

JavaScript

M4103C - Programmation Web – client riche

2ème année - S4, cours - 3/4
2017-2018

Marcel.Bosc@iutv.univ-paris13.fr

Département informatique

IUT de Villetaneuse

Université Paris-13

Table des matières

- AJAX
- JSON
- DOM

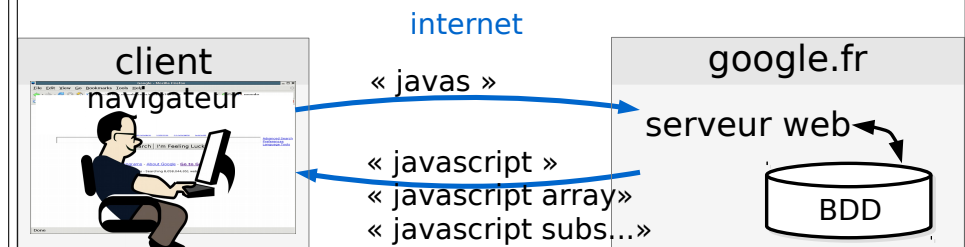
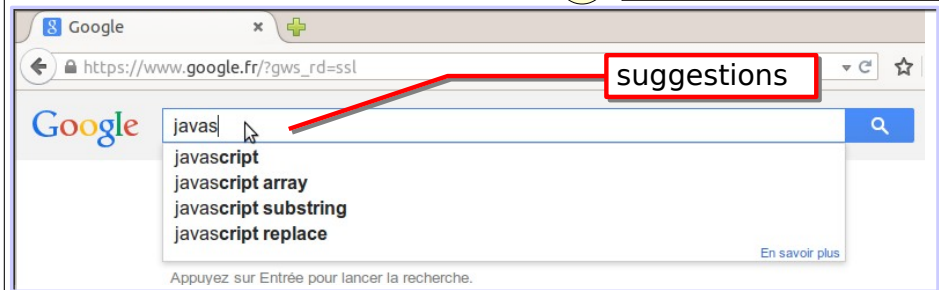
1ère partie

Ajax

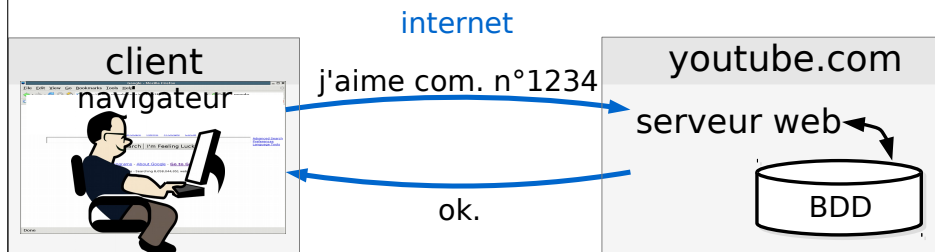
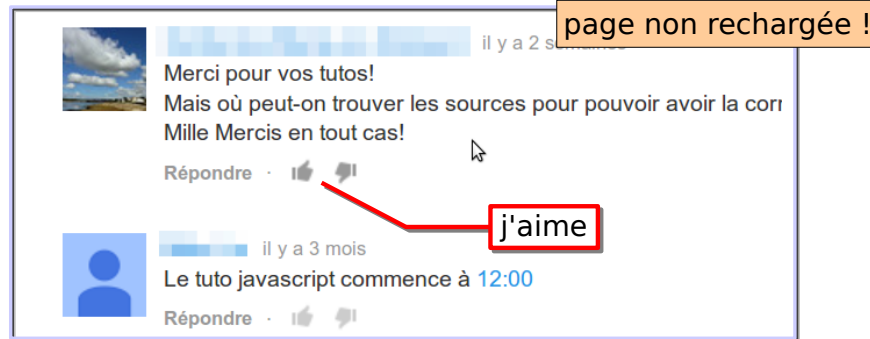
Exemple : Google



