

1 Introduction à jQuery

De l'action dans vos pages Web

Je vais peut-être trouver quelque chose là-dedans pour rendre mes pages Web plus interactives.



Vous voulez améliorer vos pages Web. Vous "maîtrisez" déjà HTML et les CSS et vous souhaitez vous former aux scripts, mais vous ne voulez pas passer votre vie à écrire du code. Vous avez besoin d'une bibliothèque de scripts qui vous permette de modifier vos pages Web à la volée et tant qu'à rêver, il faudrait aussi qu'elle soit compatible avec AJAX et PHP. Est-ce que cette bibliothèque peut faire en 3 lignes ce que la plupart des langages côté client font en 15 ? Non, vous ne rêvez pas ! Il vous suffit de passer à jQuery.

Appel prof

Pour télécharger le répertoire du TP.

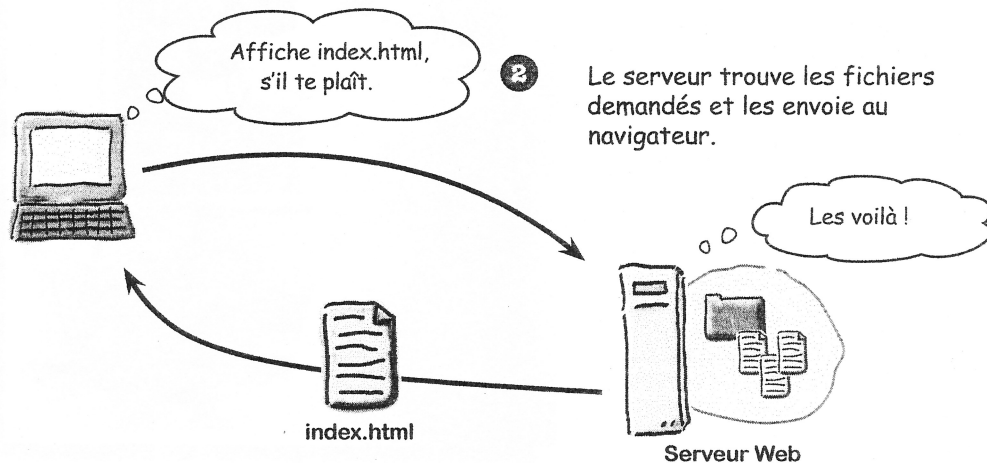
HTML et les CSS, c'est bien joli, mais...

Statique

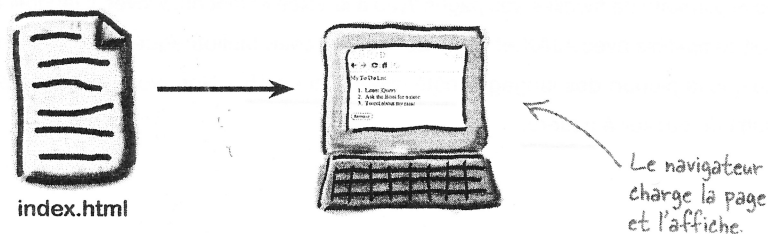
Le bon vieil HTML et les CSS sont utiles pour structurer la page et la styler. Une fois que vous avez affiché une page HTML, le problème est qu'elle est statique.

Que se passe-t-il si vous voulez modifier l'aspect de la page, ou bien ajouter ou supprimer un élément ? Vous devez faire une gymnastique d'enfer avec vos CSS, ou bien tout bêtement recharger une nouvelle page. Et cela peut vite dégénérer. Pourquoi ? Parce que HTML et les CSS ne servent qu'à contrôler la manière dont la page s'affiche.

- 1 Le navigateur demande l'affichage d'une page Web à un serveur quand on saisit l'URL dans la barre d'adresse du navigateur.



- 3 Le navigateur affiche la page HTML en se basant sur le fichier envoyé par le serveur.



on s'en tient au *script*

...vous avez besoin de la puissance des scripts

Pour modifier vos pages Web **à la volée**, sans les recharger, vous devez donner des instructions à votre navigateur, à l'aide d'une balise HTML appelée `<script>`.

Script



`<script>`
`</script>`

La balise `<script>` est placée dans le fichier HTML.



index.html

Hé, le navigateur, j'ai besoin de toi !

Je vois qu'il y a une balise `<script>` là, alors j'attends les prochaines instructions.

La balise `<script>` dit au navigateur que vous allez donner des ordres.



Mais comment je donne des ordres au navigateur ? Ça paraît bizarre...

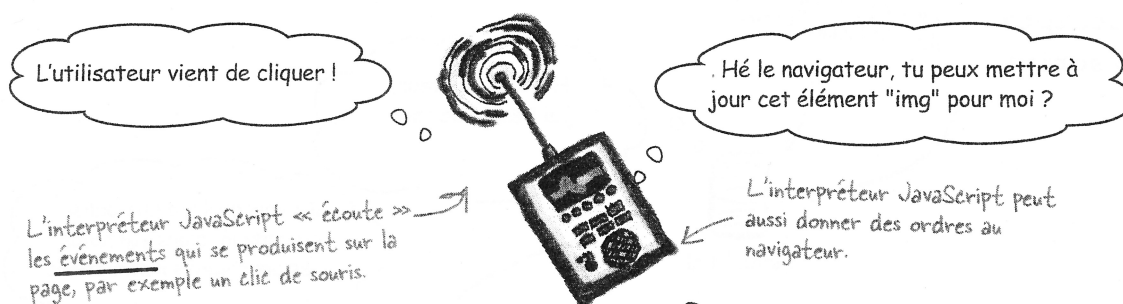


Bonne question ! Rappelez-vous que le HTML est un langage de balisage qui gère la structure des documents.

Et que les CSS (Cascading Style Sheets) contrôlent l'**aspect** et la **position** de ces éléments. Le HTML et les CSS contrôlent la construction et l'affichage, mais ils ne peuvent pas **rendre active** une page Web. Pour cela, on a besoin d'un **langage de script** comme jQuery.

jQuery (et JavaScript) entrent en scène !

Le langage que l'on utilise pour donner des ordres au navigateur s'appelle JavaScript. Chaque navigateur contient un interpréteur JavaScript intégré qui lit les instructions rédigées entre les balises `<script>` et les traduit en différents types d'actions sur la page Web.

