

DÉVELOPPEMENT JAVASCRIPT

jour n°4

JS

JS

- 7. Ajax
- 8. jQuery

JS

1. Rappels
2. Premiers pas en JS
3. La POO en JS
4. Les modules
5. Le typage
6. API DOM

7. Ajax

8. jQuery
9. bonus track

AJAX

- Définition
- XMLHttpRequest
- fetch
- Sécurité

AJAX

- = Asynchronous JavaScript And XML
- Appel à un script distant depuis JavaScript sans rechargement de page
- pas limité à XML (plutôt JSON en général)

XMLHttpRequest : GET

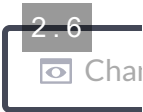
```
const request:XMLHttpRequest = new XMLHttpRequest();
request.open('GET', 'api/users');
request.onload = (event:ProgressEvent) => {
    console.log( request.responseText );
}
request.send();
```

XMLHttpRequest : POST

```
const request:XMLHttpRequest = new XMLHttpRequest();
request.open('POST', 'api/users');
request.setRequestHeader('Content-type',
                        'application/x-www-form-urlencoded');
request.onload = (event:ProgressEvent) => {
    console.log( request.responseText );
}
request.send('firstname=Jon&lastname=Targaryen');
```



A PEN BY Kumquats



2.6

Chai

```
1 <div
  id="containe
  </div>
```

CS

```
1 const request = new XMLHttpRequest();
2 request.open('GET', 'https://enable-cors.org/', true);
3 request.onload = function(){
4     document.querySelector('#container').innerHTML = request.responseText;
5 }
6 request.send();
```

960px

- [Home](#)
- [Server](#)
- [Client](#)
- [Resources](#)
- [Test](#)
-
- [Suggest!](#)
- [Help!](#)

Cross-Origin Resource Sharing (CORS) is a [specification](#) that enables truly open access across domain-boundaries. If you serve public content, please consider using CORS to open it up for universal JavaScript/browser access.

[\[more...\]](#)

Console

Assets

Comments

Shortcuts

Last saved 4 days ago

Delete

Collections

Embed

Export

Share

Notes :

<http://codepen.io/kumquats/pen/QdjYyw?editors=0010> (<http://codepen.io/kumquats/pen/QdjYyw?editors=0010>)

FETCH

- nouvelle syntaxe plus simple
- support des Promises

```
fetch('api/users')  
  .then( (response:Reponse) => response.text() )  
  .then( (reponseText:string) => console.log(responseText) )
```

FETCH : LES ERREURS

- `response.ok` : `true/false`
- `response.status` : code HTTP (200, 400, 404, 500, ...)
- `response.statusText` : 'OK', 'Bad Request', 'Not found', etc.

```
fetch('api/users')
  .then( (response:Response) => {
    if (! response.ok ) {
      throw new Error( `Erreur : ${response.statusText}` );
    }
    return response.text();
  })
  .then( (responseText:string) => console.log(responseText) )
  .catch( (error:Error) => alert( error.message ) )
```

x | Feature: Fetch

Fetch 📄 - LS

A modern replacement for XMLHttpRequest.

Usage	% of all users	
Global	87.67%	+ 0.05% = 87.72%
France	87.68%	+ 0.02% = 87.7%

Current aligned	Usage relative	Date relative	Apply filters		Show all					
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera
		2-33	4-39		10-26					
		^{1 4} 34-38	² 40		² 27					
	12-13	⁴ 39	^{2 3} 41	3.1-10	^{2 3} 28	3.2-10.2				
6-10	14-17	40-64	42-71	10.1-11.1	29-57	10.3-11.4		2.1-4.4.4	7	12
11	18	65	72	12	58	12.1	all	67	10	
		66-67	73-75	12.1-TP		12.2				

- Notes
- Sub-features (1)
- Known issues (0)
- Resources (7)
- Feedback

Notes :

<https://caniuse.com/#feat=fetch> (<https://caniuse.com/#feat=fetch>)

Notes :

Ce polyfill permet d'utiliser l'api fetch sur Internet Explorer :

<https://github.com/github/fetch> (<https://github.com/github/fetch>)

Code

Issues 8

Pull requests 3

Insights

A window.fetch JavaScript polyfill.