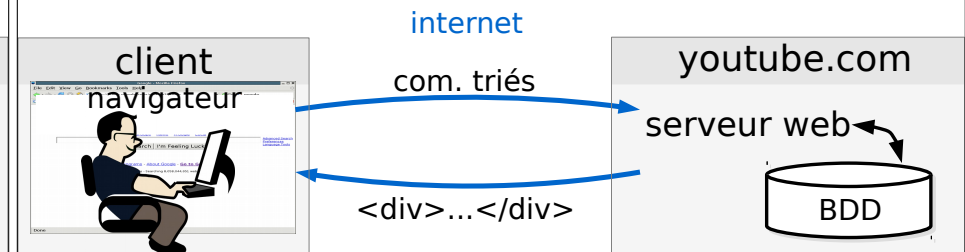
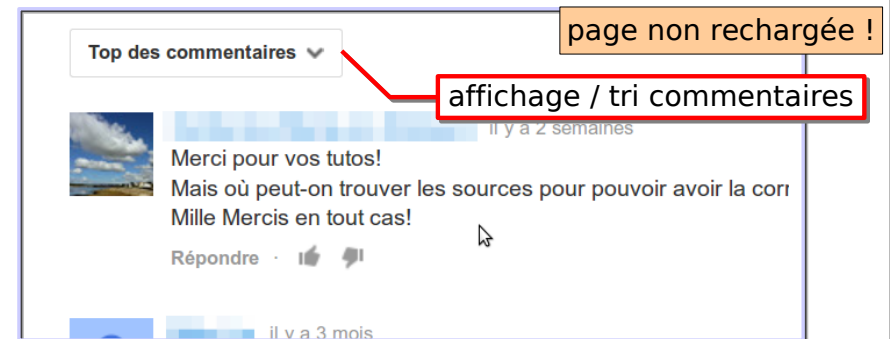
page non rechargée !



Exemple : commentaires



Exemple : commentaires



Ajax

Asynchronous JavaScript and ~~XML~~

JavaScript Asynchrone

Requête non bloquante à partir du JavaScript, au serveur, sans recharger la page.

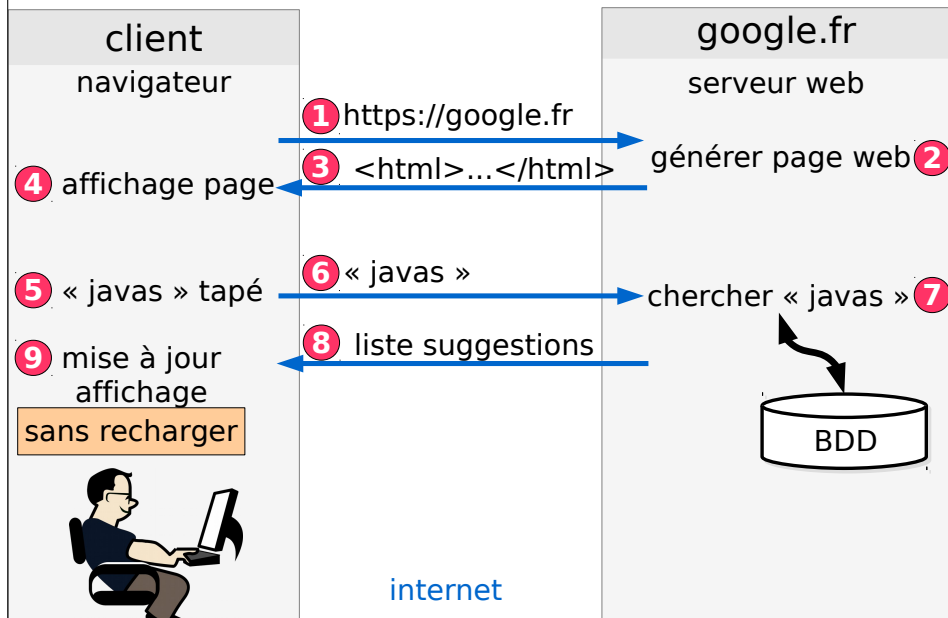
Exemple jQuery .get()

```

$.get("http://exemple.org/commentaire",
    { id: 5678 },
    function(reponse) {
        // afficher com.
    });
  
