

Programmation langage objet

Création d'une page html

Site de référence :

<http://fr.openclassrooms.com/informatique/cours/un-site-web-dynamique-avec-jquery>
<http://fr.openclassrooms.com/informatique/cours/simplifiez-vos-developpements-javascript-avec-jquery>

1. Rappel structure d'une page html

1.1. Les balises

```
<!DOCTYPE html>
2<html>
3  <head>
4  <link rel="icon" type="image/png" href="icone.png" />
5    <meta charset="utf-8" />
6    <title>Titre</title>
7  </head>
8
9  <body>
10
11 </body>
12</html>
```

2. Le langage CSS

Les **CSS**, *Cascading Style Sheets* (feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML, par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus-dessous, etc.). Le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web.

On peut écrire du code en langage CSS à trois endroits différents :

- dans un fichier `.css` (*méthode la plus recommandée*) ;
- dans l'en-tête `<head>` du fichier HTML ;
- directement dans les balises du fichier HTML *via* un attribut `style` (*méthode la moins recommandée*). (Ex : `<div align="center">`)

C'est le contenu, `<link rel="stylesheet" href="style.css" />` : qui indique que le fichier HTML est associé à un fichier appelé `style.css` qui est chargé de la mise en forme.

Exemple : on veut que le texte du document html soit de couleur rouge

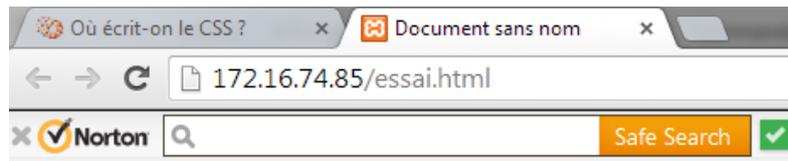
Dans le fichier html on mettra :

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="fichiercss.css" />
<title>Document sans nom</title>
</head>
<body >
  <p>couleur rouge</p>
</body>
</html>
```

Dans le fichier css on mettra :

```
@charset "utf-8";  
/* CSS Document */  
p  
{  
color: red;  
}
```

On obtient :



couleur rouge

3. Le langage javascript

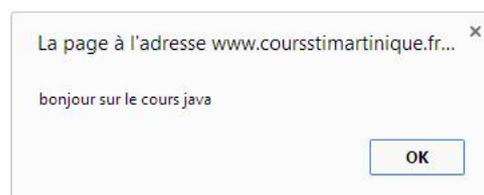
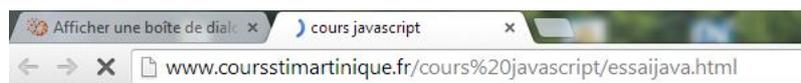
3.1. Présentation

Le JavaScript (souvent abrégé JS) est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets .

3.2. Introduction

```
<!doctype html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>cours javascript</title></head>  
<body>  
  
<script type="text/javascript">  
alert('bonjour sur le cours java');  
</script>  
  
</body>  
</html>
```

Code javascript



Dans le code HTML donné précédemment, on remarque quelques nouveautés.
Tout d'abord, un élément `<script>` est présent : c'est lui qui contient le code Javascript que voici :

```
alert('bonjour sur le cours java');
```

Il s'agit d'une instruction, c'est-à-dire une commande, un ordre, ou plutôt une action que l'ordinateur va devoir réaliser. Les langages de programmation sont constitués d'une suite d'instructions qui, mises bout à bout, permettent d'obtenir un programme ou un script complet.

Dans cet exemple, il n'y a qu'une instruction : l'appel de la fonction `alert()`.

`alert()` est une instruction simple, appelée **fonction**, qui permet d'afficher une boîte de dialogue contenant un message. Ce message est placé entre apostrophes, elles-mêmes placées entre les parenthèses de la fonction `alert()`.

3.3. Mise en place d'un script

Il existe deux méthodes pour intégrer du java script dans une page html :

- Dans la page (présentation ci-dessus)
- Externe : on fait appel à une page ayant l'extension `.js`

Ce fichier est ensuite appelé depuis la page Web au moyen de l'élément `<script>` et de son attribut `src` qui contient l'URL du fichier `.js`. Voici un exemple :

