

JavaScript

M4103C - Programmation Web – client riche

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

Marcel.Bosc@iutv.univ-paris13.fr

Département informatique

IUT de Villetaneuse

Université Paris-13

Table des matières

- Objets et tableaux
- window
- Liste jQuery / arbre
- Fermetures
- Événements
- Exemple : popup

1ère partie

Objets et tableaux

Pas de classe

Java

```
class Etudiant
{
    public String prenom;
}
...
Etudiant leila=new Etudiant();
leila.prenom="Leïla";
...
```

JavaScript

nouvel objet vide
↓
var leila={};
leila.prenom="Leïla";

propriété ajoutée

Créer un objet

```
var leila=
{
  prenom : "Leïla",
  age : 22,
  specialite : "JavaScript"
};
```

Très utilisé !

JSON

Initialisateur d'objet:

```
{
  propriete1 : valeur1,
  propriete2 : valeur2,
  ...
}
```

créer un objet
en spécifiant
des propriétés
et leurs valeurs

Propriétés dynamiques

Java

```
class Etudiant
{
  public String prenom;
}
...
Etudiant leila=new Etudiant();
leila.prenom="Leïla";
leila.age=22;
...
```

JavaScript

```
var leila=
{
  prenom: "Leïla"
};
leila.age=22; 😊
```

objet = tableau associatif

leila.age=22; <=> leila['age']=22;

Objets et fonctions

```
var leila=
{
  prenom: "Leïla"
};
leila.age=22;
leila.afficherNotes=function()
{
  ...
};
leila.afficherNotes();
```

Rappel :
fonction = valeur comme les autres

Objets et fonctions

```
var leila=
{
  prenom: "Leïla",
  age: 22,
  afficherNotes: function()
  {
    ...
  }
};
leila.afficherNotes();
```

Objets et fonctions

```
var leila=  
  {  
    prenom: "Leïla",  
    age: 22,  
    afficherNotes: xyz  
  };  
  
function xyz()  
{  
  ...  
}
```

Tableaux

```
var joueurs=['Leïla','Tom','Aziz'];
```

0	Leïla
1	Tom
2	Aziz

```
joueurs[0] → 'Leïla'  
joueurs[2] → 'Aziz'
```

```
joueurs.push('Lian');
```

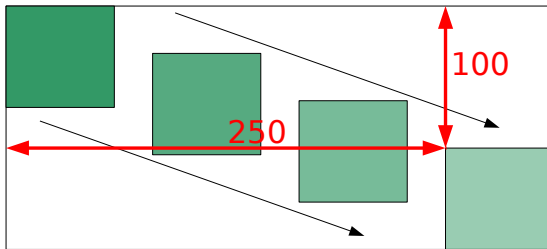
0	Leïla
1	Tom
2	Aziz
3	Lian

```
joueurs.indexOf('Aziz') → 2
```

```
joueurs.indexOf('xyz') → -1
```

```
for(var i=0;i<joueurs.length;i++){ ... }
```

Exemple : .animate()



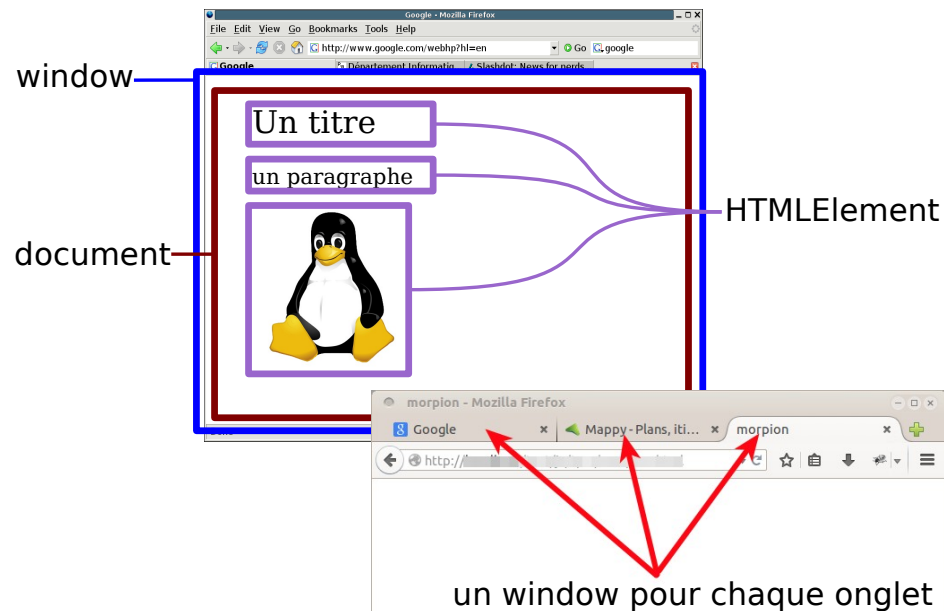
```
var options=  
  {  
    top: 100,  
    left: 250,  
    opacity: .5  
  };  
  
$('#boite').animate(options,1000);
```

