>
1. ...
5. ...

Télécharger

Utiliser des URLs en chemins relatif

Pas besoin de spécifier le protocole http ou https. Le navigateur optera de lui-même pour le protocole https s'il se trouve déjà dans un environemment sécurisé.

Télécharger

Raccourci pour l'événement ready

```
// Appel classique
$(document).ready(function() {
...
});
// Version rapide
$(function() {
});
```

Télécharger

Garder le $ lors de la déclaration d'une variable de type jQuery.

Grâce à cette convention de nommage, on distingue facilement la nature de l'objet JQuery.

```
// :(
var form = $('#contactForm');
// :)
var $form = $('#contactForm');
```

Télécharger

Du bon usage de $this

La variable $this s'utilise avantageusement au début des fonctions non déclarées, par exemple dans une boucle de type each.

```
// :(
$('li').each(function() {
$(this).on('click', function() {
$(this).addClass('active');
});
});
// :)
$('li').each(function() {
var $this = $(this);
$this.on('click', function() {
$this.addClass('active');
});
});
```

Télécharger

Certains préfèrent utiliser *that* ou *self*. Attention de ne pas oublier qu'il s'agit d'un objet jQuery.

Mettre en cache les objets jQuery

Si un objet jQuery est utilisé plusieurs fois, le mettre en cache permet d'optimiser les performances.

```
// :(
$('.menu li').each(function() { ... });
$('.menu li').each(function() { ... });
// :)
var $items = $('.menu li');
$items.each(function() { ... });
// on recycle :)
$items.each(function() { ... });
```

Télécharger

Enchaînement des fonctions

C'est asssurément l'une des fonctionalités les plus appréciée de jQuery. On peut ainsi appeler une série de méthodes dans la même foulée.

> **"Write less, do more", garder en tête le slogan de jQuery**

```
// :(
var $a = $('#about');
$a.hide();
$a.addClass();
$a.fadeIn();
```

```
$a.hide();
// :)
$('#about').hide().addClass().fadeIn().hide();
// c'est mieux
// Retour à la ligne et identation améliorent la visibilité
$('#about')
.hide()
.addClass()
.fadeIn()
.hide();
```

Télécharger

Déclarer un nouvel élément

Lors de la création d'un élément, faîtes en sorte de manipuler les éléments via les méthodes jQuery plutôt que d'insérer du code HTML brut.

```
// Don't
var $hidden = $('<input type="hidden" name="foo" value="bar" />').appendTo('#form');
// :)
var $hidden = $('<input />')
.addClass('form-control')
.attr('type', 'hidden')
.attr('name', 'foo')
.val('bar')
.appendTo('#form');
// ou bien
var $hidden = $('<input />', {
class: 'form-control',
type: 'hidden',
name: 'foo',
value: 'bar'
}).appendTo('#form');
```

Télécharger

Garder le CSS loin des manipulations de jQuery

Pas la peine de déclarer le style CSS directement à un élément. Le recours aux classes est fait pour ça.

```
// :(
$('#button').css({
'background-color': '#5cb85c',
'border-color': '#4cae4c'
```

```
});
// :)
.success {
background-color: #5cb85c;
border-color: #4cae4c;
}
$('#button').addClass('success');
```

Télécharger

Choisir le bon sélecteur

Le sélecteur désignant l'*id* est le plus rapide

Pour retrouver un élément du DOM en fonction de son *id*, jQuery utilise la méthode native *document.getElementById()* qui s'avère bien plus efficace que Sizzle.

> **Sizzle is a pure-JavaScript CSS selector engine used by jQuery**

```
// :(
$('#wrapper #inner');
$('div#inner');
$('.wrapper #inner');
// :)
$('#inner');
```

Télécharger

Du coup, mieux vaut introduire une recherche sur un *id*, quitte à enchaîner les recherches.

```
// :(
$('#container .row');
// + rapide
$('#container').find('.row');
```

Télécharger

Sélecteurs restrictifs

Il faut être spécifique sur le partie gauche de votre sélecteur, et moins en début.

```
// pas glup
$('div.data .gonzalez');
// glup glup
$('.data td.gonzalez');
```

Télécharger

Éviter le recours aux sélecteurs universels

```
// moins rapide
$('div.container > *');
// Plus rapide
$('.container').children();
```

Télécharger

Mieux vaut faire précéder les sélecteurs pseudo-class (ex *:before*) avec un tag ou un autre sélecteur. Car, si ce n'est pas le cas, le sélecteur universel * est implicitement employé.

```
// :(
$('.category :radio');
// :)
$('.category input:radio');
```

Télécharger

Privilégiez les méthodes de tri plutôt que les pseudos-sélecteurs.

Lorsque cele est possible, utiliser la méthode de tri jQuery plutôt que des pseudos-sélecteurs. La méthode querySelectorAll s'avère là encore plus rapide que la méthode Sizzle.

```
// :(
$('.item:first')
// :)
$('.item').eq(0)
```

Télécharger

Pas de javascript *inline* sur les élements HTML

Mieux vaut attacher un écouteur d'événement à l'objet.

```
<button id="saveButton" onclick="javascript: save();">Save
// :)
$('#saveButton').on('click', function() {
...
});
```

Télécharger

Choisir un namespace personnalisé pour les événements

Ainsi il est plus facile de désactiver un événement sans affecter les autres écouteurs d'événements assignés à l'élément.

```
1. $('#saveButton').on('click.bv', function() { ... });
2. //Plus tard, on peut retirer sans crainte l'écouteur l'événement
3. $('#saveButton').off('click.bv');
```

Télécharger

Ne pas passer les paramètres Ajax "en dur"

Lorsque d'un requête de type xmlHttpRequest, il faut utiliser le paramètre *data*, et non concaténer l'information au sein de l'URL.

```
1. // :(
2. $.ajax({
3. url: '/remote/url?param1=value1&param2=value2...'
4. }});
5. // :)
6. $.ajax({
7. url: '/remote/url',
8. data: {
9. param1: 'value1',
10. param2: 'value2'
11. ...
12. }
13. });
```

Télécharger

Dans le cas où les paramètres à transmettre sont très long (ex : le contenu intégral d'un article), on privilégiera la méthode POST, à la fois pour Ajax et le traitement côté serveur.

> **Internet Explorer 8 (and earlier) limits 2083 characters in URL**

Voir en ligne : Best jQuery practices · Programmer Tips

Notes

[1] Content Delivery Network : Un content delivery network (CDN) est constitué d'ordinateurs reliés en réseau à travers Internet et qui coopèrent afin de mettre à disposition du contenu ou des données (généralement du contenu multimédia

volumineux) à des utilisateurs.