```

appelée à la réception de la réponse du serveur

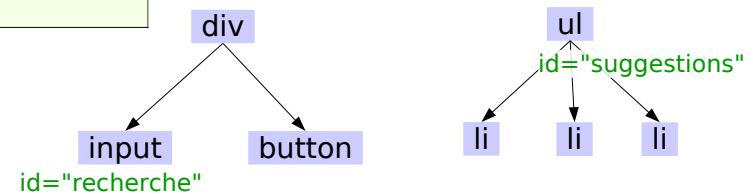
Chronologie client / serveur



Exemple : suggestion

A search input field with the text 'a' and a 'chercher' button. Below the input field, a list of suggestions is displayed: 'abricot', 'arbre', and 'amis'.

```
<div>
  <input id="recherche" type="text" />
  <button>chercher</button>
</div>
<ul id="suggestions">
  <li></li>
</ul>
```



Suggestion : JS

A search input field with the text 'a' and a 'chercher' button. Below the input field, a list of suggestions is displayed: 'abricot', 'arbre', and 'amis'.

JS

```
0 $('#recherche').keyup(function(e)
1 {
2   $.get('http://exemple.org/suggestion.php',
3     {mot: $('#recherche').val()},
4     function(reponse)
5     {
6       $('#suggestions').html(reponse);
7       $('#suggestions').show();
8     });
9 });
```

serveur

```
<li>abricot</li>
<li>arbre </li>
<li>amis </li>
```

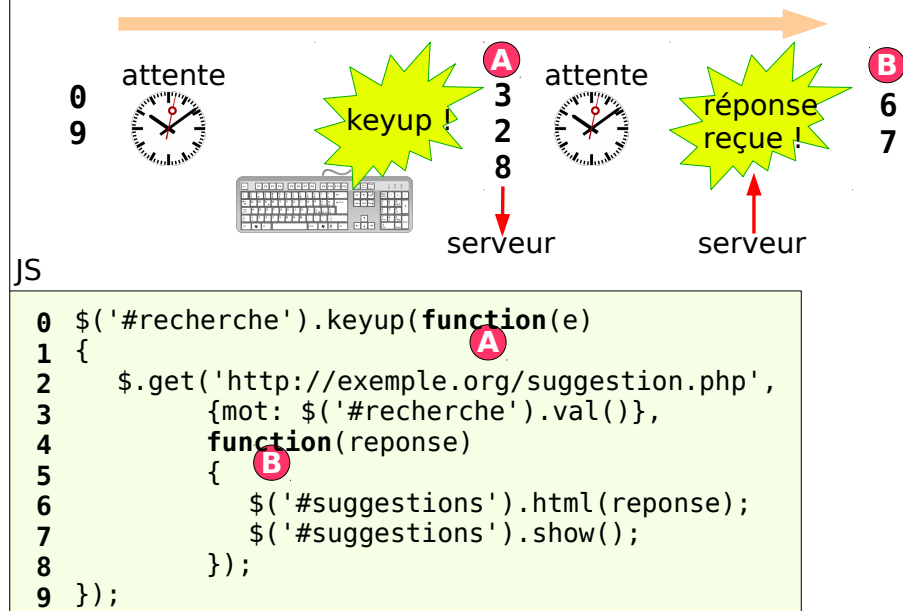
Suggestion : GET

`http://exemple.org/suggestion.php?mot=a`

JS

```
0 $('#recherche').keyup(function(e)
1 {
2   $.get('http://exemple.org/suggestion.php',
3     {mot: $('#recherche').val()},
4     function(reponse)
5     {
6       $('#suggestions').html(reponse);
7       $('#suggestions').show();
8     });
9 });
```