<http://github.github.io/fetch/>

polyfill

fetch

javascript

promise

587 commits

4 branches

28 releases

51 contributors

MIT

Branch: master

New pull request

Find file

Clone or download

mislav

Merge pull request #684 from justjavac/master

Latest commit 3674c98 on 2 Jan

test	field name can not by empty	2 months ago
.eslintrc.json	Compatibility with newer eslint-plugin-github	3 months ago
.gitignore	Build UMD dist file with Rollup	10 months ago
.travis.yml	Update npm API key	6 months ago
CODE_OF_CONDUCT.md	Fix code of conduct contact email	6 months ago
CONTRIBUTING.md	Clarify what parts of the standard we don't want to implement	5 months ago
LICENSE	2016	3 years ago
Makefile	Add flow definitions	6 months ago
README.md	Add note about UMD dist file	5 months ago
bower.json	Remove Bower dependency in development	2 years ago


JSON

=

JavaScript Object Notation

r/javascript

Posts



Join the discussion

BECOME A REDDITOR

↑


83


↓


Posted by u/rodrigo-silveira 4 years ago

How to pronounce "JSON"?


I've heard people pronounce it so many ways, it's sometimes almost difficult to know what they're talking about. According to Douglas Crockford, however, this is how it's done: <https://www.youtube.com/watch?v=zhVdWQWKRqM>
(PS: Crockford is the guy behind JSON, so whatever he says is definite and absolute truth =)

 189 Comments

 Share

 Save

81% Upvoted

 This thread is archived

New comments cannot be posted and votes cannot be cast

SORT BY

BEST

▼

↑

ChaseMoskal

186 points · 4 years ago

↓

"Jay-sawn"

Share Report Save

↑

stessie1fire

11 points · 4 years ago

COMMUNITY DETAILS

 r/javascript

527k
Subscribers

2.7k
Online

All about the JavaScript programming language!

SUBSCRIBE

CREATE POST

ADVERTISEMENT

Notes :

https://www.reddit.com/r/javascript/comments/2mapkc/how_to_pronounce_json/
(https://www.reddit.com/r/javascript/comments/2mapkc/how_to_pronounce_json/)

JSON : LÉGER & FACILE À :

- parser : JSON.parse()
- encoder :
JSON.stringify()

```
const str:string = `{
  "name": "Petyr Baelish",
  "killer": "Arya Stark",
  "episodeOfDeath": {
    "season": 7,
    "episode": 7
  },
}`;

const character:any = JSON.parse( str ); // message est un objet JS
console.log( character.episodeOfDeath.season );
```

FETCH & JSON

response.text() + JSON.parse() **VS** response.json()

```
fetch('api/users')  
  .then( (response:Response) => response.json() )  
  .then( (data:any) => console.log(data.id) )
```


SAME ORIGIN POLICY : ??

- un script d'un domaine A ne peut pas accéder à des données/scripts d'un domaine B
- solution : CORS
 - browser envoie requête http "OPTIONS"
 - serveur retourne un header http

```
Access-Control-Allow-Origin: <domaines-autorisés>
```

- browser effectue ou non la requête AJAX

Notes :

source : https://en.wikipedia.org/wiki/Cross-origin_resource_sharing (https://en.wikipedia.org/wiki/Cross-origin_resource_sharing)



enable cross-origin resource sharing

Cross-Origin Resource Sharing (CORS) is a [specification](#) that enables truly open access across domain-boundaries. If you serve public content, please consider using CORS to open it up for universal JavaScript/browser access.

[\[more...\]](#)

Why is CORS important?

JavaScript and the web programming has grown by leaps and bounds over the years, but the [same-origin policy](#) still remains. This prevents JavaScript from making requests across domain boundaries, and has spawned various hacks for making cross-domain requests.

CORS introduces a standard mechanism that can be used by all browsers for implementing cross-domain requests. The spec defines a set of headers that allow the browser and server to communicate about which requests are (and are not) allowed. CORS continues the spirit of the open web by bringing API access to all.

Notes :

<https://enable-cors.org/> (<https://enable-cors.org/>)

JS

1. Rappels
2. Premiers pas en JS
3. La POO en JS
4. Les modules
5. Le typage
6. API DOM
7. Ajax
- 8. jQuery**
9. bonus track

- Qu'est ce que c'est ?
- Manipulation du DOM
- Gestion des événements
- Appels Ajax avec jQuery
- Alternatives

PRÉSENTATION DE JQUERY ^{3.3}

- Librairie qui simplifie :
 - la manipulation du DOM,
 - la gestion d'événements,
 - la création d'animations,
 - l'utilisation d'Ajax
- Gère la compatibilité multi-navigateurs

Notes :

jQuery est une librairie JavaScript permettant de manipuler une page HTML affichée dans un navigateur. La librairie possède également des méthodes permettant de gérer les actions de l'utilisateur. Il est également possible de créer des animations avec jQuery. De même la librairie permet d'effectuer des appels Ajax afin de mettre à jour l'affichage de la page sans rechargement complet.