```
<body>  
<script src="http://mrdoob.github.com/three.js/build/three.min.js"></script>
```

4. Présentation langage JQUERY

C'est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant Ajax) et HTML, et a pour but de simplifier des commandes communes de JavaScript. La première version date de janvier 2006.

5. Introduction

Pour pouvoir utiliser la bibliothèque JQuery, il faut d'abord faire appel à une page javascript .

Il y a deux possibilités :

- Soit on télécharge le fichier sur le site et on l'enregistre sur le dossier source.

<http://jquery.com/download/>

On écrira :

```
<!doctype html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>cours jquery</title>  
</head>  
  
<body>  
<script src="jquery-1.8.3.min.js"></script>  
</body>  
</html>
```

- Soit on écrit le lien vers les ressources Google qui met à disposition de tous le framework jquery hébergé directement sur ses serveurs.

Le lien vers jquery Minified chez Google est

<https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js>

Et le lien vers jQuery Uncompressed chez Google est
<https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.js>

On écrira alors :

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>cours jquery</title>
</head>

<body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.js"></script>
</body>
</html>
```

6. La fonction jquery()

L'ensemble du framework jQuery repose sur cette fonction. C'est la fonction la plus importante du framework, on doit toujours l'utiliser dès qu'il s'agira de développer en jQuery.

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>cours jquery</title>
</head>

<body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.js"></script>
<script>
    jQuery(document).ready(function(){
        // Code en jQuery
    });
</script>

</body>
</html>
```

Remarque :

Pour simplifier l'écriture, on peut remplacer le mot jQuery par le symbole \$.
Donc la phrase : `jQuery(document).ready(function()` peut s'écrire `$(document).ready(function()`

Exercice :

- On veut afficher le message « salut »

Correction :

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>cours jquery</title>
```

```
</head>

<body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.js"></script>
<script>
  jQuery(document).ready(function(){
    alert('salut');
  });
</script>

</body>
</html>
```

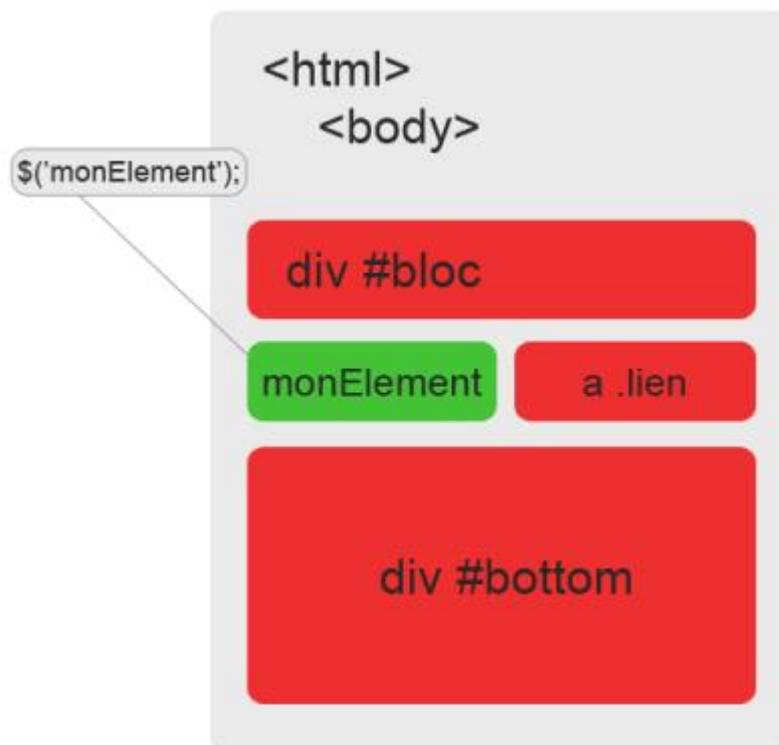
Remarque :

Pour agir avec les éléments d'une page web, on réalisera ce qu'on appelle couramment un « ciblage d'élément », qui représentera une balise, par exemple. On agira ensuite dessus grâce à des méthodes.

Exemple :

```
$('#monElement').maMethode();
```

On cible dans un premier temps l'élément qui jouera le rôle d'objet, grâce à un sélecteur, ensuite on agit dessus grâce à une méthode.



