TP 4 : TRAVERSER ET MANIPULER LE DOM

1. TRAVERSER LE DOM

Pour ce TD, nous utiliserons une page type.

Celleci comporte une liste non ordonnée avec 5 items et un tableau d'une ligne et 5 colonnes.

Il faut remarquer aussi les divisions `exemple` et `contenu` car elles interviennent plus loin dans notre étude.

1.1. TROUVER LES ENFANTS

Récupère un groupe d'éléments contenant les enfants immédiats de chacun des éléments concernés par la sélection.

```
$("div").children()
```

Cette méthode renvoie un objet jQuery.

Exercice : *Ajouter une bordure aux enfants de la division* `<div class="contenu">` *et un arrièreplan aux enfants de la cellule 3 du tableau.*

1.2. TROUVER LES PARENTS DIRECTS

parent()

Récupère un groupe d'éléments contenant les parents immédiats de chacun des éléments concernés par la sélection.

```
$("span").parent()
```

Cette méthode renvoie un objet jQuery.

Exercice : *Entourer d'une bordure les parents immédiats du troisième élément de la liste et ajoutons un arrièreplan de couleur aux parents immédiats de la troisième cellule du tableau.*

1.3. TROUVER TOUS LES PARENTS

parents()

Récupère un groupe d'éléments contenant tous les parents de chacun des éléments concernés par la sélection.

```
$("li").parents()
```

Cette méthode renvoie un objet jQuery.

Exercice *:Entourer d'une bordure les parents du troisième élément de la liste.*

1.4. TROUVER TOUS LES PARENTS JUSQU'À

parentsUntil(sélecteur ou élément)

Retourne les parents de chaque élément jusqu'à (non compris) l'élément désigné par le sélecteur.

```
$("div.box").parentsUntil("body") ;
```

Cette méthode renvoie un objet jQuery.

Exercice : Soit une division qui compote une liste de 4 items (Division 2). Cette division est elle-même comprise dans une division (Division 1). Ajouter in arrière-plan de couleur aux parents de l'item 3 de la lste jusqu'à la balise <body>.

1.5. TROUVER LES FRÈRES

siblings()

Retourne la liste des frères immédiats de chaque élément de la sélection.

```
$("div").siblings()
```

Cette méthode renvoie un objet jQuery.

Exercice : Entourer d'une bordure les frères du troisième élément de la liste et ajoutons un arrièreplan de couleur aux frères de la troisième cellule du tableau.

1.6. TROUVER LE FRÈRE PRÉCÉDENT

prev()

Retourne le frère immédiat précédent de chaque élément de la sélection.

```
$("td").prev()
```

Cette méthode renvoie un objet jQuery.

Exercice :Entourer d'une bordure le frère précédent du troisième élément de la liste et ajoutons un arrièreplan de couleur au frère précédent de la troisième cellule du tableau.

1.7. TROUVER LES FRÈRES PRÉCÉDENTS

prevAll()

Retourne les frères immédiats précédents de chaque élément de la sélection.

```
$("td").prevAll()
```

Cette méthode renvoie un objet jQuery.

Exercice : Entourer d'une bordure le frère précédent du troisième élément de la liste et ajoutons un arrièreplan de couleur au frère précédent de la troisième cellule du tableau.

1.8. TROUVER LES FRÈRES PRÉCÉDENTS JUSQU'À

prevUntil(sélecteur ou élément)

Retourne les frères précédents de chaque élément jusqu'à (non compris)l'élément désigné par le sélecteur.

```
$("p").prevUntil("h1") ;
```

Renvoie les paragraphes précédents jusqu'à la balise <h1>.

Cette méthode renvoie un objet jQuery.

Exercice : *soit une liste de définition (balise <dl>) qui comporte deux défintions (balise <dt>). Ajouter un arrière-plan de couleur aux éléments précédents de la seconde définition jusqu'à non compris la première définition.*

1.9. TROUVER LE FRÈRE SUIVANT

next()

Retourne le frère immédiat suivant de chaque élément de la sélection.

```
$("td").prev()
```

Cette méthode renvoie un objet jQuery.