millisecondes

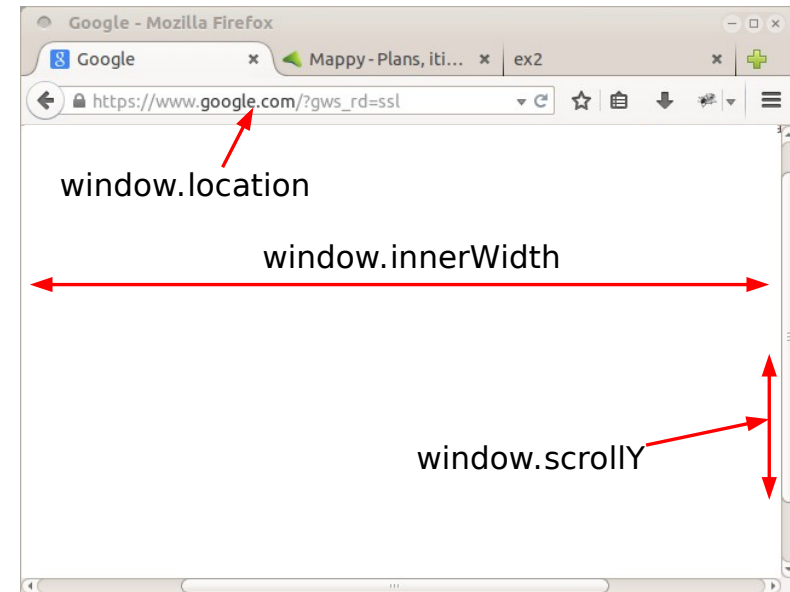
2e partie

window

Rappel : objets DOM



window



window : objet global

```
a="bonjour";
```



```
window.a="bonjour";
```

window = objet global

Vos variables globales sont en réalité des propriétés de window.

Fonctions window / JS

Méthodes window :

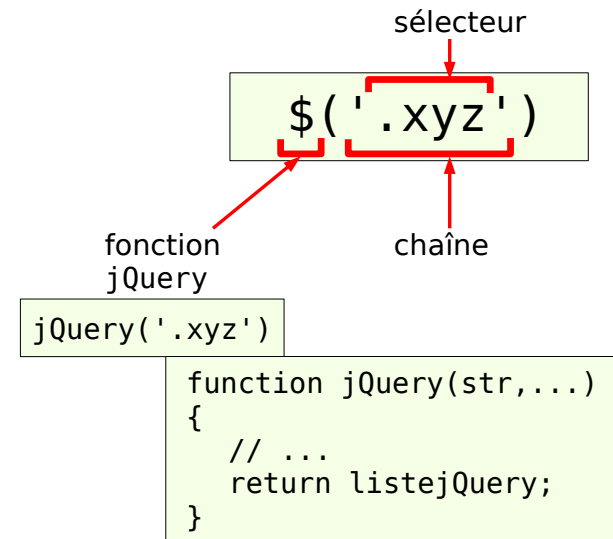
- `window.alert()` -> `alert()`
- `window.console` -> `console`
- `window.setInterval()` -> `setInterval()`
- `window.setTimeout()` -> `setTimeout()`
- ...

