

DES PAGES WEB EN MOUVEMENT

JQUERY, LES ANIMATIONS

DES PAGES WEB EN MOUVEMENT

- #La fonction animate en jQuery P.3
- #Fonctions fade et slide sur une balise P.4
- #Lancer une fonction a la fin de animate() P.5
- #Enchainer plusieurs animations P.6
- #Manipuler une animation avec stop() P.7
- #Utiliser la fil d’attente P.8

LA FONCTION ANIMATE EN JQUERY

L'animation en JavaScript peut être faite grâce aux propriétés CSS des balises, il suffit d'appeler la fonction animate() et de lui passer en paramètre les propriétés à modifier.

Fonction animate()

Permet d'animer un élément en passant en paramètre les propriétés CSS à animer

Animation dynamique

En utilisant les opérateurs JavaScript aux valeurs numériques, il est possible de modifier dynamiquement la propriété CSS

Les valeurs propres à JavaScript

En plus des valeurs classiques pour des propriétés CSS, il est possible d'utiliser les fonctions toggle(), show() ou hide() en valeurs pour les propriétés CSS



```
//Animation classique
$( '#anim1' ).animate({
  left: '250px',
  height: '150px',
  width: '150px'
});
```



```
//Augmentation dynamique
$( '#anim2' ).animate({
  left: '*=5px',
  height: '+=50px',
  width: '-=50px'
});
```



```
//Utiliser toggle, show, ou hide
$( '#anim3' ).animate({
  height: 'toggle',
  width: 'show',
  opacity: 'hide'
});
```

FONCTIONS FADE ET SLIDE SUR UNE BALISE

Certaines animations sont devenues tellement utilisées que jQuery à prévu des fonctions entières pour les réaliser et les mettre en place rapidement. Les fonctions `fadeIn()` et `fadeOut()` permettent d'afficher ou de désafficher des balises alors que les fonctions `slideUp()`, `slideDown()` ou `slideToggle()` permettent de les faire glisser.

Fonction `fadeIn()`

Permet d'afficher un élément en `display: none`

```
//Fonction fadeIn() et fadeOut()
$( '#anim1 button' ).click(function(){
    $( '#anim1 img' ).fadeIn( 'slow' );
});
```

Fonction `fadeOut()`

Permet de modifier un élément en `display: none`

```
$( '#anim2 button' ).click(function(){
    $( '#anim2 img' ).fadeOut( 'fast' );
});
```

Fonction `slideDown()`

Permet de faire glisser un élément du haut vers le bas

```
//Fonction slideDown(), slideUp() et slideToggle()
$( '#anim3 button' ).click(function(){
    $( '#anim3 img' ).slideDown( 200 );
});
```

Fonction `slideUp()`

Permet de faire glisser un élément du bas vers le haut

```
$( '#anim4 button' ).click(function(){
    $( '#anim4 img' ).slideUp( 1000 );
});
```

Fonction `slideToggle()`

Combine `slideDown()` et `slideUp()`

```
$( '#anim5 button' ).click(function(){
    $( '#anim5 img' ).slideToggle();
});
```

LANCER UNE FONCTION A LA FIN DE ANIMATE()

La fonction animate(), en plus de ces paramètres d'animations peut recevoir en option un paramètre de temps pour la durée de l'animation mais également une fonction entière qui sera exécutée à la fin de l'animation.

```
//Animation avec option de temps et fonction finale
$('#anim1').animate({
  left: '250px',
  height: '150px',
  width: '150px'
}, 2000, function(){
  //Code à exécuter à la fin de l'animation
  this.hide();
});
```

Durée de l'animation

Après l'animation

Le code JavaScript placé dans la fonction finale sera exécuter une fois l'animation finie

L'utilisation du mot-clé this fait référence à l'élément sélectionné en jQuery

ENCHAINNER PLUSIEURS ANIMATIONS

Pour réaliser une animation par étapes sur un élément HTML, il faut créer une fonction dans laquelle seront enchainées plusieurs fonctions animate() sur cet élément. Prenons pour exemple une balise <div> animée au click sur une balise #animation.

Sélection de la balise #animation

Etapas de l'animation

Pour créer des étapes il suffi d'inscrire un appel à la fonction animate() par ligne

```
$( '#animation' ).click(function(){
    var myDiv = $('div');
    myDiv.animate({height: '300px', opacity: '0.4'}, 100);
    myDiv.animate({width: '300px', opacity: '0.8'}, 200);
    myDiv.animate({height: '100px', opacity: '0.4'}, 100);
    myDiv.animate({width: '100px', opacity: '0.8'}, 400, function(){
        //Code exécuter à la fin de l'étapes
        console.log('L'animation est terminée');
    });
});
```

Comme pour une animation classique, il est possible d'ajouter une fonction à la fin de l'animation et une durée pour chaque étapes de l'animation.

MANIPULER UNE ANIMATION AVEC STOP()

La fonction stop() en jQuery permet de stopper une animation en cours, il suffi de sélectionner l'élément et de lui appliquer la fonction.

Fonction delay()

Permet d'ajouter un delai entre deux étapes de l'animation

```
//Lancer l'animation
$( '#playAnim' ).click( function() {
    $( '#imgOne' ).slideUp( 300 ).delay( 400 ).fadeIn( 400 );
    $( '#imgTwo' ).slideUp( 300 ).fadeIn( 400 );
} );
```

Fonction stop()

Permet de stopper l'animation en cours

```
//Stopper l'animation
$( "#stopAnim" ).click( function() {
    $( '#imgOne' ).stop();
    $( '#imgTwo' ).stop();
} );
```

UTILISER LA FIL D'ATTENTE

La fil d'attente (queue en anglais) permet de programmer des animations sur plusieurs étapes et de les associer à un éléments du DOM. Il est alors possible de créer plusieurs queues d'animation et de les déclencher avec la fonction dequeue().

Balise de stockage

Function queue()

Cette fonction est appelée pour chaque étapes de l'animation

Nom de la fil d'attente

Nommée la fil d'attente permet de l'authentifier précisément

Définition de l'animation

Dans le système de fil d'attente, les étapes d'animation sont des fonctions sur la balise à animer

Function next()

La fonction next() est indispensable car elle permet de préciser la fin d'une étape pour passer à la suivante

Function dequeue()

Fonction pour lancer la fil d'attente

```

$("section")
//Définition des étapes de l'animation

.queue('animationOne', function(next) {
    $("article").animate({left: '400px'});
    next();
})

.queue('animationOne', function(next) {
    $("article").animate({top: '400px'});
    next();
})

.queue('animationOne', function(next) {
    $("article").animate({left: '0px'});
    next();
})

.queue('animationOne', function(next) {
    $("article").animate({top: '0px'});
    next();
})

.dequeue('animationOne');
```