Exercice : *Entourer d'une bordure le frère suivant du troisième élément de la liste et ajouter un arrièreplan de couleur au frère suivant de la troisième cellule du tableau.*

1.10. TROUVER LES FRÈRES SUIVANTS

prevUntil(sélecteur ou élément)

Retourne les frères immédiats suivants de chaque élément de la sélection.

```
$("td").nextAll()
```

Cette méthode renvoie un objet jQuery.

Exercice : Entourer d'une bordure les frères immédiatement suivants du troisième élément de la liste et ajouter un arrièreplan de couleur aux frères immédiatement suivants de la troisième cellule du tableau.

1.11. TROUVER LES FRÈRES SUIVANTS JUSQU'À

nextUntil(sélecteur ou élément)

Retourne les frères suivants de chaque élément jusqu'à (non compris) l'élément désigné par le sélecteur.

```
$("ul > :first").nextUntil(" :last") ;
```

Renvoie tous les items de la liste compris entre le premier et le dernier.

Exercice : *faire disparaître et apparaître des lignes de tableau selon un certain critère de sélection.*

1.12. TROUVER LE CONTENU

contents()

Trouve tous les nœuds enfants situés dans les éléments de la sélection (incluant les nœuds texte). Si l'élément spécifié est une balise `<iframe>`, `contents()` trouve le contenu du document.

```
$("p").contents()
```

Cette méthode renvoie un objet jQuery.

Exercice :Entourer d'une bordure, le contenu du troisième élément de la liste et ajouter un arrièreplan de couleur au contenu de la troisième cellule du tableau.

1.13. TROUVER CERTAINS PARENTS

closest(sélecteur)

Retourne l'ensemble d'éléments contenant le parent le plus proche de l'élément sélectionné répondant au sélecteur, l'élément de départ inclus.

La méthode `closest()` vérifie d'abord si l'élément courant répond à l'expression spécifiée. Dans l'affirmative, il retourne simplement l'élément spécifié. Sinon, il continue alors de traverser le document vers le haut, parent par parent, jusqu'à ce qu'il trouve un élément répondant à la condition de l'expression. Si aucun élément n'est trouvé, la méthode ne retourne rien.

```
$("div").closest("p")
```

Cette méthode renvoie un objet jQuery.

Exercice : Retrouver les parents du troisième élément de la liste jusqu'à la division dont la classe est `contenu` et ajouter un arrière plan de couleur.

1.14. TROUVER CERTAINS DESCENDANTS

find(expression ou sélecteur)

Recherche les éléments descendants répondant aux conditions du sélecteur spécifié.

```
$("div").find("p")
```

Cette méthode renvoie un objet jQuery.

Exercice :Trouver la balise non ordonnée `<ul>` qui est un descendant de la division identifiée par `exemple`.

1.15. AJOUTER DES ÉLÉMENTS À LA SÉLECTION

add(sélecteur ou élément(s) ou Html)

Ajoute des éléments, fournis en argument, à l'ensemble des éléments sélectionnés dans la recherche. Les paramètres peuvent être :

← • un sélecteur jQuery : `$("p").add("span") ;`

← • un ou des éléments : `$("p").add(document.getElementById("a")) ;`

← • du code Html : `$("p").add("<span>Again</span>").`

← Cette méthode renvoie un objet jQuery.

← *Exercice : Après avoir retenu les éléments de la liste, ajouter à la sélection, la cellule du tableau identifiée par l'identifiant `select_table`.*

1.16. UNE LOUPE POUR AGRANDIR LES VIGNETTES

Exercice : Faire apparaître au passage du curseur sur une vignette, une loupe pour suggérer au visiteur d'agrandir celleci.

