JavaScript

M4103C - Programmation Web – client riche

2ème année - S4, cours - 3/4
2018-2019

Marcel.Bosc@iutv.univ-paris13.fr

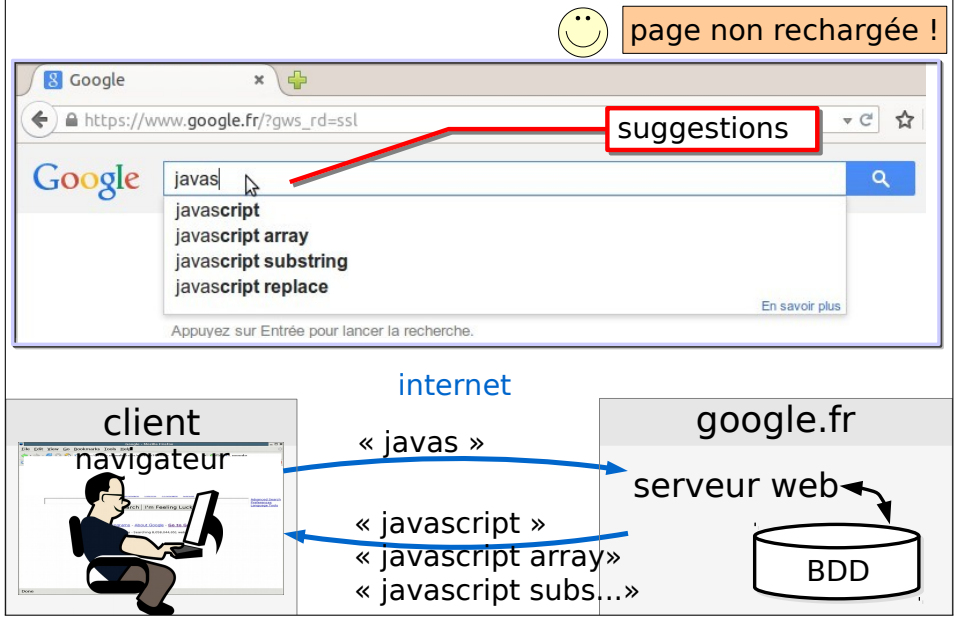
Département informatique IUT de Villetaneuse Université Paris-13