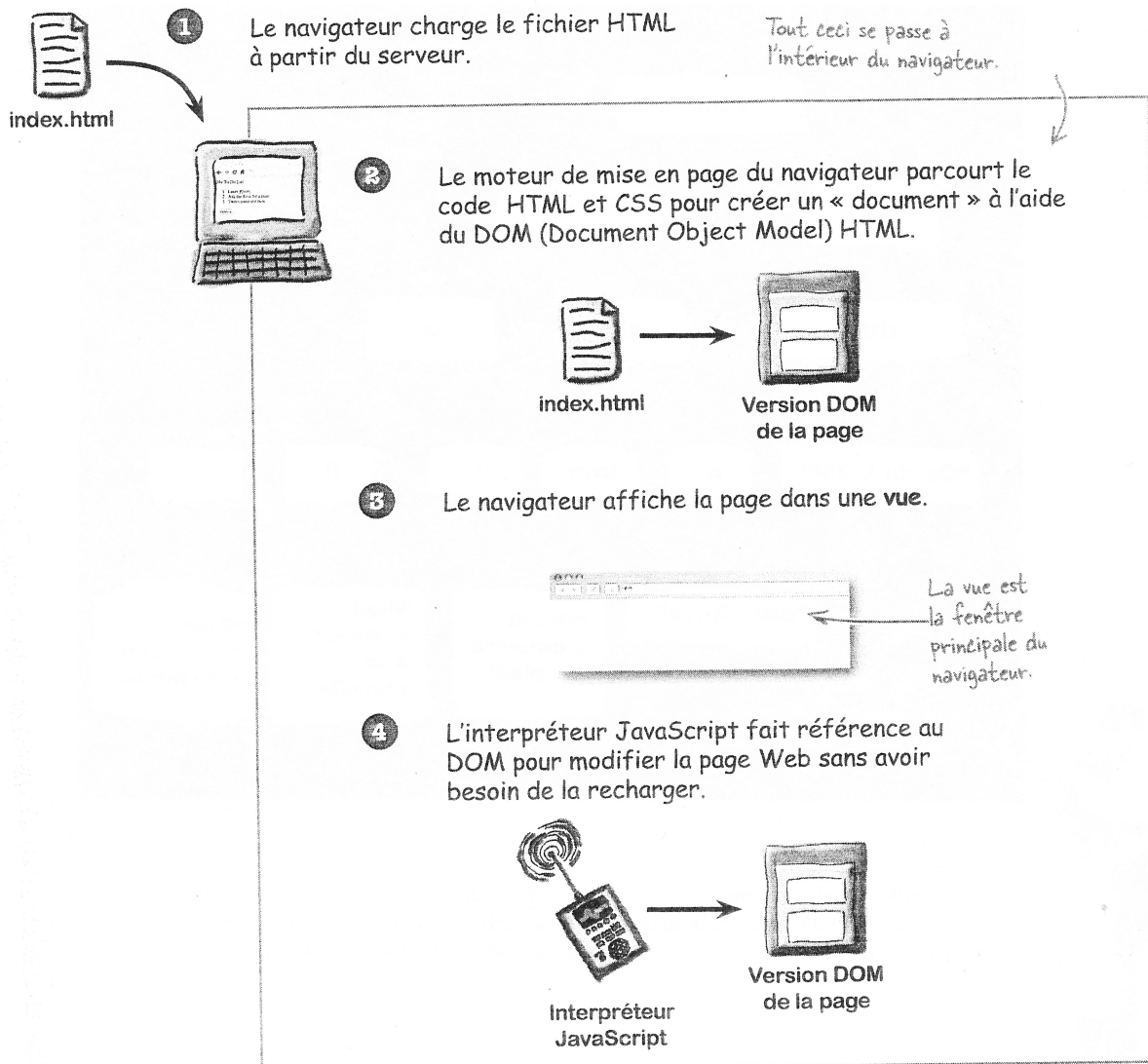


Pour donner des ordres à l'interpréteur, il faut lui parler en JavaScript. Mais, pas de panique ! C'est là où intervient jQuery. jQuery est une **bibliothèque** JavaScript spécialisée dans la modification des pages Web à la volée. Voyons voir à quoi ressemble jQuery...

jQuery est une bibliothèque JavaScript spécialisée dans la modification des pages Web à la volée.

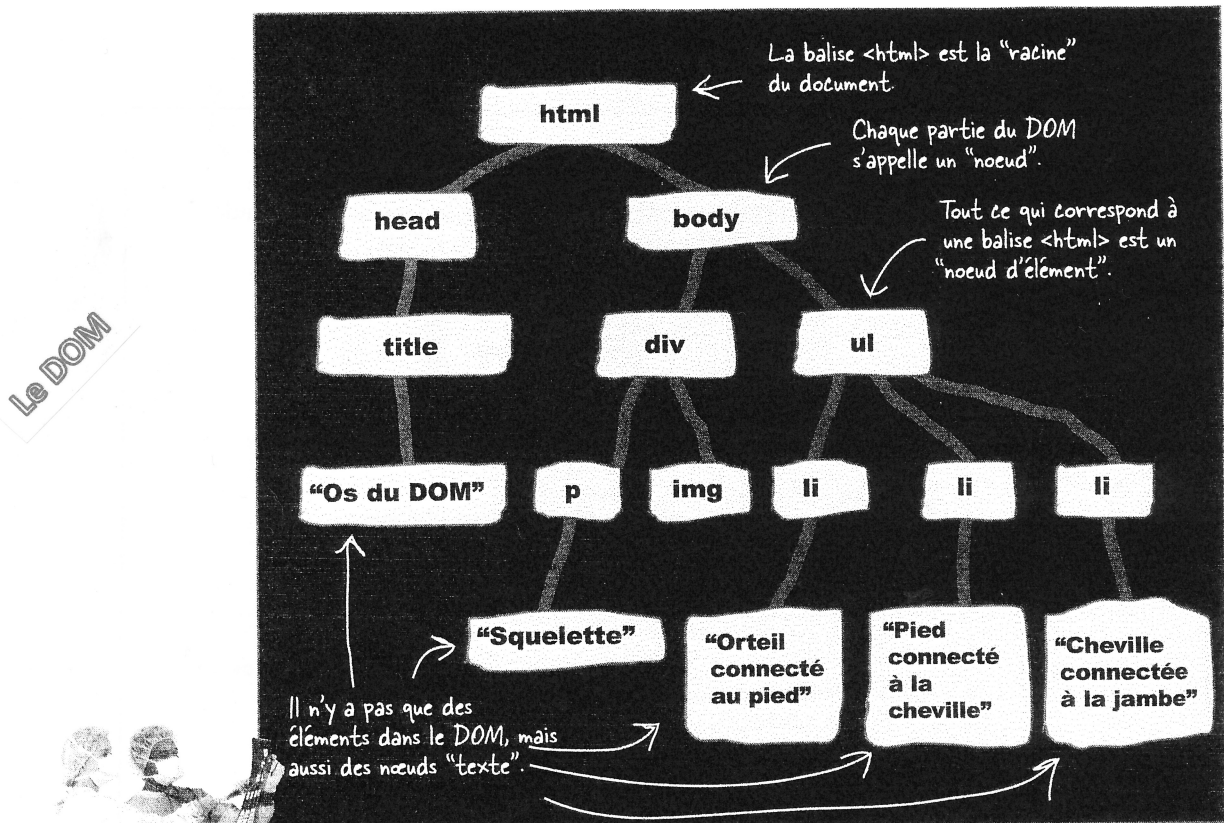
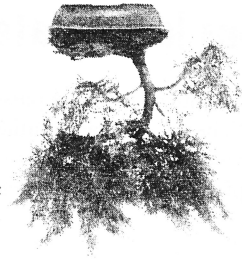
Examinons le navigateur