Fonctions JavaScript :

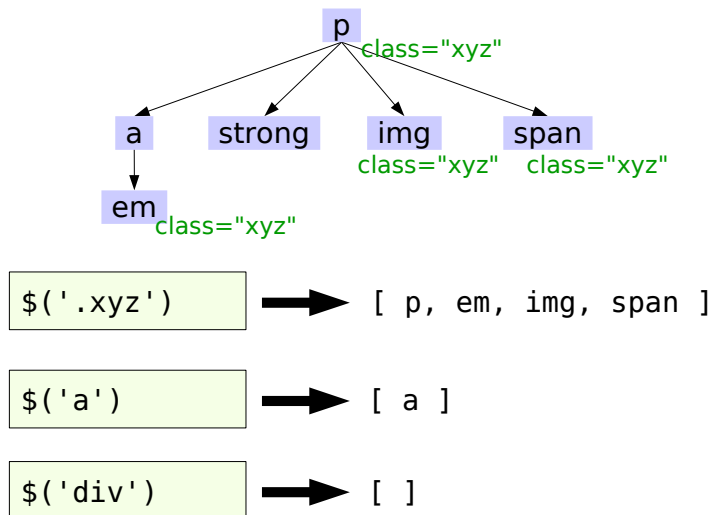
- `parseInt()`
- `eval()`
- ...

Liste jQuery / arbre

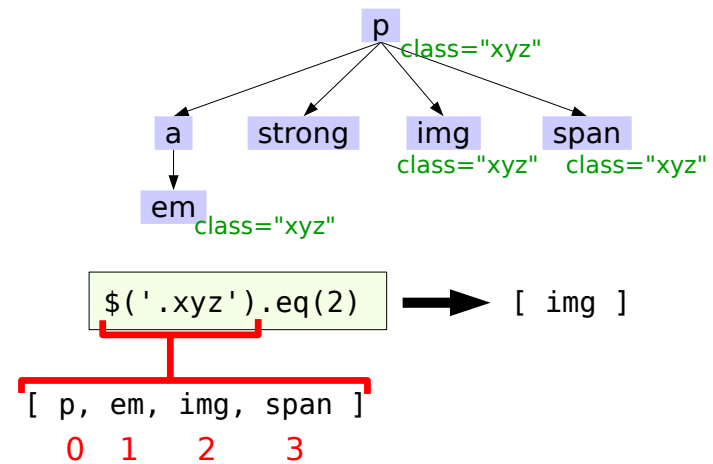
Fonction jQuery



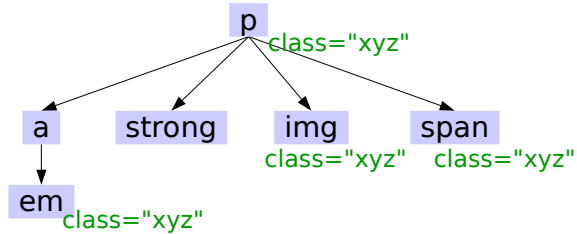
jQuery : listes



.eq(n)



.parent(), .children()

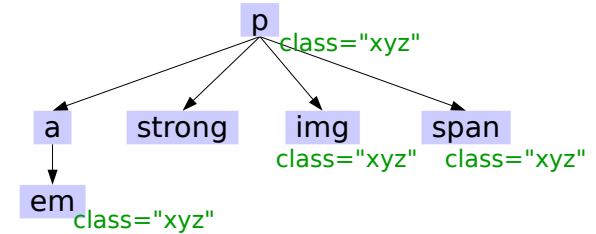


`$('img').parent()` ➔ [p]

`$('p').children()` ➔ [a, strong, img, span]

`$('p').children('.xyz')` ➔ [img, span]

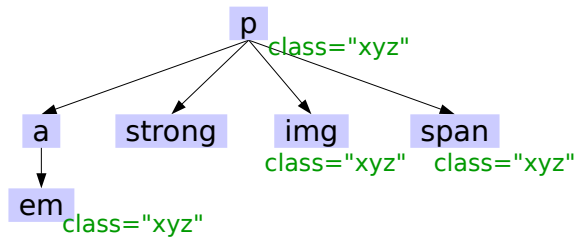
.find()



`$('p').children('.xyz')` ➔ [img, span]

`$('p').find('.xyz')` ➔ [em, img, span]

Enchaîner



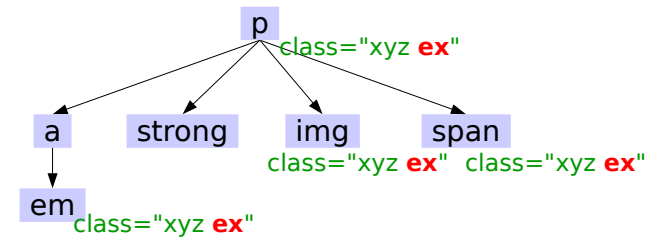
`$('p').children('.xyz')` ➔ [img, span]