PRÉSENTATION DE JQUERY

3.4

jQuery n'est pas un framework :

- ni un framework architectural comme React/Redux, Angular ou Backbone.JS
- ni un framework UI / frontend comme bootstrap

npm EnterpriseProductsSolutionsResourcesDocsSupport

npm

JoinLog In

keywords:jquery-plugin

Search

1574 packages found

1

2

3

...

79

»

Sort Packages

Optimal

Popularity

Quality

Maintenance

Who's Hiring?

MuleSoft, Tinder, Airtable and lots of other companies are hiring javascript developers.

See all 19 companies

js-cookie

p

q

Notes :

jQuery dispoe d'une quantité importante de plugins (lightbox, carousel, gestion de formulaire, etc.) :
<https://www.npmjs.com/browse/keyword/jquery-plugin> (<https://www.npmjs.com/browse/keyword/jquery-plugin>)

Si une fonctionnalité n’est pas couverte de base par jQuery vous pourrez certainement utiliser un plugin, ou écrire le votre !

Une fonction magique : \$(...)

Permet de sélectionner un (ou des) élément à partir d'un sélecteur CSS (équivalent `querySelector`)...

... et de lui (ou leur) appliquer des transformations à l'aide de différentes méthodes :

- `$(monSelecteur).html (http://api.jquery.com/html/) (...)`
- `$(monSelecteur).show (http://api.jquery.com/show/) (...)`
- `$(monSelecteur).hide (http://api.jquery.com/hide/) (...)`
- `$(monSelecteur).css (http://api.jquery.com/css/) (...)`
- ... `http://api.jquery.com/ (http://api.jquery.com/)`

Notes :

Avec jQuery tout passe par l'objet "\$". Cet objet contient le code de jQuery et s'utilise principalement comme une fonction permettant de sélectionner un ou plusieurs éléments dans la page html. Le résultat de l'appel à la fonction `$()` est lui même un objet permettant de manipuler les noeuds html qui correspondent au sélecteur qui a été passé en paramètre.

MANIPULATION DU DOM

```
<div id="king">Joffrey</div>

<script src="jquery.js"></script>
<script>
    $( '#king' ).html( 'Tommen' );
</script>
```

SlimCSSJSResult

38

```
$('.show').click(  
  function(){ $('div').show(); }  
);  
  
$('.hide').click(  
  function(){ $('div').hide(); }  
);  
  
$('.showWithDuration').click(  
  function(){ $('div').show(2000); }  
);  
  
$('.hideWithDuration').click(  
  function(){ $('div').hide(2000); }  
);
```

\$('.div').show();

\$('.div').hide();

\$('.div').show(2000);

\$('.div').hide(2000);

Resources

1x0.5x0.25x

Rerun

Notes :

<https://codepen.io/kumquats/pen/RrgZoJ/> (<https://codepen.io/kumquats/pen/RrgZoJ/>)

CHAINAGE DES MÉTHODES

Lorsqu'on appelle plusieurs méthodes sur un même objet jQuery, il est possible d'appliquer le principe du chainage :

```
$('.whiteWalker').show();  
$('.whiteWalker').addClass('blueEyes');  
$('.whiteWalker').click(goToTheWall);
```



```
$('.whiteWalker').show().addClass('blueEyes').click(goToTheWall);
```

Notes :

L'avantage de chaîner les méthodes réside principalement dans l'optimisation des performances puisque le sélecteur jQuery n'est interprété qu'une seule fois dans le deuxième exemple contre trois fois dans le premier. Attention cependant à la lisibilité du code, même si des sauts à la ligne peuvent aider à la compréhension. Il faut également noter que le debug peut être plus difficile dans le cas d'appels chaînés.