Il est temps de découvrir ce qui se passe vraiment derrière une page Web quand un navigateur l'affiche. Le navigateur utilise le DOM (Document Object Model) HTML pour créer une page à partir de balisage HTML et de code CSS ; le tout formera une page cliquable complète composée de textes, d'images, de vidéos et d'autres contenus.



Structure secrète d'une page Web

Au fil des ans, le DOM a favorisé une collaboration plus efficace entre HTML, CSS et JavaScript. Il fournit un squelette standardisé que tous les navigateurs utilisent pour faciliter le surf sur le Web. Beaucoup de gens se représentent le DOM comme un arbre : il a une *racine* et des *branches* dotées de *nœuds* à leur extrémité. Vous pouvez aussi penser que le DOM effectue une radiographie de la construction de la page.



Une radio renseigne le médecin sur la structure cachée du corps humain. Comme une radio, le DOM montre la structure qui se cache derrière la page. Mais à la différence d'une radio, JavaScript et jQuery utilisent le DOM pour *changer la structure* de la page.



Javascript et jQuery utilisent le **DOM** pour changer la **STRUCTURE** de la page.

jQuery permet d'apprivoiser le DOM

Le DOM peut paraître complexe et vous intimider, mais heureusement pour nous, jQuery en simplifie l'approche. N'oubliez pas que jQuery *c'est du JavaScript*, mais dans une version beaucoup plus accessible. Quand on veut contrôler le DOM, jQuery facilite la vie. Prenons un exemple : nous voulons modifier le code HTML à l'intérieur **du seul** élément paragraph de notre page.

Méthode JavaScript pure et dure

Je parle au document (le D de DOM).
 Récupère-moi tous les éléments qui ont la balise « p ».
 document.getElementsByTagName("p")
 [0].innerHTML = "Change la page.";
 Je veux l'élément 0.
 Définis le HTML de cet élément...
 ...avec cette valeur.

Méthode jQuery

Donne-moi un élément "paragraph".
 Modifie le HTML de cet élément avec ce qu'il y a entre parenthèses.
 \$("p").html("Change la page.");
 jQuery utilise un « moteur de sélection », ce qui veut dire que l'on peut modifier des données avec des sélecteurs comme avec les CSS.

Et maintenant, si l'on veut changer le code HTML de **cinq** éléments paragraph de notre page :

On fait une boucle sur les éléments que l'on veut modifier.
 for (i = 0; i <= 4; i++)
 {
 document.getElementsByTagName("p")
 [i].innerHTML="Change la page";
 }
 Récupère-moi l'élément sur lequel on boucle.

Comme jQuery utilise des sélecteurs CSS, on peut utiliser la même syntaxe que ci-dessus.
 \$("p").html("Change la page.");

Ça vous dirait d'embarquer avec jQuery pour aller faire un tour au pays du DOM ?

Activité 1 : Analyse de code



**Code Prêt
à l'emploi**

Saisissez le code suivant dans un éditeur de texte. Enregistrez-le et ouvrez-le dans votre navigateur, puis testez chacun des boutons (ça ne vous fera pas de mal de jeter un coup d'œil au code et d'essayer de trouver ce qu'il fait pendant que vous y êtes...).

Utilisez le projet
du répertoire
DOMVille !

```
<!DOCTYPE html>
<html><head> <title>jQuery goes to DOM-ville</title>
<style>
    #change_me {
        position: absolute;
        top: 100px;
        left: 400px;
        font: 24px arial;}

    #move_up #move_down #color #disappear {
        padding: 5px;}
</style>
<script src="scripts/jquery-X.X.X.min.js"></script>
</head>
<body>
    <button id="move_up">Move Up</button>
    <button id="move_down">Move Down</button>
    <button id="color">Change Color</button>
    <button id="disappear">Disappear/Re-appear</button>

    <div id="change_me">Make Me Do Stuff!</div>
<script>
    $(document).ready(function() {
        $("#move_up").click( function() {
            $("#change_me").animate({top:30},200);
        }); //end move_up
        $("#move_down").click( function() {
            $("#change_me").animate({top:500},2000);
        }); //end move_down
        $("#color").click( function() {
            $("#change_me").css("color", "purple");
        }); //end color
        $("#disappear").click( function() {
            $("#change_me").toggle("slow");
        }); //end disappear
    }); //end doc ready
</script>
</body>
</html>
```