Table des matières

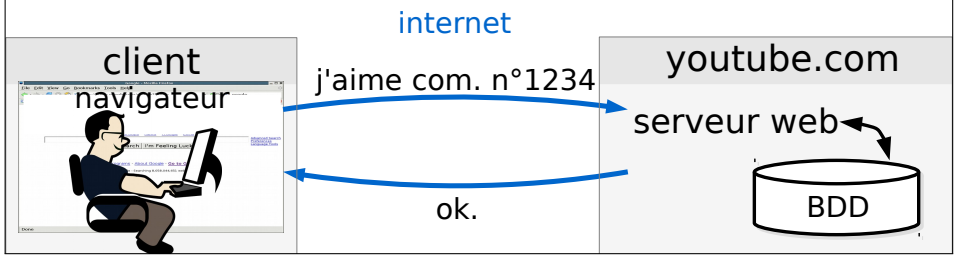
1ère partie

Ajax

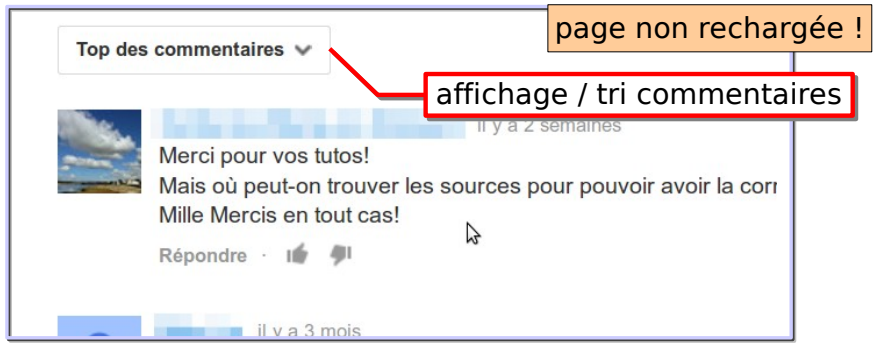
Exemple : Google

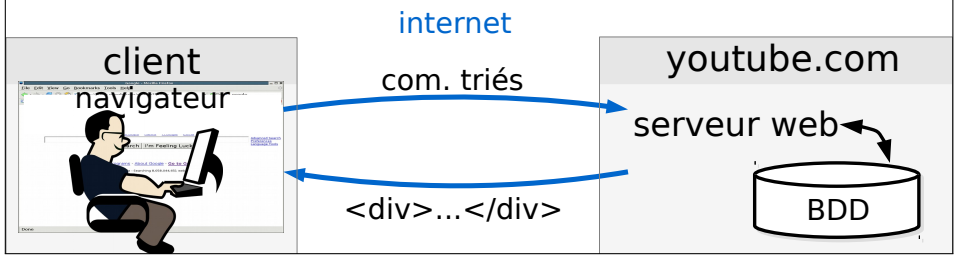


Exemple : commentaires



Exemple : commentaires





Ajax

Asynchronous **J**avaScript **a**nd ~~**X**ML~~

JavaScript Asynchrone

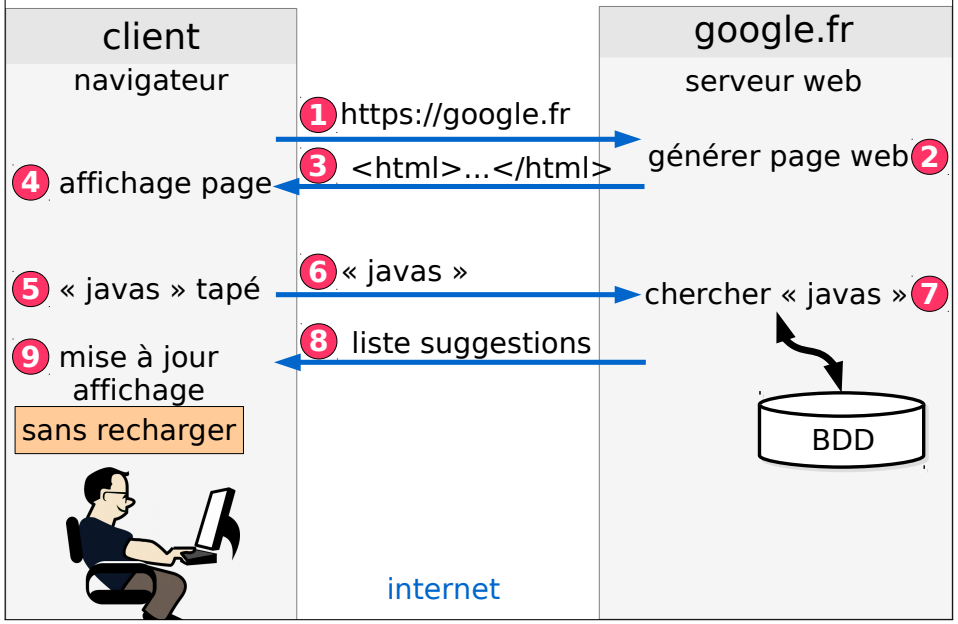
Requête non bloquante à partir du JavaScript, au serveur, sans recharger la page.

Exemple jQuery .get()

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

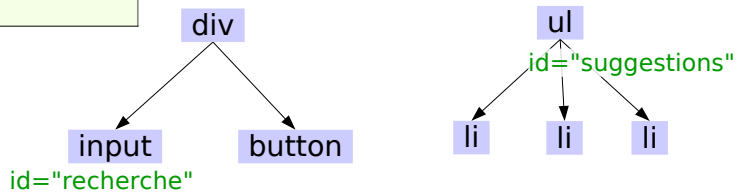
appelée à la réception de la réponse du serveur

