : {
            display : 'block',
            position : 'absolute',
            zIndex : '1'
        }
    });
}

```

```

//img2
img2 = $('<img />', {
    src : imgs[0],
    class : 'etire',
    css : {
        display : 'block',
        position : 'absolute',
        zIndex : '2'
    }
});

//Insertion de l'élément dans la bannière
img1.appendTo('#banniere');
img2.appendTo('#banniere');

//Animation : Démarrage du timer
t = setInterval(change_image, tempo*1000);

//Survol des liens de l'article
$('#navi a').mouseenter(function(){
    //l'id renvoie à l'image -> à actuel
    change_image(parseInt($(this).attr('id')));
});

});

```

Un gestionnaire de survol est affecté aux liens appartenant à l'élément dont l'id vaut 'navi'. A l'entrée dans le lien, on passe l'id du lien (c'est en fait le numéro de l'image) converti en entier (pour pouvoir comparer et calculer) à la fonction `change_image`, ce qui modifie la valeur de la variable **actuel** et affiche l'image correspondante !!