1. MANIPULER LE DOM

L'intégration du DOM a profondément modifié l'écriture du JavaScript par son traçage des éléments. Mais sa véritable révolution est, sans conteste, la possibilité de modifier et d'ajouter à la volée des éléments dans la page Html.

1.17. MODIFIER LE CONTENU

text()

Récupère le contenu texte de l'élément concerné.

Cette méthode fonctionne pour les documents Html, Xhtml et XML.

`$("div").text()` : récupère au format texte le contenu de la balise `<div>`.

Cette méthode renvoie une chaîne de caractères (String).

text(valeur)

Assigne un nouveau contenu texte (voir l'argument valeur) aux éléments concernés.

`$("div").text("les nouveaux éléments de texte")` : insère au format texte, le contenu `"les nouveaux éléments de texte"` dans la balise `<div>`.

Cette méthode renvoie un objet jQuery.

html()

Récupère au format Html, le contenu de l'élément concerné.Cette méthode ne fonctionne pas pour les documents XML (à l'exception des documents Xhtml). `$("div").html()` : récupère au format Html, le contenu de la balise `<div>`.Cette méthode renvoie une chaîne de caractères (String).

html(valeur)

Assigne un nouveau contenu Html (voir l'argument valeur) aux éléments concernés.Cette propriété n'est pas disponible pour les documents XML mais elle l'est bien pour les documents Xhtml.

`$("div").html("<b>nouveau contenu</b>")` : insère comme du Html, les éléments fournis en argument dans la balise `<div>`.

Cette méthode renvoie un objet jQuery.

Exercice : Soit un élément boîte. Au clic sur le bouton, un nouveau contenu est injecté dans celuici. Cette opération réalisée, le nouveau contenu est affiché par la méthode `text()`.

1.18. INSÉRER À L'INTÉRIEUR

1.18.1. Première méthode

append(contenu)

Ajoute du contenu à la fin mais à l'intérieur de l'élément spécifié.Le contenu peut être une chaîne de caractères, du Html ou un objet jQuery.

`$("p").append("<b>Hello</b>")` : insère à la fin du paragraphe, les éléments fournis en arguments.

Cette méthode renvoie un objet jQuery.

prepend(contenu)

Ajoute du contenu au début mais à l'intérieur de l'élément spécifié.Le contenu peut être une chaîne de caractères, du Html ou un objet jQuery.

`$("p").prepend("<b>Hello</b>")` : insère au début du paragraphe, les éléments fournis en arguments.

Cette méthode renvoie un objet jQuery.

Exercice :Ajouter du contenu au début et à la fin dans l'élément boîte du point précédent.

1.18.2. Seconde méthode

Les méthodes `appendTo()` et `prependTo()` effectuent les mêmes tâches que `append()` et `prepend()`.

La seule différence provient du placement dans le code, du contenu et de la cible. Avec `append()` ou `prepend()`, le sélecteur précédant la méthode est le conteneur dans lequel est inséré le contenu. Avec `appendTo()` ou `prependTo()`, le contenu précède la méthode et est alors inséré dans la conteneur cible.

appendTo()

Ajoute des éléments spécifiés par le sélecteur `A` à la fin d'autres spécifiés par `B` selon l'expression `$(A).appendTo(B)`.

`$("p").appendTo("#box")` :

ajoute le contenu des éléments `<p>`, à la division portant l'identifiant `"box"` et à la fin de celleci.

Cette méthode renvoie un objet jQuery.

prependTo()

Ajoute des éléments spécifiés par le sélecteur `A` au début d'autres spécifiés par `B` selon l'expression `$(A).appendTo (B)`.

`$("p").prependTo("#box")` :

ajoute le contenu des éléments `<p>`, à la division portant l'identifiant `"box"` et au début de celleci.

Cette méthode renvoie un objet jQuery.

Exercice : Reprendre les mêmes opérations de l'exercice précèdent.