Suggestion : chronologie JS



Suggestion : PHP

<http://exemple.org/suggestion.php?mot=a>

```

0 $mot=$_GET['mot'];
1 $sql="SELECT mot FROM mots WHERE mot LIKE '?%'";
2 $suggestions=database_list($sql,$mot);
3 $resultat='';
4 foreach($suggestions as $suggestion)
5 {
6   $resultat.='<li>'.htmlentities($suggestion).'</li>';
7 }
8 echo $resultat;
  
```



```

<li>abricot</li>
<li>arbre </li>
<li>amis </li>
  
```

Méthode GET

GET: peut-être répétée sans conséquences
(ne modifie pas l'état sur le serveur)

"Lire infos sur le serveur"

<http://exemple.org/recherche.php?mot=jeudi>

Exemples:
modifier l'affichage
faire une recherche

~~Contre-exemples:
payer en ligne
ajouter un commentaire
sur un forum~~

Méthode POST

POST: répétition potentiellement gênante
(change l'état du serveur)

"Écrire des infos sur le serveur"

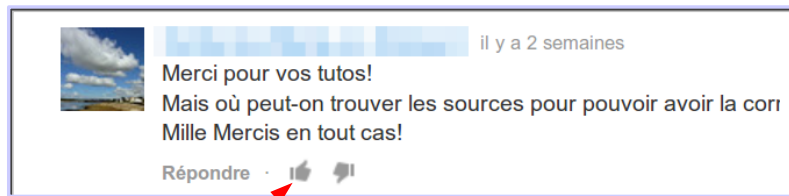
<http://exemple.org/payer.php> ~~X~~

Entêtes http

~~Contre-exemples:
modifier l'affichage
faire une recherche~~

Exemples:
payer en ligne
ajouter un commentaire
sur un forum

Exemple : .post()



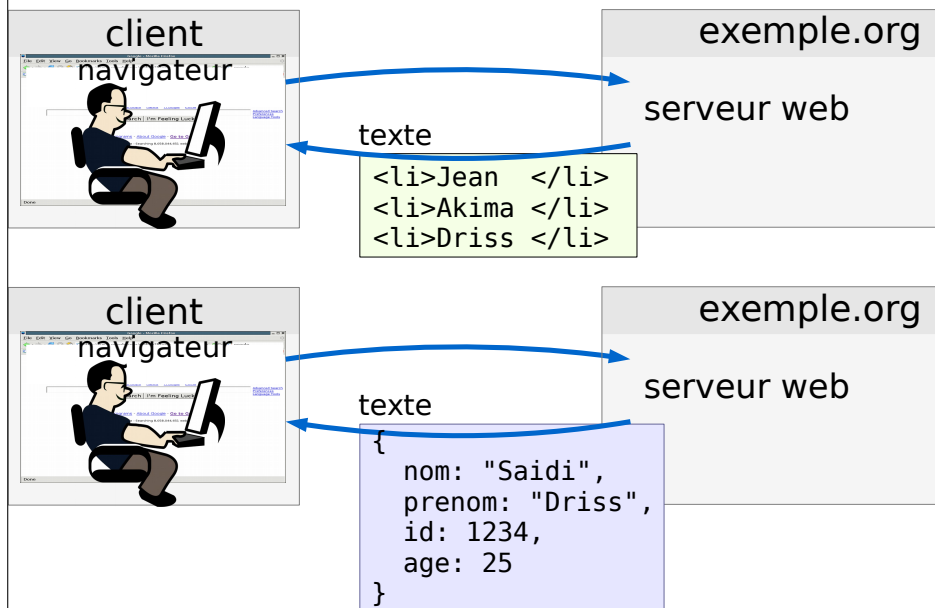
JS

```
0 $('jaime').click(function(e)
1 {
2     $.post('http://exemple.org/jaime.php',
3         {nbCom: $(this).parent().attr('id')},
4         function(reponse)
5         {
6             ...
7         });
8 });
9
```