\$(DOCUMENT).READY()

- callback exécuté une fois le DOM prêt à être manipulé
- n'attend pas le chargement des images & iframes
- peut être omis si defer ou JS en fin de body
- utile pour les balises <script> sans src

```
$(document).ready( function() {  
    // le code javascript de mon app ici  
});
```

```
$(document).ready( init );
```

```
$( init );
```

Notes :

<http://codepen.io/kumquats/pen/jrBPry> (<http://codepen.io/kumquats/pen/jrBPry>)

Download (<https://jquery.com/doc>)

Blog (<https://blog.jquery.com/>)

Browser Support (<https://jquery.c>)

Ajax ([//api.jquery.com/category/ajax](https://api.jquery.com/category/ajax))

Global Ajax Event Handlers
([//api.jquery.com/category/ajax-ajax-event-handlers/](https://api.jquery.com/category/ajax-ajax-event-handlers/))

Helper Functions
([//api.jquery.com/category/ajax-functions/](https://api.jquery.com/category/ajax-functions/))

Low-Level Interface
([//api.jquery.com/category/ajax-level-interface/](https://api.jquery.com/category/ajax-level-interface/))

Shorthand Methods
([//api.jquery.com/category/ajax-methods/](https://api.jquery.com/category/ajax-methods/))

Notes :

<http://api.jquery.com/animate/> (<http://api.jquery.com/animate/>)

Slim

CSS

JS

Result

312

```
$('.animate1').click( function(){
    $('div').animate({width: 400})
    .animate({height: 100}, 500)
    .animate({opacity: 0.5}, 200);
} );

$('.animate2').click( function(){
    $('div').animate({opacity: 1}, 200)
    .animate({height: 300}, 500)
    .animate({width: 200});
} );
```

animate1()

animate2()



Resources

1x

0.5x

0.25x

Rerun

Notes :

<https://codepen.io/kumquats/pen/mVwMGv/> (<https://codepen.io/kumquats/pen/mVwMGv/>)

GESTION DES ÉVÉNEMENTS 3.13

Ajout d'écouteurs d'événement via :

- `.on()` (<http://api.jquery.com/on/>)
- `.one()` (<http://api.jquery.com/one/>)
- `.click()` (<http://api.jquery.com/click/>)
- `.hover()` (<http://api.jquery.com/hover/>)
- `.submit()`
(<http://api.jquery.com/submit/>)

Notes :

La détection des actions de l'utilisateur passe par l'ajout d'écouteurs d'événements. jQuery met à disposition du développeur plusieurs méthodes pour cela. Elles s'utilisent de la manière suivante : `$('selector').method('eventType', eventHandler);` `.on()` est depuis la version 1.7 de jQuery la méthode recommandée pour l'ajout d'écouteurs génériques. A ne pas confondre avec `.one()` qui permet d'ajouter un écouteur d'événement unique (qui se supprime dès qu'un événement est capté). Les autres méthodes `.bind()`, `.delegate()`, et `.live()` sont aujourd'hui dépréciées et conservées pour des raisons de rétrocompatibilité.

HTML

CSS

JS

Result

3.14

```
function showDragons( event ){
  event.preventDefault();
  $( '.dragon' ).css( 'display', 'block' );
}

$('#letItBurn').click( showDragons );
```

[Où sont mes dragons ?](#)

Resources

1x 0.5x 0.25x

Rerun

Notes :

<http://codepen.io/kumquats/pen/VjPjpp/> (<http://codepen.io/kumquats/pen/VjPjpp/>)

JQUERY.AJAX()

- Permet d'effectuer des requêtes Ajax
- Se configure à l'aide d'un [objet settings](http://api.jquery.com/jquery.ajax/#jQuery-ajax-settings) contenant des paires clé/valeur (<http://api.jquery.com/jquery.ajax/#jQuery-ajax-settings>)
- Promises ou callbacks qui s'exécutent en cas de succès ou d'erreur

Notes :

AJAX est un des principaux apports de JavaScript, cette technique permet de charger des données sans recharger la page complète. Il est donc possible d'acquérir de nouvelles informations depuis le serveur afin de les afficher dans la page sans la rafraichir. jQuery permet de simplifier l'utilisation d'Ajax et notamment de gérer les actions en cas de succès ou d'erreur lors de l'appel au serveur. Les promises sont désormais la bonne pratique pour la gestion des retours à la place des callbacks renseignés dans l'objet settings (cf.