`$('p').children('.xyz').eq(0)` ➔ [img]

`$('a').parent()` ➔ [p]

`$('a').parent().find('img')` ➔ [img]

Effet sur toute la liste

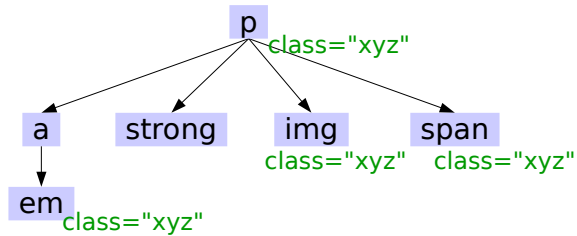


`$('.xyz').addClass('ex')`

[p, em, img, span]

Les fonctions jQuery s'appliquent souvent à toute la liste

.each()



```

$('.xyz').each(function()
{
    $(this).css('left',100*Math.random());
});
  
```

.each() :
parcourir tous les éléments de la liste ... « boucle »

.attr()

```
<a id="lien5" class="xyz ex" href="page.html">...</a>
```

```

$('a').attr('id')    => 'lien5'
$('a').attr('class') => 'xyz ex'
$('a').attr('href') => 'page.html'
  
```

```
$('a').attr('href','scores.html'),
```

```
<a id="lien5" class="xyz ex" href="scores.html">...</a>
```

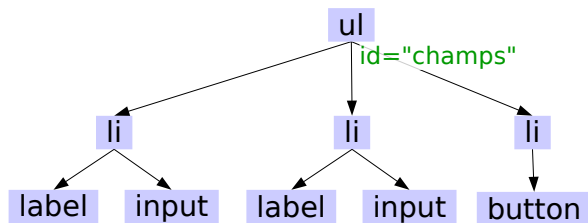
Exemple : .parent()

```

<ul id="champs">
  <li><label>Nom : </label><input type="text"/></li>
  <li><label>Prénom :</label><input type="text"/></li>
  <li><button>Envoyer</button> </li>
</ul>
  
```

Nom :

Prénom :



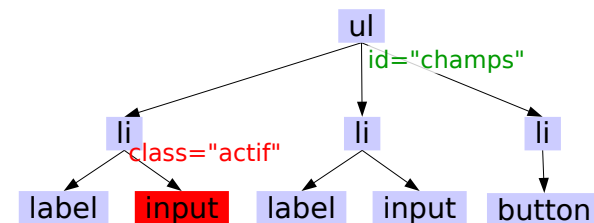
Exemple : .parent()

```

#champs li.actif
{
  background-color: #ff0;
}
  
```

Nom :

Prénom :



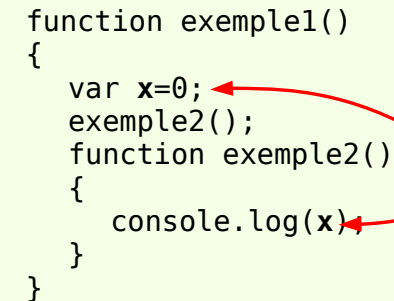
```

$('input').focus(function()
{
    $(this).parent().addClass('actif');
});
  
```

Fermetures

Fermetures

```
function exemple1()
{
  var x=0;
  exemple2();
  function exemple2()
  {
    console.log(x);
  }
}
```

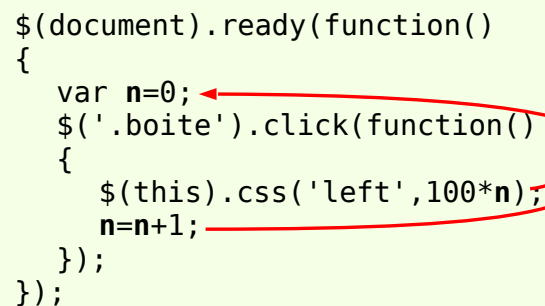


Fermeture :

Une fonction utilisant des variables définies dans une autre fonction contenante.