1.19. INSÉRER À L'EXTÉRIEUR

after(contenu)

Ajoute du contenu spécifié en argument après l'élément de la sélection.Le contenu peut être une chaîne de caractères, du Html ou un objet jQuery.

`$("p").after("<b>Hello</b>")` :

ajoute après la balise de paragraphe, le contenu fourni en argument.

Cette méthode renvoie un objet jQuery.

before(contenu)

Ajoute du contenu spécifié en argument avant chaque élément de la sélection.Le contenu peut être une chaîne de caractères, du Html ou un objet jQuery.

`$("p").before("<b>Hello</b>")` :

ajoute avant la balise de paragraphe, le contenu fourni en argument.

Cette méthode renvoie un objet jQuery.

Exercice: Ajouter du contenu avant et après l'élément boîte du point précédent.

1.20. ENTOURER UN ÉLÉMENT

wrap(html ou élément)

Entoure chaque élément sélectionné avec l'élément fourni en argument. Cette

procédure est très utile pour injecter une structure de code additionnelle dans un document sans modifier la sémantique originelle du document.

```
$("p").wrap("<div class='wrap'></div>");
```

Cette méthode renvoie un objet jQuery.

wrapAll(html ou élément)

Entoure tous les éléments de la sélection par un seul élément. Différent de la fonction `wrap()` qui entoure chaque élément de la sélection par un nouvel élément (voir exemples ciaprès).

```
$("p").wrapAll("<div></div>");
```

Cette méthode renvoie un objet jQuery.

wrapInner(html ou élément)

Entoure les enfants d'un élément (les nœuds textes inclus) par un autre élément.

```
$("p").wrapInner("<b></b>");
```

Cette méthode renvoie un objet jQuery.

Exercice :Appliquer ces méthodes à un exemple similaire aux précédents.

1.21. REMPLACER UN ÉLÉMENT

replaceWith()

Remplace l'élément courant par le contenu spécifié, sous forme de code Html ou d'objet DOM. La fonction retourne l'élément JQuery qui vient d'être remplacé et supprimé du DOM.

```
$("#div1").replaceWith("<div>Nouvelle division</div>");
```

Cette méthode renvoie un objet jQuery.

Commentaires : La méthode `html()` remplace le contenu de l'élément tandis que `replaceWith()` remplace l'élément luimême.

Exercice : Modifier un bouton de soumission de formulaire après un clic sur celui-ci.

1.22. ENLEVER UN ÉLÉMENT

1.22.1. Supprimer un élément

remove()

Supprime de l'arbre du DOM, tous les éléments répondant aux critères de sélection. Cependant cette méthode ne supprime pas les éléments de l'objet jQuery, ce qui permet une utilisation ultérieure de ces éléments même si ceuxci ne figurent plus dans le document.

`$("p").remove()` : supprime le paragraphe.

Cette méthode renvoie un objet jQuery.

Exercice : *Prendre un bouton de type case à cocher (checkbox). Lorsque celuici est sélectionné, un formulaire de ligne de texte est affiché. Lorsque celuici est désélectionné, la ligne de texte disparaît.*

1.22.2. Vider un élément

empty()

Supprime tous les nœuds enfants de l'élément sélectionné.

`$("p").empty()` : supprime le contenu du paragraphe.

Cette méthode renvoie un objet jQuery.

Exercice : Illustrer par un exemple, la différence entre les méthodes `empty()` et `remove()`.

1.23. COPIER UN ÉLÉMENT

clone()

Copie (clone) les éléments DOM trouvés et les sélectionne. Cette fonction est utile pour créer des copies d'éléments et les déplacer vers un autre endroit spécifié du DOM.

`$("p").clone()` : copie le paragraphe.Paramètres (optionnel) : indiquez `true` si vous souhaitez cloner les gestionnaires d'événements associés à la sélection.

Cette méthode renvoie un objet jQuery.

Exercice : Cloner une division et son contenu (`<div id="div1">` pour les déplacer à un autre endroit de la page (`<div id="div2">`).