2ème partie

JSON

HTML / Données



JSON

JavaScript Object Notation

JSON

Format de fichier texte, utilisant la syntaxe JavaScript pour représenter des données (objets, tableaux ...)

très utilisé !



beaucoup de langages PHP

JSON : exemples

Objet simple

```
{
  nom: "Saidi",
  prenom: "Driss",
  id: 1234,
  age: 25
}
```

Objet complexe

```
{
  nom: "Collège Grange du Bois",
  ville:
    {
      nom: "Savigny-le-Temple",
      "nom-court": "Savigny",
      code : 77176
    },
  adresse: "2 av. Victor..."
}
```

Tableau simple

```
[
  "Fraise",
  "Chocolat",
  "vanille"
]
```

Tableau d'objets

```
[
  { nom: "Wang",
    id : 4321 },
  { nom: "Amara",
    id : 5612 }
]
```

PHP: json_encode()

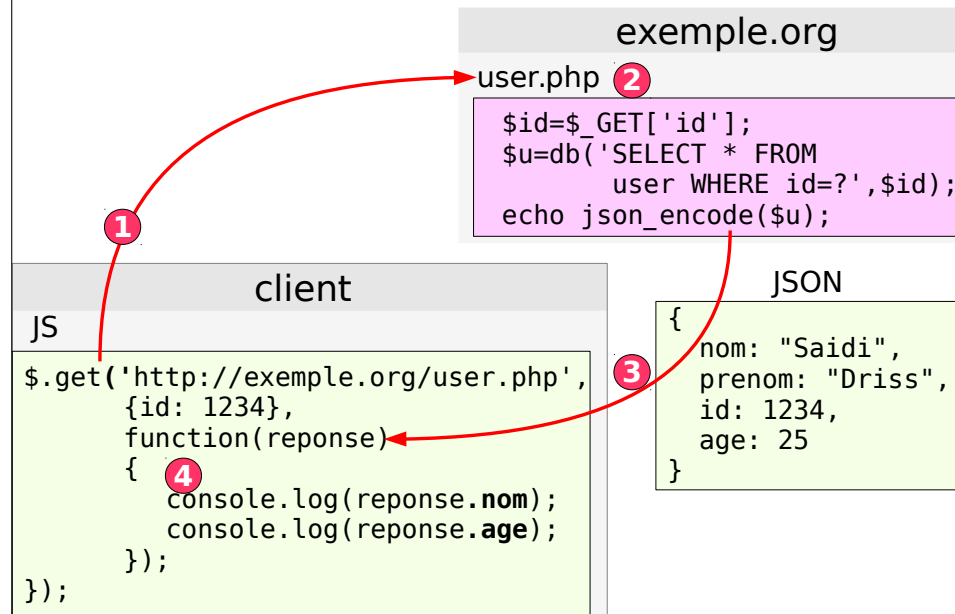
PHP

```
$user=[
  'nom'    =>'Saidi',
  'prenom'=>'Driss',
  'id'     =>1234,
];
$user['age']=25;
header('Content-Type: application/json');
echo json_encode($user);
```

JSON

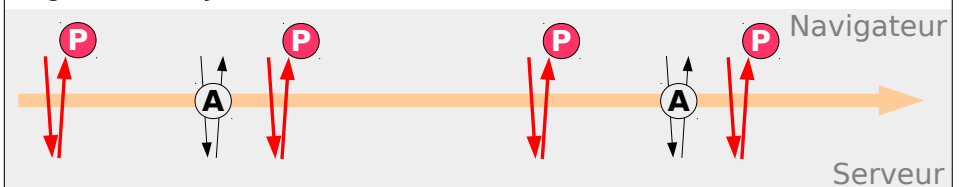
```
{
  nom: "Saidi",
  prenom: "Driss",
  id: 1234,
  age: 25
}
```