Fermetures

```
$(document).ready(function()
{
  var n=0;
  $('.boite').click(function()
  {
    $(this).css('left',100*n);
    n=n+1;
  });
});
```

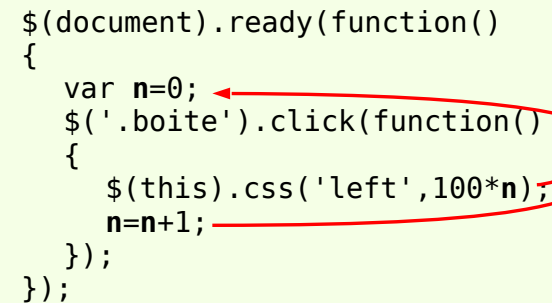


Programmation événementielle



Fermetures : facile ?

```
$(document).ready(function()
{
  var n=0;
  $('.boite').click(function()
  {
    $(this).css('left',100*n);
    n=n+1;
  });
});
```



Très utilisé, très intuitif !



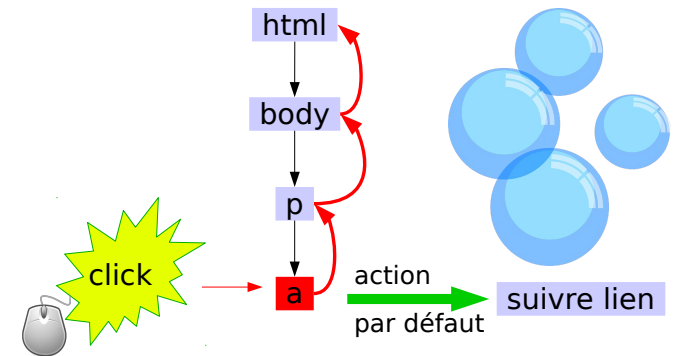
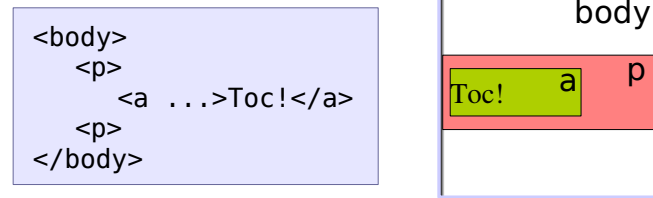
Quoi !? 😬

Accéder aux variables locales d'une autre fonction ?!

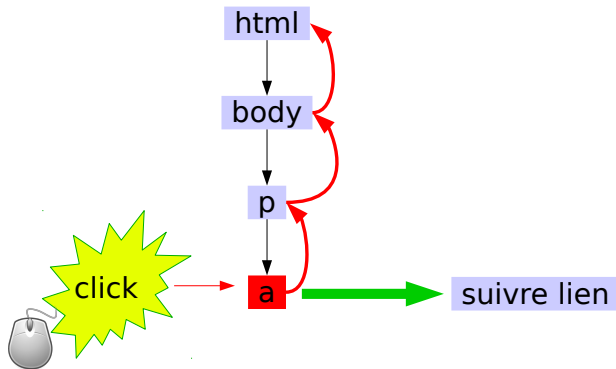
L'autre fonction a déjà fini d'exécuter ?!