Une méthode en jQuery est une fonction un peu spéciale qui agira directement sur l'objet (ou l'élément ciblé, si vous préférez) indiqué. Il est possible de chaîner les méthodes, c'est-à-dire que l'on peut en mettre plusieurs à la suite, pour agir sur l'objet en une seule fois.

```
$('#monElement').premiereMethode().deuxiemeMethode();
```

La première méthode agit d'abord sur l'objet, avant la deuxième. On a donc un ordre logique : les méthodes s'enclenchent les unes derrière les autres.

Exemple :

- On veut afficher le message bonjour lorsqu'on clique sur une image

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>cours jquery</title>
</head>

<body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.js"></script>

<script>
$(document).ready(function(){
  $('#image').click(function(){
    alert('Bonjour');
  });
});
</script>
<div></div>
</body>
</html>
```

7. Les évènements

Comme JavaScript, on peut intégrer à des balises des évènements.

- On aura par exemple un évènement qui pourra écouter la souris de l'utilisateur : quand il va cliquer sur un certain élément de votre page que vous aurez ciblé avec un sélecteur, vous pourrez exécuter du code jQuery.

Exemple :

```
$('#image').click(function(){
  alert('Bonjour');
});
```

- Un autre évènement peut écouter le clavier de l'utilisateur : quand il va presser des touches de son clavier, vous pourrez exécuter là encore du code.

Exemple :

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>cours jquery</title>
</head>

<body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.js"></script>

<script>
$(document).ready(function(){
  $('#image').click(function(){
    alert('Cliqué !');
  });
});
$(document).keyup(function(touche){ // on écoute l'évènement keyup()
```

```

var appui = touche.which || touche.keyCode; // le code est compatible tous navigateurs grâce à ces deux propriétés
if(appui == 13){ // si le code de la touche est égal à 13 (Entrée)
    alert('Vous venez d'appuyer sur la touche Entrée !'); // on affiche une alerte
}
});

</script>
<div></div>
</body>
</html>

```

7.1. Le cas des formulaires

Les éléments de formulaire possèdent eux aussi leur lot d'évènements associés. Par exemple, lorsqu'un champ est focalisé, on dit qu'il « prend le *focus* », car c'est sur lui que se trouve notre pointeur. Un évènement permet de contrôler ce genre de chose.

Action	Fonction
Focalisation	focus()
Sélection (p.e. dans une liste)	select()
Changement de valeur	change()
Envoi du formulaire	submit()

Exemple :

- On veut récupérer la valeur dans un objet edit lorsqu'on change la valeur et on veut l'afficher.

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>cours jquery</title>
</head>

<body>
<label for="textfield">Ecrire un mot :</label>
<input type="text" name="textfield" id="texte">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.js"></script>

<script>
$(document).ready(function(){
$('#texte').change(function(){
var texte1=document.getElementById('texte').value;
alert(texte1);
});
});
</script>

</body>
</html>

```

7.2. Le déclenchement virtuel

jQuery permet de simuler le déclenchement d'évènements grâce à une simple méthode. Pour faire court, vous n'avez pas besoin d'attendre que l'utilisateur remplisse une action précise pour lancer du code : vous pouvez exécuter virtuellement un évènement grâce à `trigger()` ! Il suffit de donner le type de l'évènement en tant qu'argument.

```
1 $('p').click(function(){
2   alert('Cliqué !');
3 });
4
5 $('p').trigger('click'); // déclenche l'action au chargement du script
```

7.3. Annuler le comportement d'un élément par défaut

Chaque évènement possède son propre groupe d'éléments spécifiques à traiter : par exemple, la soumission de formulaire ne s'applique pas à tous les cas. Cependant, certains éléments ont un comportement par défaut, défini par le navigateur. Le cas le plus courant est le lien hypertexte : son comportement va être de rediriger l'utilisateur vers l'adresse donnée.

Une méthode en jQuery permet d'annuler tous comportement par défaut. Il s'agit de `preventDefault()`.