index.html

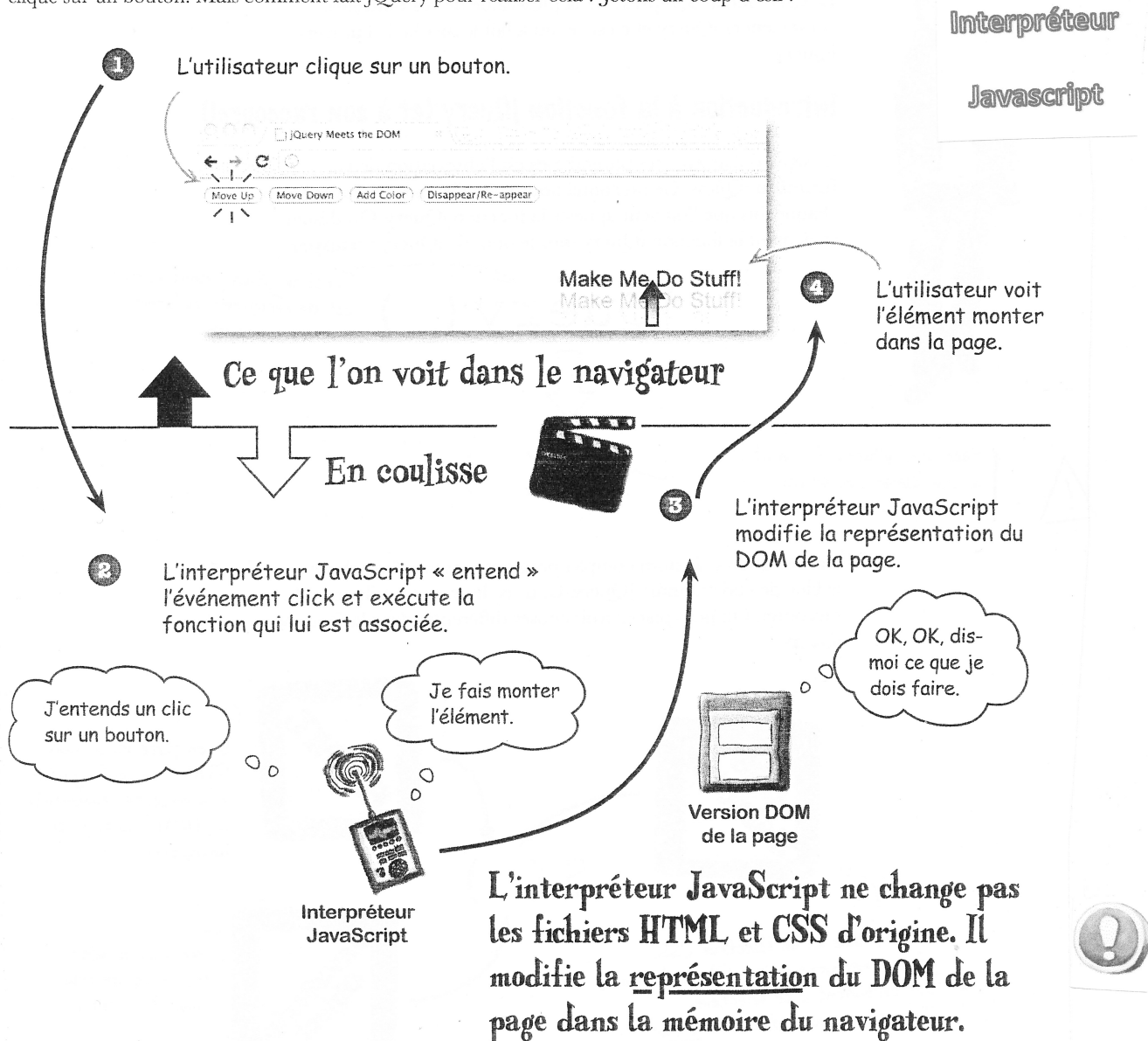


Lorsque vous avez saisi et testé le code ci-dessus complétez le DR1

X.X.X version du fichier dans le répertoire scripts.

Comment ça marche ?

Plutôt classe la manière dont jQuery arrive à manipuler la page, hein ? Ce qu'il faut retenir d'important, c'est que le **code original HTML et CSS n'est en rien modifié** quand on clique sur un bouton. Mais comment fait jQuery pour réaliser cela ? Jetons un coup d'œil :



C'est quoi tous ces symboles dollar dans le code ?



Le signe dollar (\$) représente la plus-value de vos nouvelles compétences jQuery et c'est ce qui a fait le succès de l'univers jQuery.

Introduction à la fonction jQuery (et à son raccourci)

Le signe dollar avec ses parenthèses est l'abréviation de la **fonction** jQuery. Ce raccourci nous évite d'écrire « `jQuery()` » chaque fois que l'on veut appeler la fonction jQuery. On désigne également la fonction jQuery sous le nom de jQuery **wrapper**.

jQuery ()

Fonction jQuery dont le but est de récupérer les éléments entre parenthèses.



→ \$ ()

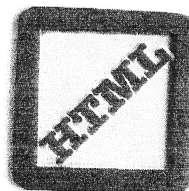


Raccourci jQuery. On saisit un seul caractère au lieu d'en taper six.

L'abréviation et le nom complet pointent vers la même chose : le bloc de code nommé jQuery. Dans ce livre, nous utiliserons le raccourci. On peut placer trois choses différentes dans la fonction jQuery.



Si l'on place ici un sélecteur CSS, jQuery retourne le jeu d'éléments qui correspondent à ce sélecteur. C'est ce que vous utiliserez le plus.



Si on place ici du code HTML, on peut ajouter à la volée des éléments du DOM à la page du navigateur.