Chronologie client / serveur

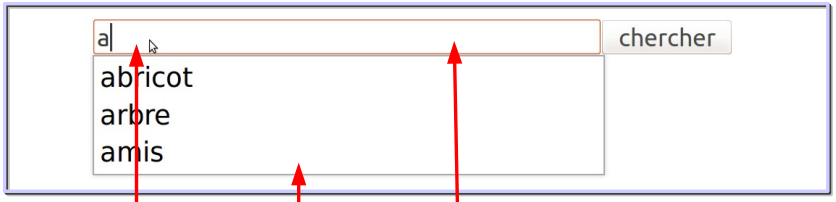


Exemple : suggestion


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



Suggestion : JS



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</l
<li>arbre  </l
<li>amis   </l
```

Suggestion : GET

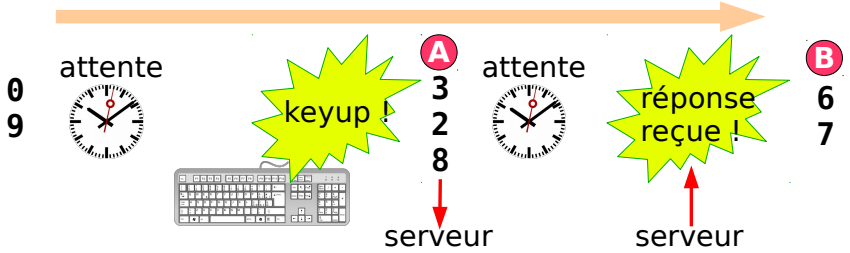
$_GET['mot']

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



JS

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

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;
```

BDD

```
<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"

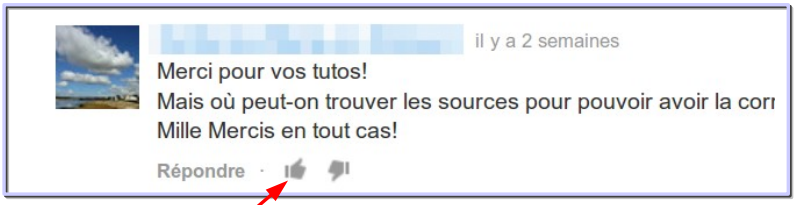
http://exemple.org/payer.php

Entêtes http

Contre-exemples:
modifier l'affichage
faire une recherche

Exemples:
payer en ligne
ajouter un commentaire
sur un forum

Exemple : .post()



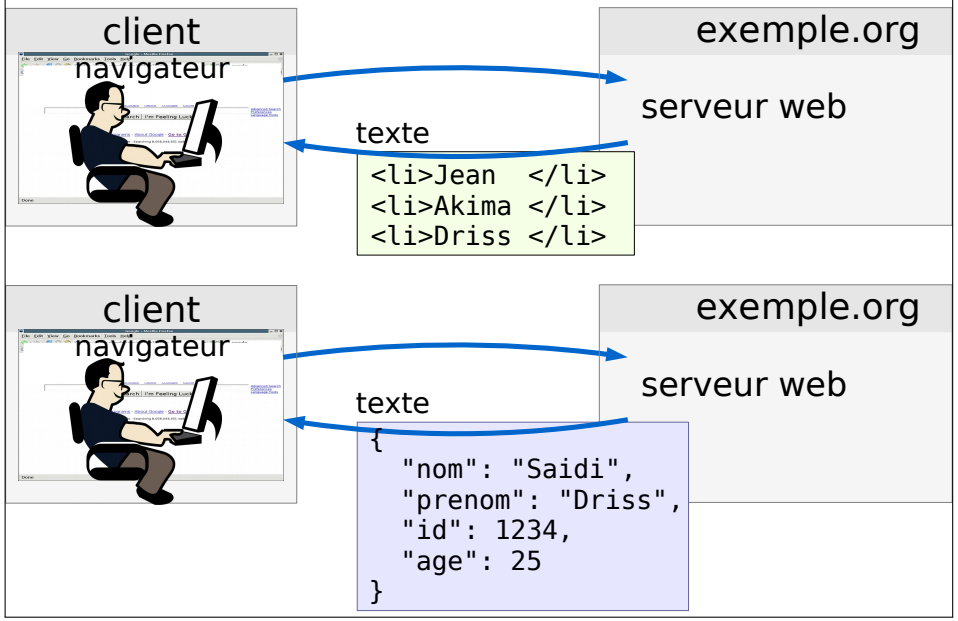
JS

```
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 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()