```
1 $('a').click(function(e){
2   e.preventDefault(); // annule l'action du lien
3 });
```

8. Listes des évènements possibles

8.1. L'écoute sur la souris

Action	Fonction
Clic	<code>click()</code>
Double-clic	<code>dblclick()</code>
Passage de la souris	<code>hover()</code>
Rentrer dans un élément	<code>mouseenter()</code>
Quitter un élément	<code>mouseleave()</code>
Presser un bouton de la souris	<code>mousedown()</code>
Relâcher un bouton de la souris	<code>mouseup()</code>
Scroller (utiliser la roulette)	<code>scroll()</code>

8.2. L'écoute sur le clavier

Tout comme la souris, le clavier peut tout à fait être écouté par un script jQuery. L'utilisation la plus fréquente que l'on peut en faire est de détecter l'appui d'une touche et de lancer une action.

Les évènements associés ne sont pas très nombreux, on en trouvera seulement trois :

- `keydown()`, qui se lance lorsqu'une touche du clavier est enfoncée ;
- `keypress()`, qui se lance lorsqu'on maintient une touche enfoncée ;
- `keyup()`, qui se lance lorsqu'on relâche une touche préalablement enfoncée.

9. Modifier et connaître les caractéristiques des éléments

9.1. Connaître les propriétés des éléments

Exemple :

On veut récupérer la valeur d'un slider dont l'id est `slider1` dans une variable `position`.

```
<div><input type="range" id="slider1" name="left-image" step="10" value="8" min="8" max="58" />
</div>
```

On va écrire :

```
var position=$('#slider1').val();
```

Explication:

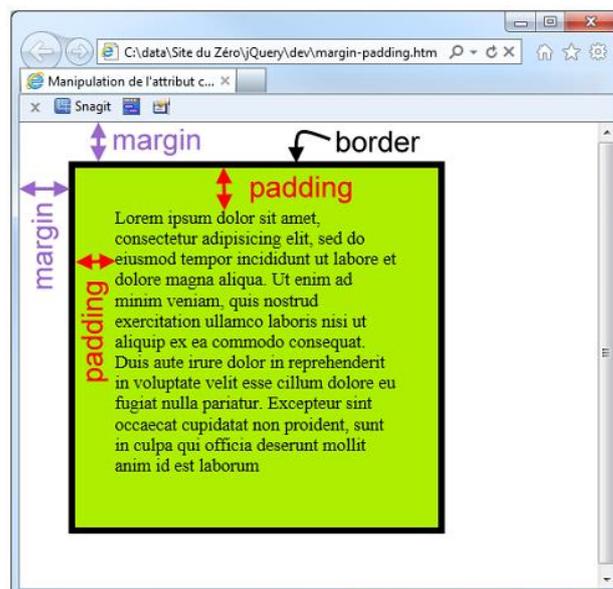
On trouve deux éléments :

- `$('#slider1')` : l'ID de l'élément
- `val()` : propriété de l'élément

9.1.1. Liste des propriétés

9.1.1.1. Dimensions et de redimensionner les éléments de type block :

- `width()` : largeur de l'élément, de la fenêtre ou du document, sans inclure les marges (`padding`, `border` et `margin`). Cette méthode peut être utilisée comme `getter` (pour connaître la largeur d'un élément) ou comme `setter` (pour modifier la largeur d'un élément).



- `innerWidth()` : largeur de l'élément, en incluant le `padding` gauche et droit.
- `outerWidth()` : largeur de l'élément, en incluant le `padding` gauche et droit et `border`.

- `outerWidth(true)` : largeur de l'élément, en incluant padding gauche et droit, border et margin gauche et droit.
- `height()` : hauteur de l'élément, de la fenêtre ou du document, sans inclure les marges (padding, border et margin). Cette méthode peut être utilisée comme getter (pour connaître la hauteur d'un élément) ou comme setter (pour modifier la hauteur d'un élément).
- `innerHeight()` : hauteur de l'élément, en incluant le padding supérieur et inférieur.
- `outerHeight()` : hauteur de l'élément, en incluant border et padding supérieur et inférieur.
- `outerHeight(true)` : hauteur de l'élément, en incluant border, padding supérieur et inférieur et margin supérieur et inférieur.

9.1.1.2. Position

Pour gérer la position des éléments dans une page HTML, on utilisera les méthodes suivantes :

- `offset()` : position absolue d'un élément dans la page ;
- `position()` : position relative d'un élément dans son parent .

Les positions retournées par ces méthodes ont deux composantes : l'abscisse left et l'ordonnée top. Vous utiliserez donc :

- `offset().left` et `offset().top` pour connaître la position absolue d'un élément.
- `position().left` et `position().top` pour connaître la position d'un élément dans son parent.

9.2. Modifier les propriétés des éléments

Exemple :

On veut modifier la valeur d'un label avec comme id « valeurcurseur » avec une variable positionleft