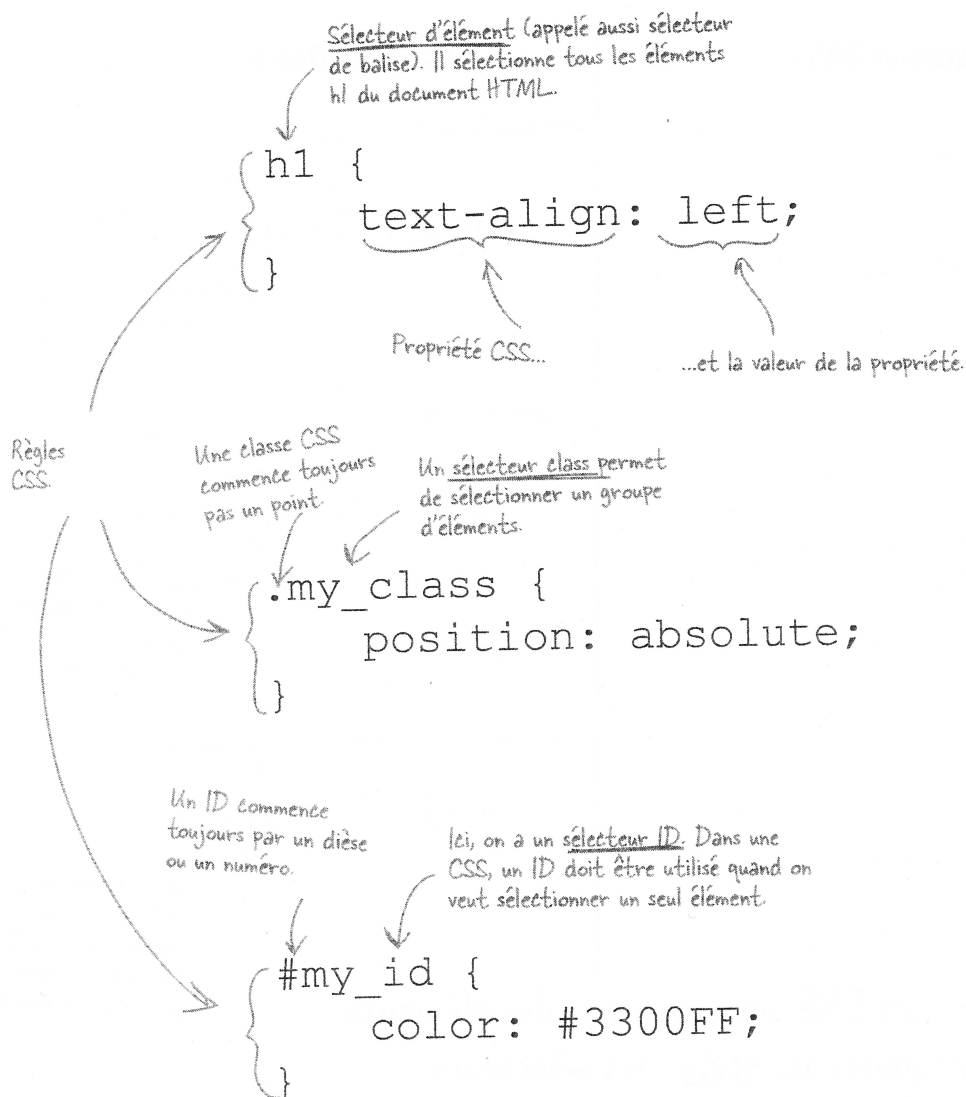


Ne vous occupez pas trop de cela pour l'instant.

jQuery sélectionne les éléments de la même manière que CSS

Sélecteurs

Vous en savez déjà plus sur jQuery que vous ne le pensez. On travaille principalement avec jQuery en employant des **sélecteurs** (les mêmes sélecteurs que vous utilisez dans vos CSS). Si vous n'êtes pas très au clair avec les sélecteurs CSS, nous allons faire une petite révision.



Les styles à la rencontre des scripts

Ce qu'il y a de bien avec jQuery, c'est qu'il utilise les mêmes sélecteurs CSS que ceux que l'on emploie pour styler la page afin d'en *manipuler les éléments*.

Sélecteurs d'élément

Sélecteur CSS

Sélecteur d'élément

```
h1 {  
    text-align: left;  
}
```

Sélecteurs class

Sélecteur class

```
.my_class {  
    position: absolute;  
}
```

Sélecteurs ID

Sélecteur ID

```
#my_id {  
    color: #3300FF;  
};
```

Sélecteur jQuery

Sélecteur d'élément jQuery

↓
Méthode
\$("h1").hide();

Masque tous les éléments h1 de la page.

Sélecteur class jQuery

↓
Méthode
\$(".my_class").slideUp();

Fait monter tous les éléments qui sont membres de la classe CSS my_class

Sélecteur ID jQuery

↓
Méthode
\$("#my_id").fadeOut();

Cette instruction jQuery fait disparaître un élément qui a un CSS ID égal à my_id jusqu'à ce qu'il devienne invisible.

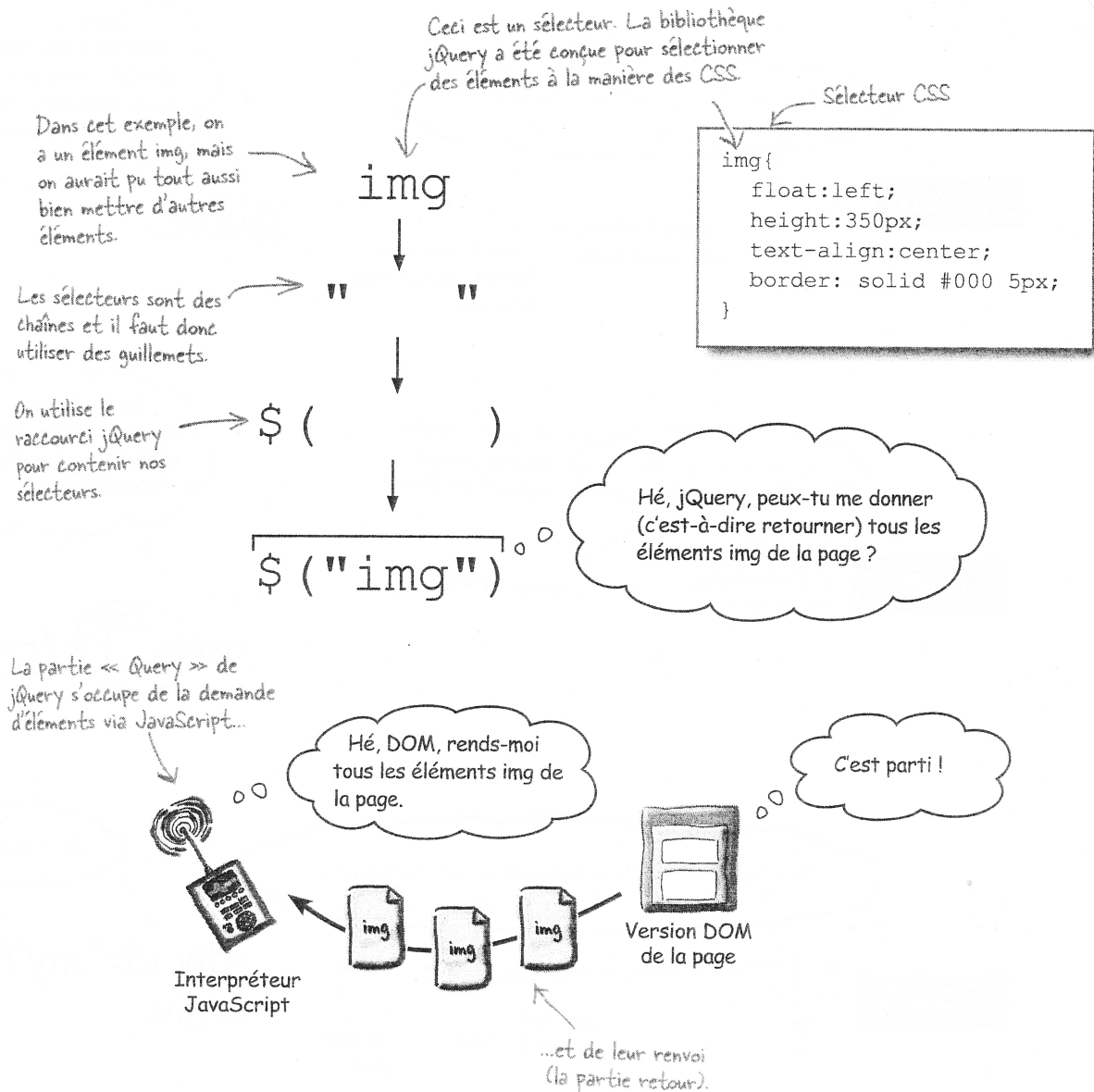
Les sélecteurs CSS sélectionnent des éléments pour leur ajouter un style ; les sélecteurs jQuery sélectionnent des éléments pour leur ajouter un comportement.