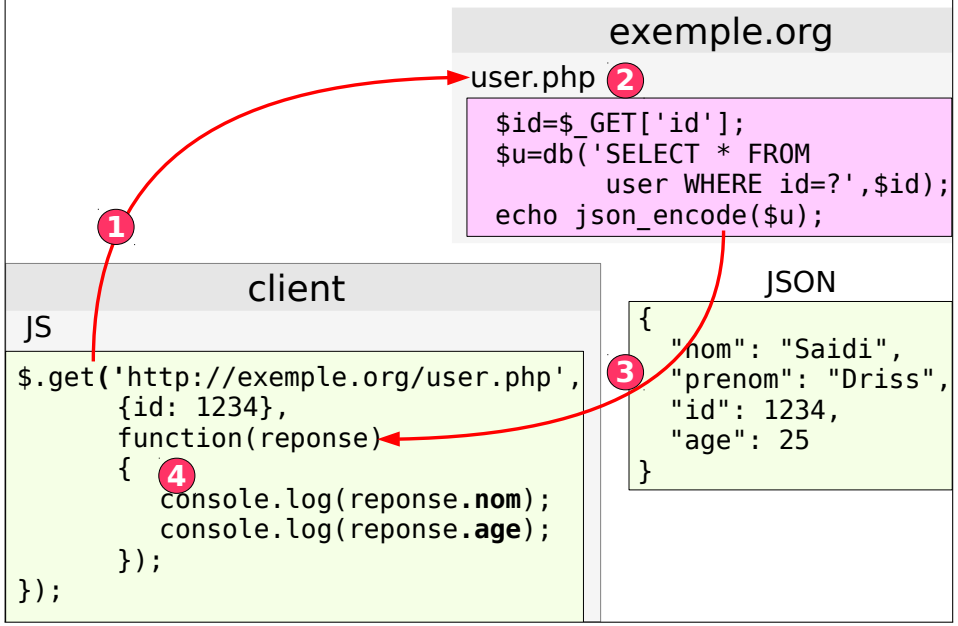
PHP

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

JSON

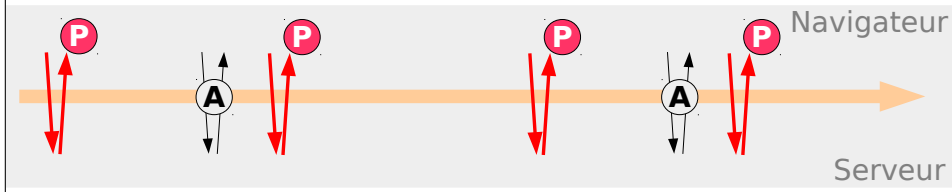
```
{
  "nom": "Saidi",
  "prenom": "Driss",
  "id": 1234,
  "age": 25
}
```

JS : réponse JSON

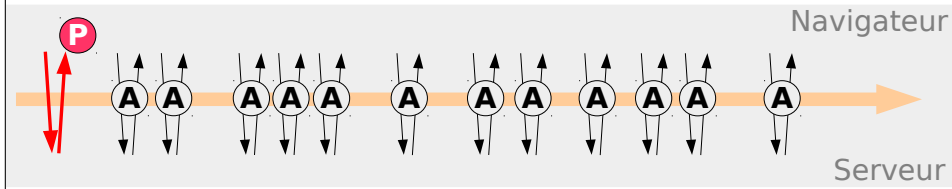


Page + JS vs application

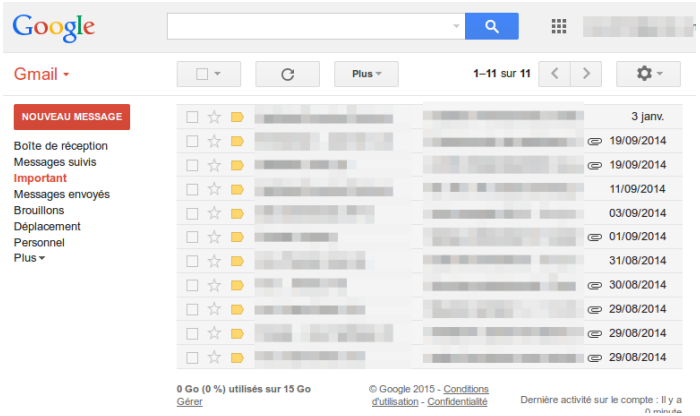
Pages web + JS



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		72%
Mozilla Firefox		9%
Explorer + Edge		9%
Safari		5%
Mobile		54%