```
<div>
<input type="text" id="valeurcurseur" value="1">
</div>
```

On va écrire :

```
$('#valeurcurseur').val(positionleft);
```

Deuxième possibilité : on veut modifier les paramètres d'un paragraphe

```
$('#p').css({'color':'red'});
```

9.3. Exercice

Cahier des charges :

- On veut récupérer la position d'un slider dans un label
- On veut pouvoir déplacer une image à droite et à gauche avec deux boutons



Correction:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>cours jquery</title>
<style type="text/css">
#divimage{
    position:absolute;
    top:350px;
    left:20px;
}
</style>
</head>

<body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.js"></script>
<script>
$(document).ready(function(){

$('#bdroite').click(function(){
    var posdiv = $('#divimage').offset();
    posdiv.top = 350;
    posdiv.left = (posdiv.left)+100;
    $('#divimage').offset(posdiv);
    $('#valeurposition').val(posdiv.left);
});
$('#bgauche').click(function(){
    var posdiv = $('#divimage').offset();
    posdiv.top = 350;
    posdiv.left = (posdiv.left)-100;
    $('#divimage').offset(posdiv);
    $('#valeurposition').val(posdiv.left);
```

```

});
$('#slider').change(function(){
    var positionslider=$('#slider').val();
    $('#valeurcurseur').val(positionslider);
});
});

</script>

<div id="divimage">

</div>
<h4>Déplacer curseur </h4>
<div><input type="range" id="slider" name="left-image" step="10" value="0" min="0" max="50" />
</div>
<h4>Valeur curseur</h4>
<div>
<input type="text" id="valeurcurseur" value="0">
</div>
</div>
<h4>Position image (left)</h4>
<div>
<input type="text" id="valeurposition" value="20">
</div>
<p>
<input type="submit" name="submit" id="bgauche" value="gauche">
<input type="submit" name="submit" id="bdroite" value="droite">
</p>
</body>
</html>

```

9.4. Lire ou les modifier les valeurs textuelles

- text() retourne/modifie la valeur textuelle stockée dans l'élément ;
- html() retourne/modifie le code HTML stocké dans l'élément.

Exemple :

On veut récupérer le premier mot d'une en-tête4 dans une variable premier

```
var premier=$('#h4:first').text()
```

Exemple d'instruction :

Instructions	Résultat
alert (\$('#p').html ());	Affiche le code HTML du premier élément (voir image suivante).
alert (\$('#p:first').html ());	Affiche le code HTML du premier élément.
alert (\$('#p:last').html ());	Affiche le code HTML du dernier

	élément.
<pre> \$('p').html(function(index,actuel) { alert('Paragraphe ' + (index+1) + ' : '+actuel);}); </pre>	Affiche individuellement le code HTML de chaque élément.

10. Cas particuliers

10.1. Fermer et ouvrir une balise

- Fermer une balise

On utilisera la fonction **hide**. (**duration** [, **easing**] [, **complete**])

Caractéristiques:

duration (default: 400)

Type: Number or String

Une chaîne ou un nombre à déterminer combien de temps l'animation se déroulera.

easing (default: swing)

Type: String

Une chaîne indiquant le type d'animation : linear etc.

complete

Type: Function()

Une fonction à appeler une fois l'animation terminée.

- Ouvrir une balise

On utilisera la fonction **show**. (**duration** [, **easing**] [, **complete**])

Caractéristiques:

duration (default: 400)

Type: Number or String

Une chaîne ou un nombre à déterminer combien de temps l'animation se déroulera.

easing (default: swing)

Type: String

Une chaîne indiquant le type d'animation : linear etc.

complete

Type: Function()

Une fonction à appeler une fois l'animation terminée.