Sélecteurs jQuery

Comme son nom l'implique, jQuery concerne les **requêtes**. On demande quelque chose à l'aide d'un sélecteur et l'interpréteur JavaScript demande au DOM d'exécuter la requête. Si vous demandez un élément comportant des éléments imbriqués, jQuery fournit aussi les éléments imbriqués. Nous allons isoler un sélecteur jQuery pour étudier en détail son fonctionnement.

Requêtes



parlez-vous jQuery ?

Traducteur jQuery

Afin de vous montrer comme il est facile d'apprendre jQuery, voici le détail de quelques expressions jQuery que vous utiliserez quand vous voyagez au royaume du DOM.

`$("button").click(function() {})`



Hé, les éléments
button...

...quand l'utilisateur
vous clique dessus...

...je veux que vous
fassiez quelque chose
pour moi.

Quand un
utilisateur me clique
dessus, j'exécute
toutes les instructions
jQuery entre crochets.

Click Me!

`$("p").hide;`



Hé, les éléments
p (c'est-à-dire
paragraph)...

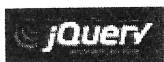
Un point-virgule
termine chaque
instruction jQuery.

...devenez invisibles.

Le texte qui
apparaît entre chaque
élément paragraph va
disparaître.

`<p>...</p>`

`$("#myTop").css({"background-color": "blue"});`



Hé, l'élément
avec l'ID
myTop...

...définis ta règle CSS...

...pour que la
couleur de fond...

...devienne
bleue.

Je vire au bleu.

`<div id="myTop">
</div>`

Il n'y a pas de questions idiotes

Q : Pourquoi avoir créé jQuery s'il ne fait qu'utiliser JavaScript ? Est-ce que JavaScript n'est pas suffisant ?

R : JavaScript est fantastique pour un tas de choses (notamment la manipulation du DOM), mais il est assez complexe. La manipulation du DOM au niveau le plus bas ne va vraiment pas de soi, et c'est là qu'intervient jQuery. Il masque une bonne partie de la complexité du DOM et facilite grandement la création d'effets. jQuery a été conçu par John Resig et vous en apprendrez plus sur lui à l'adresse suivante : <http://ejohn.org/about>.

Q : C'est quoi toute cette histoire autour du symbole dollar ?

R : C'est juste un raccourci pour que vous n'ayez pas à écrire « jQuery » sans arrêt ! Mais quand vous travaillez avec d'autres langages côté client, l'utilisation du libellé « `jQuery()` » évite les conflits de nommage.

Q : Vous venez de parler de « langage côté client ». C'est quoi exactement ?

R : Les développeurs Web désignent souvent le navigateur comme un *client* car il consomme des données issues d'un serveur Web. Un langage de script côté client est un langage qui donne des ordres au navigateur en coulisse, alors qu'un langage côté serveur donne des instructions au serveur.

Q : Quelle est l'origine du DOM ?

R : Bonne question ! Les développeurs et les créateurs de pages Web se sont lassés des incohérences entre les navigateurs et ont émis le besoin d'un standard qu'ils pourraient utiliser pour communiquer avec les pages Web de tout navigateur. Le Consortium World Wide Web (W3C) a travaillé de manière collaborative à la définition du standard et vous pouvez obtenir des détails là-dessus à <http://w3.org/dom>.

Q : Quand je veux télécharger jQuery, il y a une version de production et une version développeur. Quelle est la différence entre les deux ?

R : La version de production est *réduite*, ce qui signifie que toute une série de caractères inutiles et d'espaces ont été supprimés. Elle est optimisée pour s'exécuter plus vite dans un environnement de production, mais il est plus difficile de voir ce qui s'y passe. La version développeur, qui est bien plus lisible, est conçue pour ceux qui veulent farfouiller dans le code jQuery pour le modifier ou même l'étendre (après tout, c'est de l'open source !).



Activité 2 : Mini-projet

Votre première prestation jQuery

Vous venez de décrocher un poste de développeur Web pour la fondation SOS Animaux. L'équipe commerciale veut démarrer sa campagne annuelle de collecte de fonds par le relooking de la page « Aidons nos amis à poils ». Ils vous ont donné une copie d'écran de la page de l'année dernière avec les détails de ce qu'ils veulent modifier.

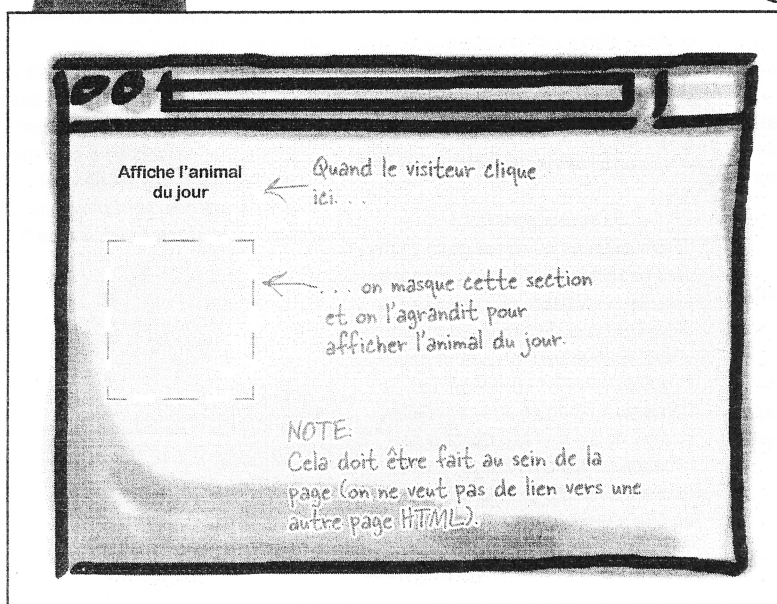


Josh du service commercial veut plus d'interactivité.