JS : réponse JSON

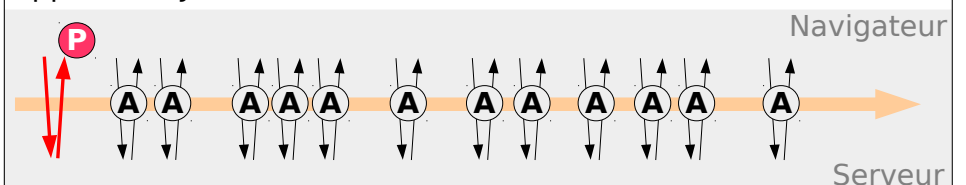


Page + JS vs application

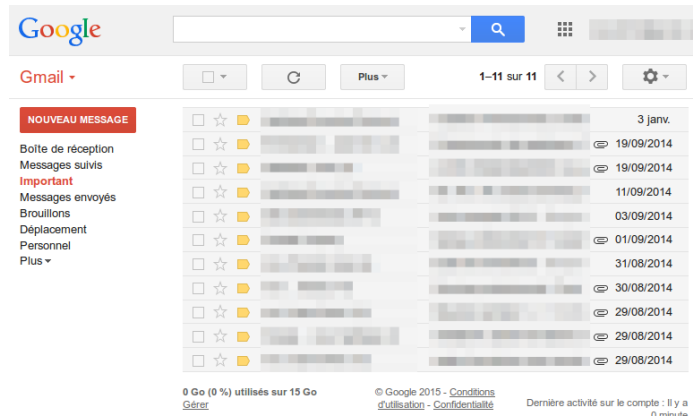
Pages web + JS



Application JS



Application JS



Frameworks : Angular, React, ...

3ème partie

DOM

jQuery vs DOM

jQuery

Simplicité 😊
Comptabilité navigateurs

DOM

Performance chargement
Performance exécution
Propriétés
Text Node
Cookies
Env. particuliers

Navigateurs



Chrome



63%

Mozilla Firefox



15%

Explorer + Edge



13%

Safari



5%

Mobile



50%

Bureau

Compatibilité

caniuse.com

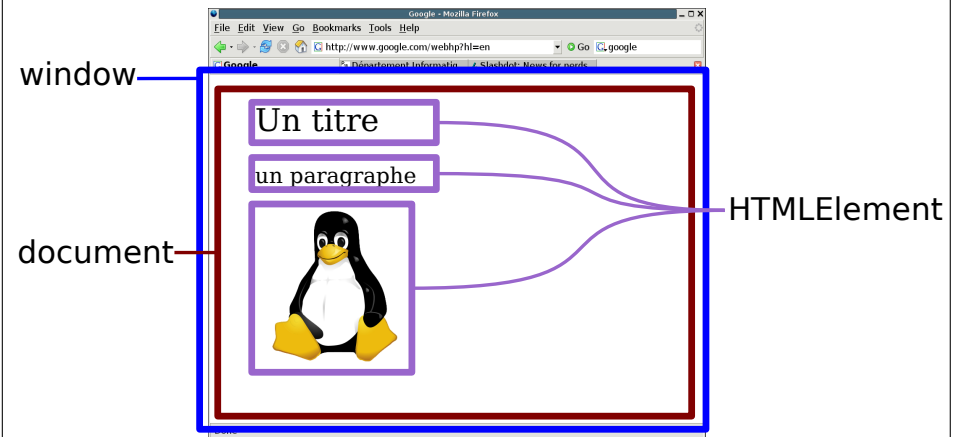
querySelector/querySelectorAll - REC Global

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *
		31				
		35				
		36				
8		37				
9	34	38				
10	35	39	7.1	26	7.1	
11	36	40	8	27	8.1	8

MDN : developer.mozilla.org

Ordinateur	Mobile
Fonctionnalité	Chrome Firefox (Gecko) Internet Explorer Opera Safari (WebKit)
Support de base	1 3.5 (1.9.1) 8 10 3.2 (525.3)
	bug 416317 WebKit bug 16587