1.24. QUELQUES APPLICATIONS

1.24.1. Ajouter un pied de page et des liens de retour

Nous allons ajouter avec quelques lignes de jQuery, un pied de page et des liens de retour vers le haut de la page.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html
xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">

<head><meta http-equiv="Content-Type" content="text/html;
charset=iso- 8859-1" /><title>jQuery</title><style type="text/css">a
{ color: black;}#foot { border-top: 1px solid black; margin-top:
15px;} </style>

</head>
```

```
<body>

<h2>Titre de la page</h2>

<div id="contenu">

<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent sodales, nisi eget aliquet interdum, urna risus sodales
urna, at auctor nisl tellus rutrum nibh. Pellentesque lorem diam,
tincidunt eget tincidunt non, rutrum non felis. In hac habitasse
platea dictumst. Ut nibh leo, placerat a tristique consequat, vesti-
bulum at tortor. In nec tristique turpis.</p>

<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent sodales, nisi eget aliquet interdum, urna risus sodales
urna, at auctor nisl tellus rutrum nibh. Pellentesque lorem diam,
tincidunt eget tincidunt non, rutrum non felis. In hac habitasse
platea dictumst. Ut nibh leo, placerat a tristique consequat, vesti-
bulum at tortor. In nec tristique turpis.</p>

<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent sodales, nisi eget aliquet interdum, urna risus sodales
urna, at auctor nisl tellus rutrum nibh. Pellentesque lorem diam,
tincidunt eget tincidunt non, rutrum non felis. In hac habitasse
platea dictumst. Ut nibh leo, placerat a tristique consequat, vesti-
bulum at tortor. In nec tristique turpis.</p>

<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent sodales, nisi eget aliquet interdum, urna risus sodales
urna, at auctor nisl tellus rutrum nibh. Pellentesque lorem
diam,tincidunt eget tincidunt non, rutrum non felis. In platea dic-
tumst. Ut nibh leo, placerat a tristique vestibulum at tortor. In
nec tristique turpis.</p>

</div>

</body> </html>
```

À ce stade, les pieds de pages et les liens de retour n'apparaissent pas dans le fichier Xhtml.

Ajoutons le code jQuery.

```
<script type="text/javascript">//<![CDATA[$(document).ready(func-
tion(){$('<a id="top" name="top"></a>').prependTo('body'); $
("p").after("<a href='#top'>Haut de page</a>"); $
("#contenu").after("<div id='foot'><i>Notes de pied de
page</i></div>");

}); //]]> </script>
```

Expliquons ce script.

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

Initialisation du DOM dans jQuery.

```
$('<a id="top" name="top"></a>').prependTo('body');
```

Plaçons d'abord l'ancre (`<a id="top" name="top"></a>`) au début de la page. La méthode `prepend()` appliquée à la

balise `<body>` permet de placer l'ancre juste après celleci. `$("p").after("<a href='#top'>Haut de page</a>");`

Cette ligne de code placera le lien de retour vers le haut de la page (`<a href='#top'>Haut de page</a>`) après chaque occurrence de la balise `<p> ... </p>`.

```
$("#contenu").after("<div id='foot'><i>Notes de pied de
page</i></div>");
```

Pour le pied de page (`<div id='foot'><i>Notes de pied de page</i></div>`), il sera inséré après la division contenu qui contient les différents paragraphes.