Et son chef veut des effets visuels plus riches.

Nous devons dynamiser la page de l'année dernière. À l'heure actuelle, le visiteur clique sur un bouton et une image apparaît, mais elle ne reste pas sur la page. On veut que l'image apparaisse quand l'utilisateur clique et disparaisse quand il clique à nouveau.

L'image se contente d'apparaître. Est-ce qu'elle peut glisser lentement dans une sorte de fondu ?



Personne n'est assez fou pour laisser tomber les gens du service commercial le premier jour car il faut se les mettre dans la poche ! Alors nous allons retrousser nos manches...

AVANT

Avant de voir comment ajouter des fonctionnalités jQuery à la page, examinons la manière dont le code HTML et CSS a été rédigé (les fichiers de l'année dernière sont listés ci-dessous).

```
<!DOCTYPE html><html> <head>
<title>Furry Friends Campaign: jQuery
Proof-of-Concept</title>
<link rel="stylesheet" type="text/
css" href="styles/my_style.css">
</head>
<body>
<div id="showfriend">
<a href="#">Our Furry Friends Need
Your Help

</a>
</div>
```



index.html

Cette balise d'ancrage a les états « hover » et « active » définis dans la CSS. L'utilisateur survole le lien et l'image apparaît.

L'image du chien est imbriquée dans une balise d'ancrage. Cette image ne doit pas apparaître tant que l'utilisateur n'a pas cliqué sur le lien de la balise d'ancrage.

```
a:link img, a:visited img {
display:none;
}
a:hover img, a:active img {
display:block;
}
a{
text-decoration:none;
color: #000;
}
```



my_style.css

Ce sélecteur CSS modifie la propriété display de l'image imbriquée en « none » de telle sorte qu'elle n'est plus visible quand la page est chargée la première fois.

Quand l'utilisateur survole avec sa souris ou clique sur la balise d'ancrage, la propriété display de l'élément img est changée en « block ». L'image apparaît ensuite d'un seul coup.



Bon, est-ce qu'on peut maintenant se lancer et commencer à coder en jQuery toutes les fonctionnalités demandées ?

C'est possible, mais vous risquez de compliquer les choses.

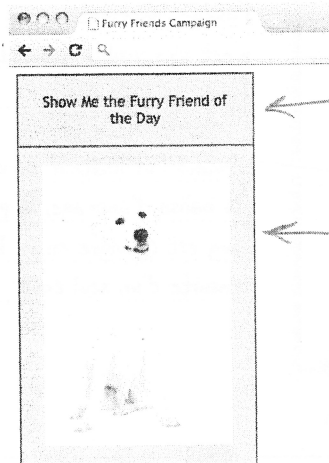
Avant d'utiliser jQuery pour réaliser les effets voulus par les commerciaux, on doit s'assurer qu'il dispose de tout ce qu'il faut pour faire des miracles. Comme vous le savez déjà, une des tâches principales de jQuery consiste à manipuler des éléments HTML, si bien qu'il faut avoir une bonne structure. Pour traiter les éléments, jQuery utilise les mêmes sélecteurs que CSS, de telle sorte que nous devons aussi avoir des styles correctement définis.



**Bonne structure et styles
correctement définis**

Analysez vos besoins

Quand vous étudiez votre structure, c'est toujours une bonne idée de repenser à ce que vous tentez de créer. Les commerciaux veulent qu'une image apparaisse dans un fondu quand les visiteurs cliquent sur la section « Montrez-moi l'ami à poils du jour » de la page. Quelles sont les modifications nécessaires dans le code HTML et CSS ?



← Définissons cette division comme cliquable.

← Et définissons cette division pour qu'elle soit masquée au départ. On lui attribue l'ID picframe.

MAINTENANT

faisons un peu de rangement

Utilisez le projet du
répertoire **CHAPITRE 1**

Définissons le code HTML et CSS

Commencez par réfléchir à ce que vous devez modifier dans le code HTML et CSS avant d'écrire une instruction jQuery. Ouvrez les fichiers jQuery du chapitre 1

Trouvez le dossier *Begin* du chapitre 1, puis ajoutez le code ci-dessous en gras aux fichiers, comme cela est indiqué.

À faire !

```
<!DOCTYPE html>
<html><head>
  <title>Furry Friends Campaign</title>
  <link rel="stylesheet" type="text/css" href="styles/my_style.css">
</head>
<body>
  <div id="clickMe">Show Me the Furry Friend of the Day</div>
  <div id="picframe">
    
  </div>
  <script src="scripts/jquery-X.X.X.min.js"></script>
  <script>
    $(document).ready(function() {
      $("#clickMe").click(function() {
        // Imbriquez l'image
        // furry_friend.jpg dans la
        // div picframe.
      });
    });
  </script>
</body>
</html>
```

Cela crée une div cliquable et nous allons la styler dans le fichier CSS ci-dessous de telle sorte qu'elle ait le même aspect que la div picframe.

Voici la div picframe qui va s'ouvrir en glissant pour afficher la photo de l'animal.

Imbriquez l'image furry_friend.jpg dans la div picframe.



index.html

```
#clickMe {
  background: #D8B36E;
  padding: 20px;
  text-align: center;
  width: 205px;
  display: block;
  border: 2px solid #000;
}

#picframe {
  background: #D8B36E;
  padding: 20px;
  width: 205px;
  display: none;
  border: 2px solid #000;
}
```

Cela style la div clickMe pour qu'elle ait le même aspect que la div picframe.

Définissez le sélecteur picframe à "display: none" pour qu'il ne s'affiche pas quand la page se charge.



my_style.css

X.X.X version du fichier dans le répertoire scripts.

jQuery à la loupe



Maintenant que le code HTML et CSS est prêt, analysons le code entre les balises `<script>`.

Dès que possible, je commence à exécuter le code entre les crochets !

Hé, DOM...

...quand tu seras prêt et chargé...

...je veux que tu fasses quelque chose pour moi.



DOM

```
$(document).ready(function() {
```

```
    $("#clickMe").click(function()
```

```
    {
```

Le point sépare le sélecteur de la méthode.