Principaux objets DOM

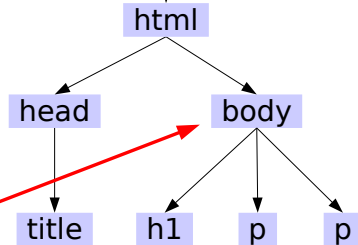


Window & Document

window

```
window.document
window.location
window.alert()
window.console()
...
```

document



document

```
document.body
document.getElementById()
document.createElement()
document.cookie
...
```

document.getElementById()

d = un élément DOM

DOM

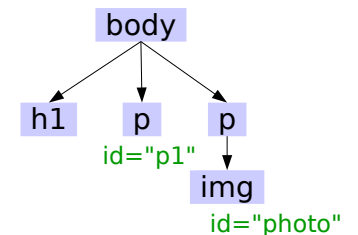
```
var d=document.getElementById('photo');
```

j=« liste » d'un seul élément jQuery

jQuery

```
var j=$('#photo');
```

```
<body>
  <h1>Ceci est un titre</h1>
  <p id="p1">Un paragraphe</p>
  <p>2e paragraphe
    
</body>
```



document.createElement()

DOM

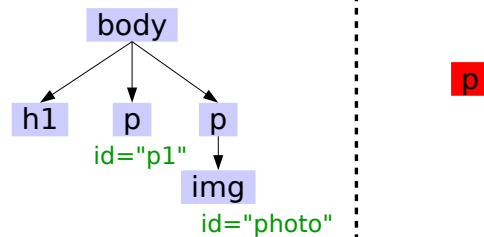
```
var d=document.createElement('p');
```

pas encore dans arbre DOM !

jQuery

```
var j=$('<p></p>');
```

pas encore dans arbre DOM !



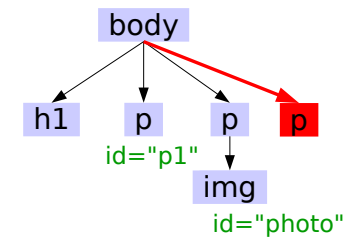
node.appendChild()

DOM

```
var d=document.createElement('p');
document.body.appendChild(d);
```

jQuery

```
var j=$('<p></p>');
$('body').append(j);
```



DOM & jQuery

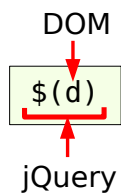
j=« liste » d'un seul élément jQuery

```
var j=$('#photo');
```

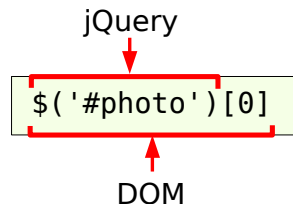
d = un élément DOM

```
var d=document.getElementById('photo');
```

