

# Introduction à jQuery en 5 minutes

<http://jquery.com>

# Introduction à jQuery en 10 minutes

```
$('#myNode').find('li :visible')
  .not(':gt(4)')
  .find('a[href$=pdf]')
  .filter(':contains(ouch)').each(function () {
    $(this).css("font-size", "150%");
  });
```

- Faciliter les besoins inhérents à la programmation Javascript / Web
  - Manipulation du DOM
  - Manipulation de la CSS
  - Gestion des événements
  - Appels Ajax
  - Écriture de widgets

- Le navigateur fournit quelques primitives simples pour ça :  
    `document.getElementById()`  
    `document.getElementsByTagName()`  
    Spécifications DOM (*getAttribute*, *childNodes*, etc.)  
    ... (mais pas beaucoup plus)
- Certains navigateurs fournissent plus mais ce n'est pas standard

```

$('#domid'); // => document.getElementById('domid');
$('div'); // => document.getElementsByTagName('div');

// les div avec la classe CSS myclass
$('div.myclass');
// tous les div pour lesquels l'attribut attr = value
$('div[attr=value]');
// tous les span dans un div
$('div span');
// tous les input de type hidden dans le formulaire myForm
$('#myForm input:hidden');
  
```

Il y a des sélecteurs pour faire tout et n'importe quoi :

<http://api.jquery.com/category/selectors/>

```
// Getter
// Lire la valeur de l'attribut action du formulaire myForm
$('#myForm').attr('action')
"http://www.cubicweb.org/edit"

// Setter
// Changer la valeur de l'attribut action du formulaire myForm
$('#myForm').attr('action', 'http://thenewsubmiturl')
```

```
>>> $('#entityForm').children()  
[div#progress, fieldset, iframe#eformframe]
```

```
>>> $('#entityForm').parent()  
[div.formBody]
```

```
// Getter
>>> $('table.listing').css('background-color')
"transparent"

// Setter
>>> $('table.listing').css('background-color', 'red')
```



- La sélection de noeuds renvoie un objet *jQuery* qui peut se manipuler comme une collection de noeuds DOM :

```
>>> $forms = $('form')
[form view, form#pageContentForm, form view]
>>> $forms.length
3
>>> $forms[0]
<form action="http://intranet.logilab.fr/crm/view">
```

- On peut chaîner le résultat avec d'autres fonctions *jQuery*

```
>>> function getId() {return this.id;}
>>> $('form').find('input').map(getId)
["norql", "", "rqlboxsubmit", "rql"]
```

```
function myCallback(event) { alert('hello !'); }
```

- La forme générale avec utilisation de la fonction *bind*

```
$('#mydiv').bind('click', myCallback)
```

- Pour les cas courants, il y a des raccourcis :

```
$('#mydiv').click(myCallback);  
$('#mydiv').dblclick(myCallback);  
$('#mydiv').blur(myCallback);  
$('form input:radio').click(myCallback)
```

- générer un événement :

```
$('.li.active').trigger('click');  
// ou ...  
$('.li.active').click();  
$('.li.active').trigger('custom-event-would-work-too');  
  
MyObj = {};  
$(MyObj).trigger('event-triggerable-on-anything');
```

```
$.ajax({ url: "http://cubicweb.org/myresource",  
        success: function(data) {  
            alert("Here's what I got " + data);  
        }  
        // + plein d'autres options  
    });
```

Pour voir toutes les options : <http://api.jquery.com/category/ajax/>

- Disponibles sous forme de *plugins*
- L'utilisation est globalement toujours la même
  - Le plugin rajoute une fonction disponible sur l'objet "collection" *jQuery*
  - On sélectionne le noeud qui nous intéresse et on appelle la fonction fournie par le *plugin*

```
$( '#myInput' ).autocomplete(  
    {url: "http://the-url/resource"} )
```

- Une kyrielle de plugins officiels : <http://plugins.jquery.com/>
- Servi par google :

```
google.load("jquery", "1.4.2");
```