<http://stackoverflow.com/questions/14754619/jquery-ajax-success-callback-function-definition/14754681#14754681> (<http://stackoverflow.com/questions/14754619/jquery-ajax-success-callback-function-definition/14754681#14754681>))

Download (<https://jquery.com/download>)

Blog (<https://blog.jquery.com/>)

Browser Support (<https://jquery.com/browser-support>)

Ajax (<https://api.jquery.com/category/ajax>)

Global Ajax Event Handlers
(<https://api.jquery.com/category/ajax-event-handlers/>)

Helper Functions
(<https://api.jquery.com/category/ajax-functions/>)

Low-Level Interface
(<https://api.jquery.com/category/ajax-level-interface/>)

Shorthand Methods
(<https://api.jquery.com/category/ajax-methods/>)

Notes :

<http://api.jquery.com/jQuery.ajax/> (<http://api.jquery.com/jQuery.ajax/>)

HTMLJS

Result

317

```
$.ajax('https://enable-cors.org/').done( function( data ){
    $('#container').html( data );
});

//ou
/*
$.ajax({
    url:'http://enable-cors.org/',
    success: function( data ){
        $('#container').html( data );
    }
});
*/
```

Enable CORS

- [Home](#)
- [Server](#)
- [Client](#)
- [Resources](#)
- [Test](#)
-
- [Suggest!](#)
- [Help!](#)

enable cross-origin resource sharing

Cross-Origin Resource Sharing (CORS) is a [specification](#) that enables truly open access across domain-boundaries. If you serve public content, please consider using CORS to open it up for universal JavaScript/browser access.

[\[more...\]](#)

Why is CORS important?

JavaScript and the web programming has grown by leaps and bounds over the years, but the [same-origin policy](#) still remains. This prevents JavaScript from making requests across domain boundaries, and has spawned various hacks for making cross-domain requests.

CORS introduces a standard mechanism that can be used by all browsers for implementing cross-domain requests. The spec defines a set of headers that allow the browser and server to communicate about

Resources

1x0.5x0.25x

Rerun

Notes :

<http://codepen.io/kumquats/pen/vKgKde/> (<http://codepen.io/kumquats/pen/vKgKde/>)

JQUERY



POURQUOI SE PASSER DE JQUERY ?

- temps de chargement et d'exécution
- rend notre code dépendant
- parfois surdimensionné
- parfois pas nécessaire...

Notes :

Ressources pour alimenter la réflexion : <http://alistapart.com/blog/post/choosing-vanilla-javascript> (<http://alistapart.com/blog/post/choosing-vanilla-javascript>) <https://gomakethings.com/ditching-jquery-for-vanilla-js/> (<https://gomakethings.com/ditching-jquery-for-vanilla-js/>) <https://speakerdeck.com/rem/i-know-jquery-now-what> (<https://speakerdeck.com/rem/i-know-jquery-now-what>)

ALTERNATIVES

ALTERNATIVES

YOU MIGHT NOT NEED JQUERY

jQuery and its cousins are great, and by all means use them if it makes it easier to develop your application.

If you're developing a library on the other hand, please take a moment to consider if you actually need jQuery as a dependency. Maybe you can include a few lines of utility code, and forgo the requirement. If you're only targeting more modern browsers, you might not need anything more than what the browser ships with.

At the very least, make sure you know what [jQuery is doing for you](#), and what it's not. Some developers believe that jQuery is protecting us from a great demon of browser incompatibility when, in truth, post-IE8, browsers are pretty easy to deal with on their own.

Tweet "You might not need jQuery"

0

Star YouMightNotNeedjQuery on Github

11,14

Notes :

<http://youmightnotneedjquery.com/> (<http://youmightnotneedjquery.com/>)

microjs

I need ...

Tweet

DOMBrew
27 2

Clean API high performance DOM
builder

1 kB

Pass-Meter.js
16 4

Simple password strength testing.

0.7 kB

Smoothie Charts
1,917 201

Smoooooooooth JavaScript
charts for realtime streaming
data

4 kB

Plates
814 65

Plates (short for templates) binds data to
markup. Plates has NO special syntax. It
works in the browser and in Node.js.

2.3 kB

Notes :

<http://microjs.com> (<http://microjs.com>)

POUR OU CONTRE ?

Pour	Contre
Simplicité	Performances
Compatibilité	Dépendance
Documentation	Apprentissage
Maintenue & testée	Overkill ?
Communauté	
Plugins	Plugins...

Notes :

jQuery n'est parfois pas nécessaire mais si l'on décide de s'en passer il faut quant même s'assurer que le code produit est compatible avec les navigateurs ciblés. cf.

https://docs.google.com/document/d/1LPaPA30bLUB_pubLIMF0RlhdnPxePXm7oW02iiT6o/edit?pref=2&pli=1

(https://docs.google.com/document/d/1LPaPA30bLUB_pubLIMF0RlhdnPxePXm7oW02iiT6o/edit?pref=2&pli=1)

MERCI !

MERCI !