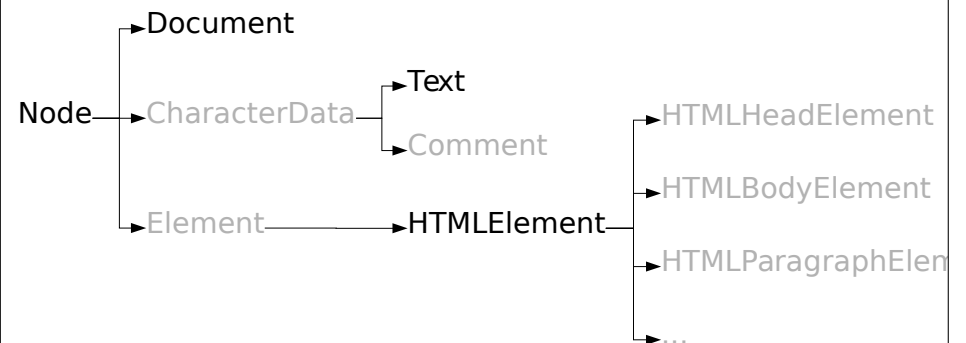
DOM → jQuery



jQuery → DOM



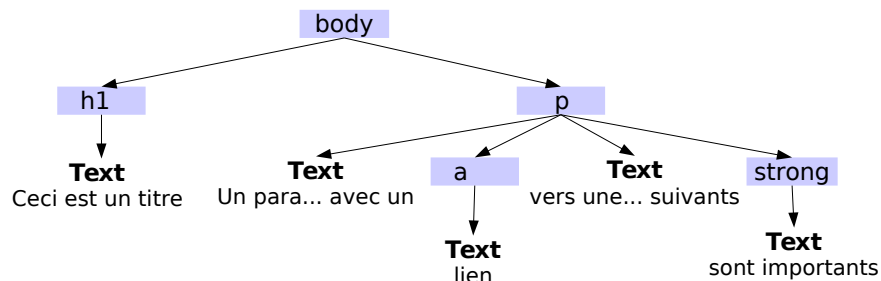
Node / HTMLElement



Simplifié

Text

```
<body>
  <h1>Ceci est un titre</h1>
  <p>
    Un paragraphe de texte avec un
    <a href="page2.html">lien</a> vers une autre
    page. Les mots suivants
    <strong>sont importants</strong>
  </p>
</body>
```



Node & HTMLElement

« Propriétés »

```
element.className
element.innerHTML
element.textContent
element.nodeName
element.nodeType
element.style
```

Arbre DOM

```
element.parentNode
element.children
element.childNodes
element.insertBefore()
element.removeChild()
```

```
element.addEventListener()
```

Chercher éléments

```
element.getElementsByClassName()
element.getElementsByTagName()
element.querySelector()
element.querySelectorAll()
```

element.className

```
d.className → "intro important"
```

DOM

```
d.className="conclusion secondaire"
```

```
j.attr('class')
```

jQuery

```
j.hasClass("intro")
```

```
j.addClass("xyz")
```

```
j.removeClass("intro")
```

```
<p class="intro important">Un paragraphe</p>
```

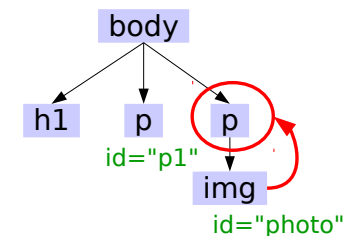
node.parentNode

DOM

```
var d=document.getElementById('photo').parentNode;
```

jQuery

```
var j=$('#photo').parent();
```



node.children

```
var liste=document.body.children;  
for(var i=0;i<liste.length;i++)  
{  
    // ... liste[i] ...  
}
```

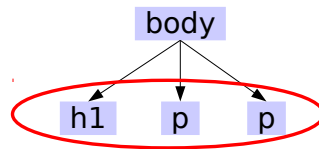
DOM

```
var liste=$('body').children();  
liste.each(function()  
{  
    // ... this ...  
});
```

jQuery

children / childNodes

Text



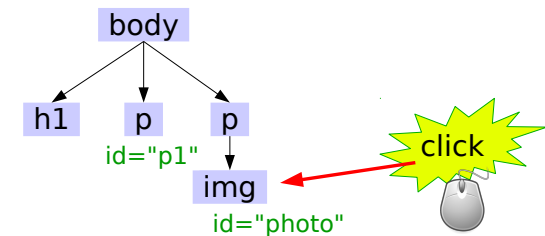
element.addEventListener()

```
var d=document.getElementById('photo');  
d.addEventListener('click',function(){ ... });
```

DOM

```
$('#photo').click(function(){ ... });
```

jQuery



Ce document est distribué librement.

Sous licence GNU FDL :

<http://www.gnu.org/copyleft/fdl.html>

Les originaux sont disponibles au format LibreOffice

<http://www-info.iutv.univ-paris13.fr/~bosc>