Événements

Événements : bubbling

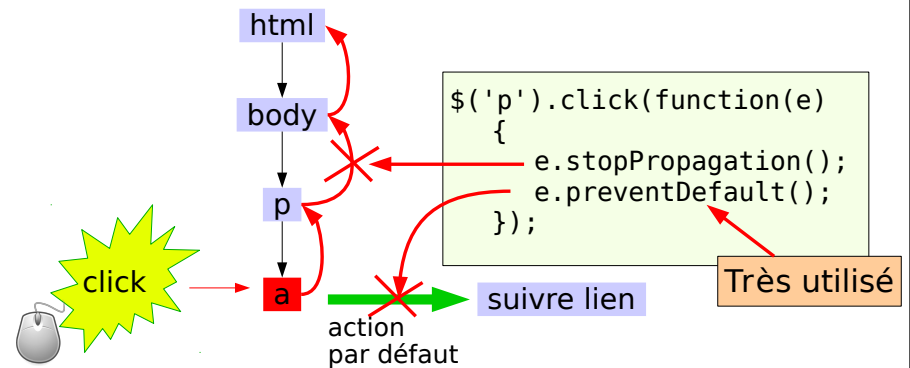


Événements : bubbling



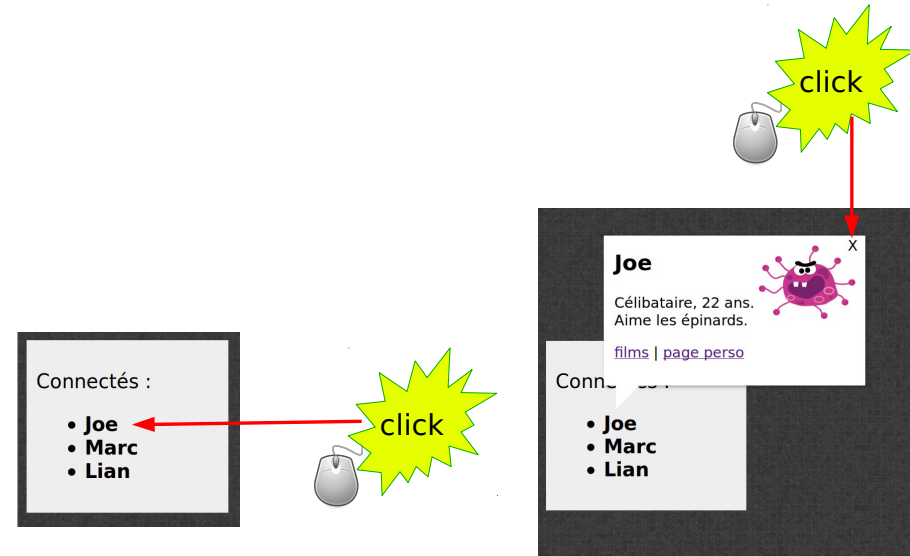
```
$('html').click(function(e){/* this:html e.target:a */});
$('body').click(function(e){/* this:body e.target:a */});
$('p').click(function(e){/* this:p e.target:a */});
$('a').click(function(e){/* this:a e.target:a */});
```

Événements



Exemple : popup

Exemple : popup



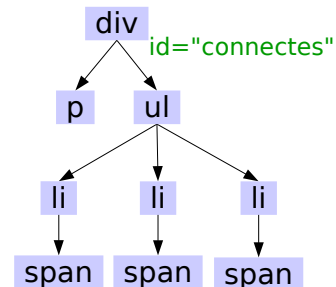
Exemple : HTML

```
<div id="connectes">
  <p>Connectés : </p>
  <ul>
    <li><span class="popup-mot" data-popup="joe" >Joe </sp>
    <li><span class="popup-mot" data-popup="marc">Marc</sp>
    <li><span class="popup-mot" data-popup="lian">Lian</sp>
  </ul>
</div>

<div id="popup-joe" class="popup">
  <span class="fermer">X</span>
  
  
  <h2>Joe</h2>
  <p>Célibataire, 22 ans.<br/>Aime les épinards.</p>
  <p>
    <a href="...">films</a> | <a href="...">page perso</a>
  </p>
</div>
```

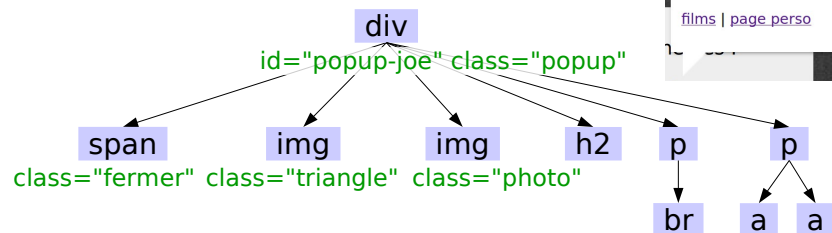
Exemple : liste

```
<div id="connectes">
  <p>Connectés : </p>
  <ul>
    <li><span class="popup-mot" data-popup="joe" >Joe </sp>
    <li><span class="popup-mot" data-popup="marc">Marc</sp>
    <li><span class="popup-mot" data-popup="lian">Lian</sp>
  </ul>
</div>
```



Exemple : popup

```
<div id="popup-joe" class="popup">
  <span class="fermer">X</span>
  
  
  <h2>Joe</h2>
  <p>Célibataire, 22 ans.<br/>Aime les épinards.</p>
  <p>
    <a href="...">films</a> | <a href="...">page perso</a>
  </p>
</div>
```