```
});
```

Fin du script.Cet exemple illustre bien, la concision du code jQuery même pour réaliser des opérations complexes.

1.24.2. Ajouter et enlever des éléments d'une liste

Soit une liste non ordonnée. Au clic sur un lien, ajoutons un élément de liste. Au clic sur un autre lien, supprimons un élément de liste.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html
xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">

<head><meta http-equiv="Content-Type" content="text/html;
charset=iso- 8859-1" /><title>jQuery</title><style type="text/css">
ul { list-style-type: square;}

a { color: black;} </style></head><body>

<ul><li>Item<li>Item<li>Item<li>Item<li>Item</ul><a href="#"
id="add">Ajouter un item</a><br /> <a href="#" id="remove">Enlever
un item</a> </body>

</html>
```

Le code jQuery :

```
<script type="text/javascript">//<![CDATA[$(document).ready(func-
tion(){var i = $('li').size() + 1; $('a#add').click(function() {$
('<li>Item ' + i + '</li>').appendTo('ul'); i++;

}); $('a#remove').click(function() { $('li:last').remove();i--;});
});//]]></script>
```

Explications :Le script s'articule autour des deux opérations suivantes.

```
$('a#add').click(function()     {$('<li>Item     '     +     i     +
'</li>').appendTo('ul');
```

Au clic sur le lien **Ajouter un item,** on ajoute comme dernier élément, un élément `<li> ... </li>` à la liste non ordonnée.

```
$('a#remove').click(function() { $('li:last').remove();
```

Au clic sur le lien **Ajouter,** on supprime le dernier élément `<li> ... </li>` (`li:last`).Il faut cependant ajouter un compteur pour incrémenter ou décrémenter le numéro associé à l'item.

```
var i = $('li').size() + 1; $('a#add').click(function() {$('<li>Item
' + i + '</li>').appendTo('ul'); i++;});
```

On ajoute ainsi un compteur (var `i`). Celuici démarre au nombre d'éléments de liste (size()). On veillera à ajouter une unité car, comme souvent, JavaScript démarre ses comptages à 0. Une fois l'élément de liste ajouté, la variable `i` est incrémentée d'une unité (i++).

```
$('a#remove').click(function() { $('li:last').remove();i--;});
```

De façon similaire, à chaque fois qu'un élément de liste est supprimé, le compteur `i` est décrémenté d'une unité.

```
});
```

Fin de script.

1.24.3. Ajouter une icône aux liens externes

La petite icône qui suit les liens externes est très à la mode sur les sites Web 2.0. En ce qui nous concerne, jQuery permet d'ajouter facilement ces petites images.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html
xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">

<head><meta http-equiv="Content-Type" content="text/html;
charset=iso- 8859-1" /><title>jQuery</title><script type="text/ja-
vascript" src="jquery.js"></script> <script type="text/javascript">
//<![CDATA[$(document).ready(function() {$("a").filter(function() {
return this.hostname !== location.hostname;}).after('<img src="ex-
ternal.png" class="external">');});//]]></script><style
type="text/css">a { color: black;

text-decoration: none;} .external { border: none;

margin-left: 10px;}

</style></head><body><a href="http://www.editions-eni.fr/"
```

```
title="Lien externe" >Editions Eni</a><br />

<a href="#">Lien interne</a><br /> <a href="#">Lien
interne</a><br /> </body></html>
```

Pour identifier les liens externes, nous testons le nom de domaine de la page (location.hostname) par rapport au nom de domaine (this.hostname) des liens. Nous nous assurons que les deux ne correspondent pas (this.hostname `&& location.hostname !== this.hostname`).