Bureau

Statistiques: 11/2018 http://en.wikipedia.org/wiki/Usage_share_of_web_browsers

Compatibilité

caniuse.com

querySelector/querySelectorAll - REC — Global

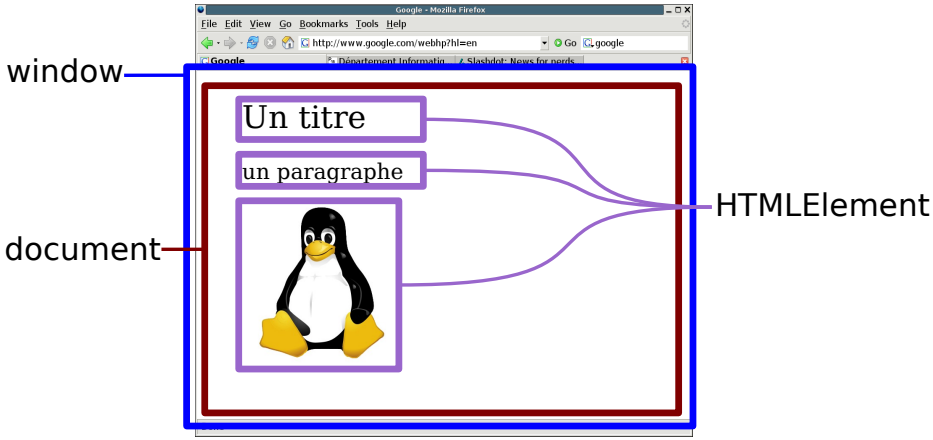
IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *
		31				
		35				
		36				
8		37				
9	34	38				
10	35	39	7.1	26	7.1	
11	36	40	8	27	8.1	8

MDN : developer.mozilla.org

Ordinateur | Mobile

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

Principaux objets DOM



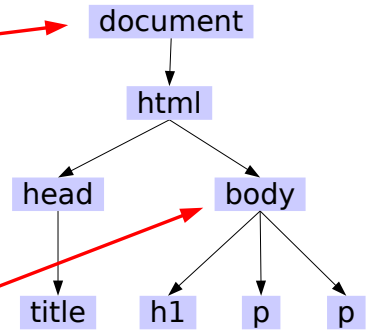
Window & Document

window

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

document

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



document.getElementById()

DOM

d = un élément DOM

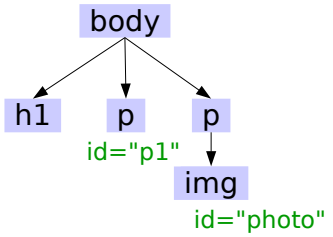
```
var d=document.getElementById('photo');
```

jQuery

j=« liste » d'un seul élément jQuery

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

```
<body>
   <h1>Ceci est un titre</h1>
   <p id="p1">Un paragraphe</p>
   <p>2e paragraphe
      <img id="photo" src="ex.png"...
   </p>
</body>
```



document.createElement()

DOM

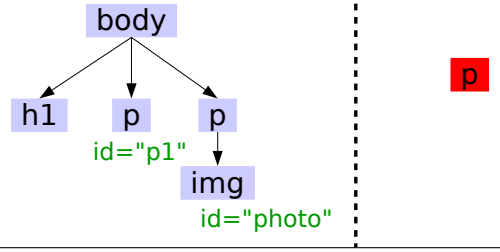
```
var d=document.createElement('p');
```

pas encore dans arbre DOM !

jQuery

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

pas encore dans arbre DOM !