Exemple : CSS

containing block

```
.popup {
  position: absolute;
  width: 200px;
}
```

```
.popup>.fermer {
  position: absolute;
  top: 0;
  right: 0;
}
```

```
.popup>.triangle {
  position: absolute;
  left: 10px;
  bottom: -20px;
}
```

Exemple : .offset()

```
.popup {
  position: absolute;
  ...
}
```

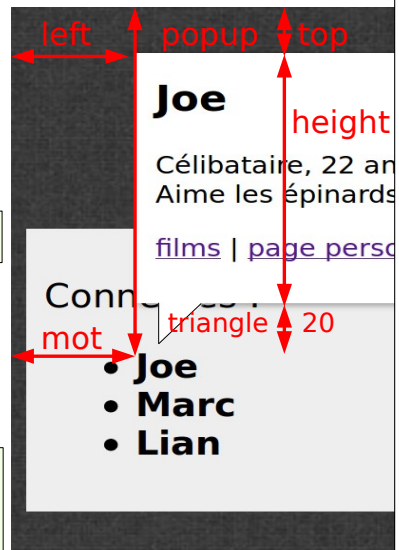
mot cliqué

```
var pos=$(this).offset()
```

objet JS

```
{
  top: 200px,
  left: 80px
}
```

```
pos.top-=$( '#popup-joe' ).height();
pos.top-=20;
$( '#popup-joe' ).offset(pos);
```



Exemple : .offset()

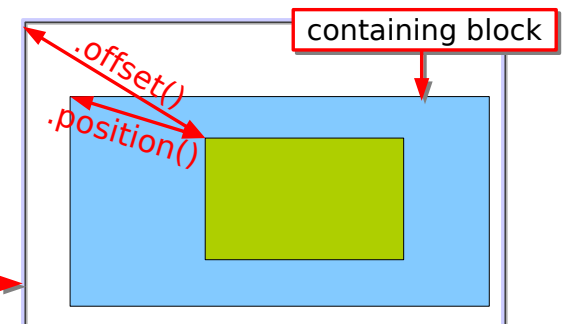
```
$('#popup-joe').offset({left: 100, top: 50});
```

```
<div id="popup-joe" >...</div>
```

```
<div id="popup-joe" style="left: 100; top: 50">...</div>
```

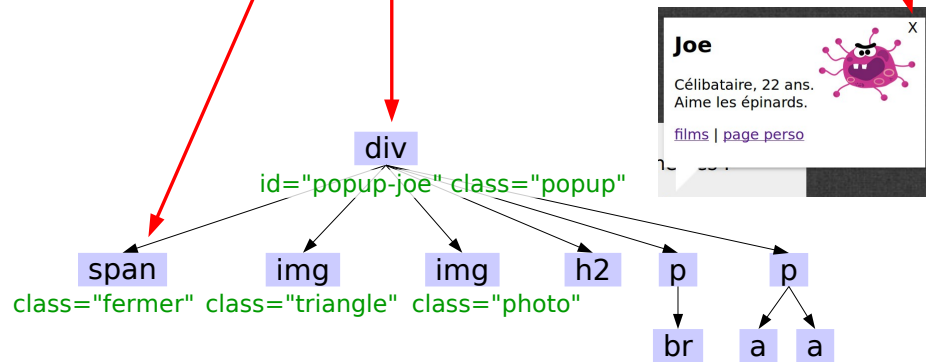
.offset()
.position()

page



Exemple : fermer

```
$( '.popup>.fermer' ).mousedown(function()
{
    $(this).parent().fadeOut();
});
```



Exemple : JS

```
0  $( '.popup-mot' ).mousedown(function()
1  {
2      var mot=$(this);
3      var id="popup-"+mot.attr('data-popup');
4      var popup=$( '#' + id );
5      var pos=mot.offset();
6      pos.top-=popup.outerHeight()+20;
7      popup.fadeIn();
8      popup.offset(pos);
9  });
10
11 $( '.popup>.fermer' ).mousedown(function()
12 {
13     $(this).parent().fadeOut();
14 });
```

```
<li><span class="popup-mot" data-popup="joe" >Joe </span></li>
<div id="popup-joe" class="popup">
```

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>