Exercice :

- On veut fermer et ouvrir une div avec deux boutons

Corriger :

Fichier html

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```

<meta charset="utf-8">
<title>fermeture et ouverture d'une balise</title>
<link href="fichiercss.css" rel="stylesheet" type="text/css">
</head>

<body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.js"></script>
<script type='text/javascript'>
$(document).ready(function(){
    //fermer
    $('#fermer').click(function(){
        $('#div1').hide('slow',function({});
    });
    //ouvrir
    $('#ouvrir').click(function(){
        $('#div1').show('slow',function({});
    });
});
</script>
<div id="div1">
    <center><p>bonjour</p>
    <p>ouverture et fermeture d'une div</p>
    <p>clicquer sur le bouton correspondant</p>
</center>
</div>
<p>&nbsp;</p>
<p>
    <input type="button" id="ouvrir" value="ouvrir">
    <input type="button" id="fermer" value="fermer">
</p>
</body>
</html>

```

Fichier css :

```

@charset "utf-8";
/* CSS Document */
#div1{
    width: 200px;
    height: 150px;
    background-color: #F1090D;
}

```

<http://www.coursstimartinique.fr/exercice%20final/ouvrir%20et%20fermer%20une%20fenetre/index.html>

10.2. Animation

L'objectif est de réaliser une animation fluide sur une propriété CSS lors d'un évènement. Pour cela nous allons utiliser la fonction animate().

- Caractéristiques :

.animate(properties [, duration] [, easing] [, complete])

Type: [PlainObject](#)

duration (default: 400)

Type: Number or String

Une chaîne ou un nombre à déterminer combien de temps l'animation se déroulera.

easing (default: swing)

Type: String

Une chaîne indiquant le type d'animation : linear etc.

complete

Type: Function()

Une fonction à appeler une fois l'animation terminée.

- Remarque :

Toutes les propriétés animées doivent être animés d'une **seule valeur numérique**.

La plupart des propriétés qui sont non-numérique ne peuvent pas être animés en utilisant les fonctionnalités jQuery de base (par exemple, la largeur, la hauteur, ou à gauche peuvent être animés mais background-color ne peut pas être, à moins que le `jQuery.Color ()` plugin est utilisé. Les valeurs de propriété sont traitées comme un nombre de pixels, sauf indication contraire. Les unités em et % peuvent être spécifiées le cas échéant.

En plus des propriétés de style, certaines propriétés non-style telles que scrollTop et scrollLeft , ainsi que des propriétés personnalisées, peuvent être animés.

Propriétés CSS sténographie (par exemple la police, fond, frontière) ne sont pas entièrement pris en charge. Si vous souhaitez animer taille de la police, vous devez utiliser `fontSize` ou l'équivalent CSS `'font-size'` plutôt que `'police'`.

11. Exercice :

11.1. Lorsqu'on clique sur le bouton on veut que la couleur de fond passe du rouge au vert et à la fin de l'animation le bouton se déplace et s'élargis.

Récupérer le plugin <http://code.jquery.com/color/jquery.color-2.1.2.js> ou

<http://www.coursstimartinique.fr/jquery/animation%20couleur/jquery.color-2.1.2.js> avant

11.2. Correction

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>exercice animate</title>
<link href="couleur.css" rel="stylesheet" type="text/css">
</head>

<body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.js"></script>
<script type="text/javascript" src="jquery.color-2.1.2.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$('#bouton').click(function(e){
//changement de couleur
$('#bouton').animate({backgroundColor:'green'},5000,function() {
$('#bouton').animate({width:'300',left:'100'},500,"linear",function() {
});
});
});
});
```

```
});  
});  
});  
</script>  
<div id="bouton"></div>  
</body>  
</html>
```

<http://www.coursstimartinique.fr/jquery/animation%20couleur/couleur.html>

Fichier CSS :

```
@charset "utf-8";  
/* CSS Document */  
#bouton{  
    position:relative;  
    width: 200px;  
    height: 150px;  
    border-radius: 10px;  
    background-color: red;  
    left:10px;  
}
```

11.3. Exercice2 :

Lorsqu'on appuie sur un bouton on veut faire apparaître une image et lorsqu'on appuie sur un autre bouton, on veut faire disparaître l'image.