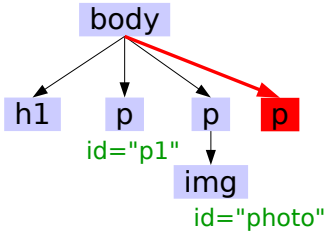
node.appendChild()

DOM

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

jQuery

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



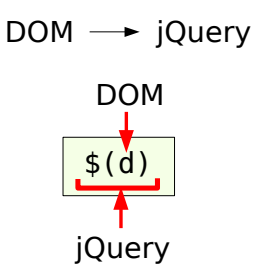
DOM & jQuery

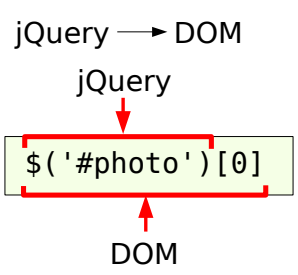
j=« liste » d'un seul élément jQuery

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

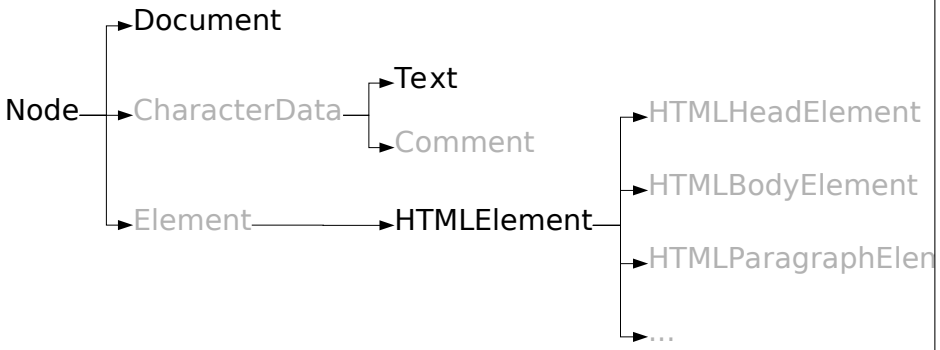
d = un élément DOM

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





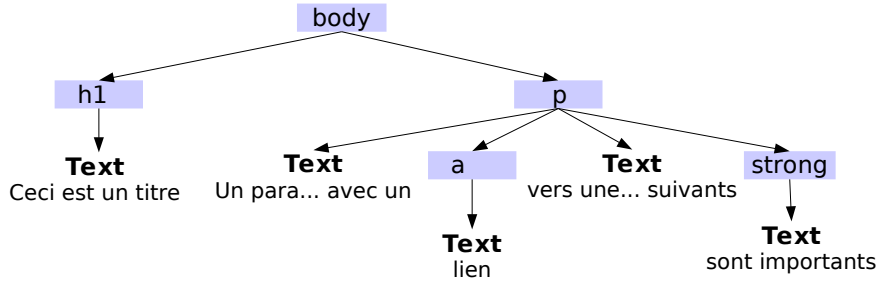
Node / HTMLElement



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

DOM

```
d.className  → "intro important"
d.className="conclusion secondaire"
```

jQuery

```
j.attr('class')
j.hasClass("intro")
j.addClass("xyz")
j.removeClass("intro")
```

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

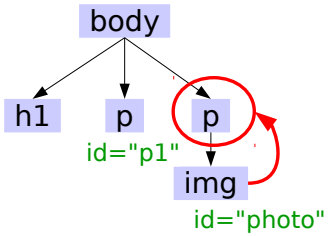
node.parentNode

DOM

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

jQuery

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



node.children

DOM

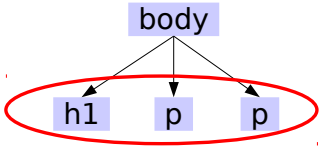
```
var liste=document.body.children;
for(var i=0;i<liste.length;i++)
{
   // ... liste[i] ...
}
```

jQuery

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

children / childNodes

Text



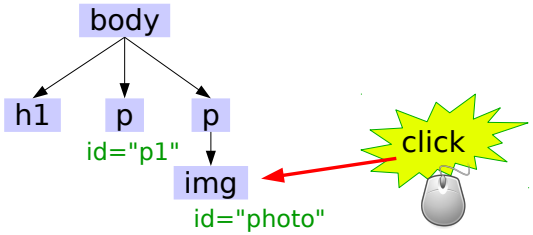
element.addEventListener()

DOM

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

jQuery

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



Marcel.Bosc@iutv.univ-paris13.fr