```
.after('<img src="external.png" class="external">');
```

Ce filtre appliqué, le script insère après le lien (after) l'icône de lien externe (<img `src="external.png" class="external">`).

1.24.4. Ajouter des bords arrondis

Mis à part les éléments ajoutés par les images, le design des pages Html est essentiellement linéaire avec des lignes verticales et horizontales. C'est le cas, par exemple, des tableaux, des divisions et jadis des cadres. La présence de lignes courbes ou de coins arrondis est une tendance qui apporte non seulement une certaine originalité mais aussi une touche de "douceur".

Avant de se pencher sur le code, nous avons besoin des images pour l'arrondi des bords. À cet effet, nous utilisons un générateur de bords arrondi soit par exemple, celui proposé par le site www.roundedcorner.com.

Nous obtenons alors les images rounded_tl.png (tl pour top left), rounded_tr.png (tr pour top right), rounded_bl.png (bl pour bottom left) et rounded_br (br pour bottom right).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><html
xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="fr-FR"> <head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" /><title>jQuery</title><script type="text/javascript"
src="jquery.js"></script><script type="text/javascript">//<![CDATA[$
(document).ready(function(){$('div.rounded').wrap('<div
class="cadre"></div>'); $('div.cadre').prepend('<div
class="cadre_tl"></div>'); $('div.cadre').prepend('<div
class="cadre_tr"></div>'); $('div.cadre').append('<div
class="cadre_bl"></div>'); $('div.cadre').append('<div
class="cadre_br"></div>');});//]]></script><style type="text/css">
#contenu { background-color: #9cf;

padding: 0 8px;} p { margin: 0; }

.cadre { background-color: #9cf; width: 275px;}

.cadre_tl, .cadre_tr, .cadre_bl, .cadre_br{ width: 100%;height:
11px; font-size: 1px;

}
```

```
.cadre_tl{background: url('rounded_tl.png') no-repeat top left;

.cadre_tr{background: url('rounded_tr.png') no-repeat top right;
float: right;}.cadre_bl{background: url('rounded_bl.png') no-repeat
bottom left;float: right;}.cadre_br{background:
url('rounded_br.png') no-repeat bottom right;}</style></head><body>
<br /><div id="contenu" class="rounded"><p>Lorem ipsum dolor sit
amet, consectetuer adipiscing elit. Sed vitae diam egestas felis
pellentesque bibendum. Proin posuere felis tellus. Phasellus eu fe-
lis. Nullam lacus ante.</p></div></body>

</html>
```

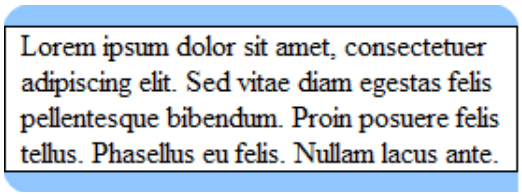
Explications :

`$("div.contenu").wrap("<div class="cadre"></div>");`Commençons par ajouter audessus et endessous (méthode `wrap()`) de la division dont la classe est `contenu`, la division dont la classe est `cadre`.

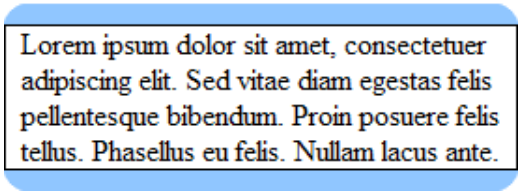
`$("div.cadre").prepend("<div class='cadre_tl'></div>");`À l'intérieur de la division `cadre`, ajoutons en premier élément (méthode `prepend()`), l'image rounded_tl.png contenue en arrièreplan dans la division `cadre_tl`.

`$("div.cadre").prepend("<div class='cadre_tr'></div>");`Procédons de même (méthode `prepend()`) pour l'image rounded_tr.png contenue en arrière-plan dans la division `cadre_tr`.

`$("div.cadre").append("<div class='cadre_bl'></div>");`Ajoutons maintenant, toujours à l'intérieur de la division cadre mais en dernière position (méthode `append()`), l'image rounded_bl.png contenue en arrièreplan dans la division `cadre_bl`.



`$("div.cadre").append("<div class='cadre_br'></div>");`Enfin, procédons de même (méthode `append()`), pour l'image rounded_br.png contenue en arrièreplan dans la division `cadre_br`.



Il est possible d'écrire le code de façon plus concise. Ainsi, les lignes avec `pre-`

pend() et append() deviendraient :

```
$("div.cadre").prepend("<div class='cadre_hd'></div><div
class='cadre_hg'></div>");$("div.cadre").append("<div
class='cadre_bd'></div><div class='cadre_bg'></div>");
```