Avant :



Après :



11.4. Correction fichier html

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>fermeture et ouverture d'une balise</title>
<link href="fichiercss.css" rel="stylesheet" type="text/css">
</head>

<body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    //apparaître l'image du chat
    $('#ouvrir').click(function(){
        $('#div1').animate({opacity:0},1000,"linear",function() {});

        $('#chat').animate({opacity:1},1000,"linear",function() {});
    });

    //cacher l'image du chat
    $('#fermer').click(function(){
        $('#div1').animate({opacity:1},1000,"linear",function() {});

        $('#chat').animate({opacity:0},1000,"linear",function() {});
    });
});
</script>
<div id="div1">
    <center><p>bonjour</p>
    <p>ouverture et fermeture d'une div</p>
    <p>clicquer sur le bouton pour faire apparaître l'image du chat</p>
</center>
</div>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p id="p1">
    <input type="button" id="ouvrir" value="apparaître">
    <input type="button" id="fermer" value="cacher">
</p>

</body>
</html>
```

Fichier CSS :

```
@charset "utf-8";
/* CSS Document */
#div1{
    top:0px;
    left:0px;
    width: 300px;
```

```

    height: 225px;
    background-color: #F1090D;
    opacity: 1;
    z-index:10;
    position:absolute;
}
#chat{
    top:0px;
    left:0px;
    position:absolute;
    z-index:9;
    opacity:0;
}

```

<http://www.coursstimartinique.fr/exercice%20final/ouvrir%20et%20fermer%20une%20fenetre/transparence.html>

14. Rotation

- Animation circulaire

L'objectif est de faire tourner une image autour d'un axe en donnant un effet de perspective.

Pour cela on va utiliser le plugin circulate. Dans un premier temps il faut installer le plugin jquery easing pour utiliser les différentes fonctions « easing ».

<http://gsgd.co.uk/sandbox/jquery/easing/jquery.easing.1.3.js>

- Propriétés

Vous pouvez paramétrer quelques options de base comme la vitesse d'animation, la hauteur et la largeur de la rotation. Il en existe trois autres :

sizeAdjustment : valeur d'ajustement (%). Par défaut à 100%. Ce paramètre permet de changer la taille de l'élément HTML pour jouer sur l'effet de profondeur.

loop : animation en boucle (true ou false). Par défaut à false. Vous pouvez arrêter l'animation en appelant la fonction jQuery stop() sur l'élément HTML.

zIndexValue : tableau de quatre valeurs numériques qui définissent la propriété CSS z-index des quatre points de l'animation

- Exercice :
 - On veut faire tourner une boule autour d'un élément en cliquant dessus :
- Correction :

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Document sans nom</title>
<link href="rotationplanetecss.css" rel="stylesheet" type="text/css">
</head>

<body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.js"></script>
<script type='text/javascript' src="jquery.easing.1.3.js"></script>
<script type='text/javascript' src="jquery.circulate.js"></script>
<script type='text/javascript'>
$(document).ready(function(){
    $('#planete').click(function(e){
        $('#planete').css("left", $('#planete').position().left).circulate({
            sizeAdjustment: 20,

```

```

    speed: 3000,
    width: 800,
    height: 300,
    loop: true,
    zIndexValues: [9, -50, 50, 9]
  });
});
});
</script>


</body>
</html>

```

Fichier CSS :

```

@charset "utf-8";
/* CSS Document */
#icone{
    position:absolute;
    top:200px;
    left:300px;
    z-index:9;
}
#planete{
    position:absolute;
    top:130px;
    left:400px;
    z-index:9;
}

```

<http://www.coursstimartinique.fr/exercice%20final/rotation%20planete/rotation%20planete.html>

- Animation par rapport au centre le l'objet
L'objectif est de faire tourner une image autour de son centre .
Pour cela on va utiliser le plugin rotate.

<http://www.coursstimartinique.fr/exercice%20final/jQueryRotate.js>

Son utilisation est très simple:

```

$(document).ready(function(){
$('#image').rotate(20);
});

```

Il suffit d'indiquer l'angle.

Réalisation d'une animation :

```

rotate({
angle:0,    angle de départ
animateTo:-90, angle de fin d'animation
duration: 2000, durée de l'animation
center: ["0px", "0px"], coordonnées du centre de rotation

```

Remarque : rotation en continue

```
var rotation = function (){
  $("#image").rotate({
    angle:0,
    animateTo:360,
    center: ["150px", "100px"],
    callback: rotation
  });
}
rotation();
```

Exemple :

<http://www.coursstimartinique.fr/exercice%20final/rideau/rideau.html>

<http://www.coursstimartinique.fr/exercice%20final/boite/boite.html>

Remarque : ouvrir le rideau avant de faire tourner le ventilateur

- **Autres types de plugin jquery**

<https://plugins.jquery.com/>