En connectant le bouton doté d'un ID clickMe à l'événement click, ce code rend le bouton cliquable.

Le code de l'action prévue quand on clique sur le bouton se place entre ces crochets (qu'on appelle aussi « bloc de code »).

```
    });
```

Le point-virgule est un terminateur. Il met fin à l'instruction jQuery click.

```
});
```

Ce point-virgule termine notre fonction jQuery ready.

glissons sur ces choses-là



Mais notre page ne fait
toujours rien de nouveau
pour l'instant !

**Vous avez raison. Notre code HTML et CSS
est prêt et il nous faut, maintenant passer à
jQuery.**

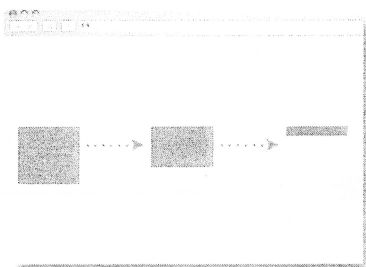
On veut que la div `picframe` s'affiche en glissant dans un
fendu. Heureusement, les développeurs de jQuery ont créé
des *effets* qui autorisent cela.

`.slideUp()`
`.slideDown()`
`.slideToggle()`

Glissons...

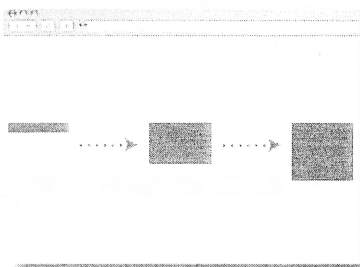
Le premier effet à implémenter est que l'image glisse quand elle
apparaît puisque c'est le souhait de l'équipe commerciale. Il y a
trois façons de gérer un glissement :

```
$("div").slideUp();
```



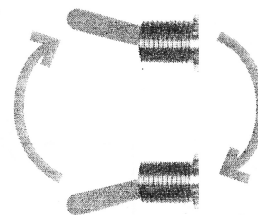
La méthode `slideUp` modifie la
propriété `height` de l'élément
jusqu'à ce qu'il soit égal à 0, puis
elle cache l'élément.

```
$("div").slideDown();
```



La méthode `slideDown`
modifie la propriété `height`
de l'élément en partant de 0
jusqu'à la valeur définie dans le
style CSS.

```
$("div").slideToggle();
```



L'action `slideToggle` dit « S'il est en
haut, je le fais glisser vers le bas ; s'il
est en bas, je le fais glisser vers le
haut ».

Que le fondu soit en toi

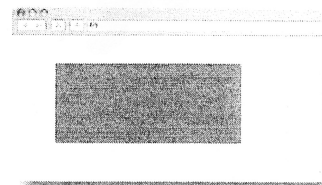
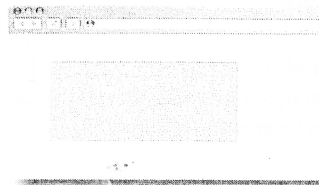
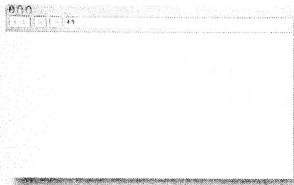
On veut aussi que l'image apparaisse graduellement, en passant de l'invisible au visible. Encore une fois, jQuery a tout prévu et possède une méthode qui s'appelle *fade*. Cette méthode est assez similaire à ce que nous venons de voir pour le glissement : on a `FadeIn`, `FadeOut`, `FadeTo` et `FadeToggle`. Pour l'instant, nous allons utiliser `FadeIn`, qui permet de contrôler les propriétés `opacity` et `transparency` des éléments HTML.

Voici ce que l'on veut voir apparaître en fondu ; dans ce cas, il s'agit d'une image.

On peut spécifier la rapidité du fondu en indiquant une valeur entre parenthèses, généralement exprimée en millisecondes (ms).

`.fadeIn()`

```
$("img").fadeIn();
```



Quand un élément apparaît en fondu, il passe de l'invisible (transparent) au visible (opaque).



COLLE

Combien d'instructions jQuery pensez-vous qu'il faut pour réaliser l'effet désiré ?

Essayez d'écrire ces instructions sur un bout de papier. Si vous n'êtes pas sûr de vous, écrivez en langage naturel ; puis commencez à habituer votre cerveau à penser en jQuery.

c'était facile, hein ?

Ça y est ?

C'est à peine croyable, mais il faut seulement **deux lignes** de code jQuery pour faire fonctionner ces effets. Vous commencez maintenant à comprendre pourquoi tant de gens apprécient jQuery. Ajoutez les lignes en gras ci-dessous à votre fichier *index.html*, et vous pouvez foncer.



```
<!DOCTYPE html>
<html>
  <head>
    <title>Furry Friends Campaign</title>
    <link rel="stylesheet" type="text/css" href="styles/my_style.
css">
  </head>
  <body>
    <div id="clickMe">Show me the Furry Friend of the Day</div>
    <div id="picframe">
      
    </div>
    <script src="scripts/jquery-X.X.X.min.js"></script>
    <script>
      $(document).ready(function()
        $("#clickMe").click(function() {
          $("img").fadeIn(1000);
          $("#picframe").slideToggle("slow");
        });
      });
    </script>
  </body>
</html>
```

En jQuery, il est important de séquencer nos effets de telle sorte qu'ils ne se chevauchent pas.

On exécute d'abord l'effet de fondu sur notre image.

On a ajouté un bonus entre parenthèses pour améliorer les effets.



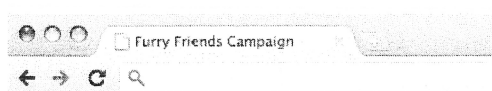
index.html

X.X.X version du fichier dans le répertoire scripts.

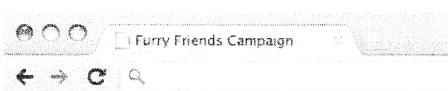


BANC D'ESSAI

Ouvrez la page dans votre navigateur favori pour vous assurer que tout fonctionne correctement.



Cliquez ici.



Votre image doit apparaître en glissant dans un fondu.



Attention !

Vérifiez dans plusieurs navigateurs.

Ce n'est pas parce que jQuery fonctionne de la même manière sur tous les navigateurs que les styles que vous avez définis dans votre fichier CSS, ou les styles dynamiques que vous appliquez aux éléments de votre page, réagiront de manière identique dans tous les navigateurs !