

Internet

jQuery jQuery Mobile Partie 2

Olivier Pons / 2013



Objectif

jQuery / jQuery Mobile

Sommaire

1. Fondamentaux DOM
2. jQuery
3. jQuery Mobile



1. Fondamentaux DOM - Définition

Le Document Object Model (ou DOM) est un standard du W3C qui décrit une interface indépendante de tout langage de programmation et de toute plate-forme, permettant à des programmes informatiques et à des scripts d'accéder ou de mettre à jour le contenu, la structure ou le style de documents XML et HTML.

http://fr.wikipedia.org/wiki/Document_Object_Model

1. Fondamentaux DOM - Définition

Plusieurs niveaux de DOM : DOM 1, DOM 2...

DOM 3

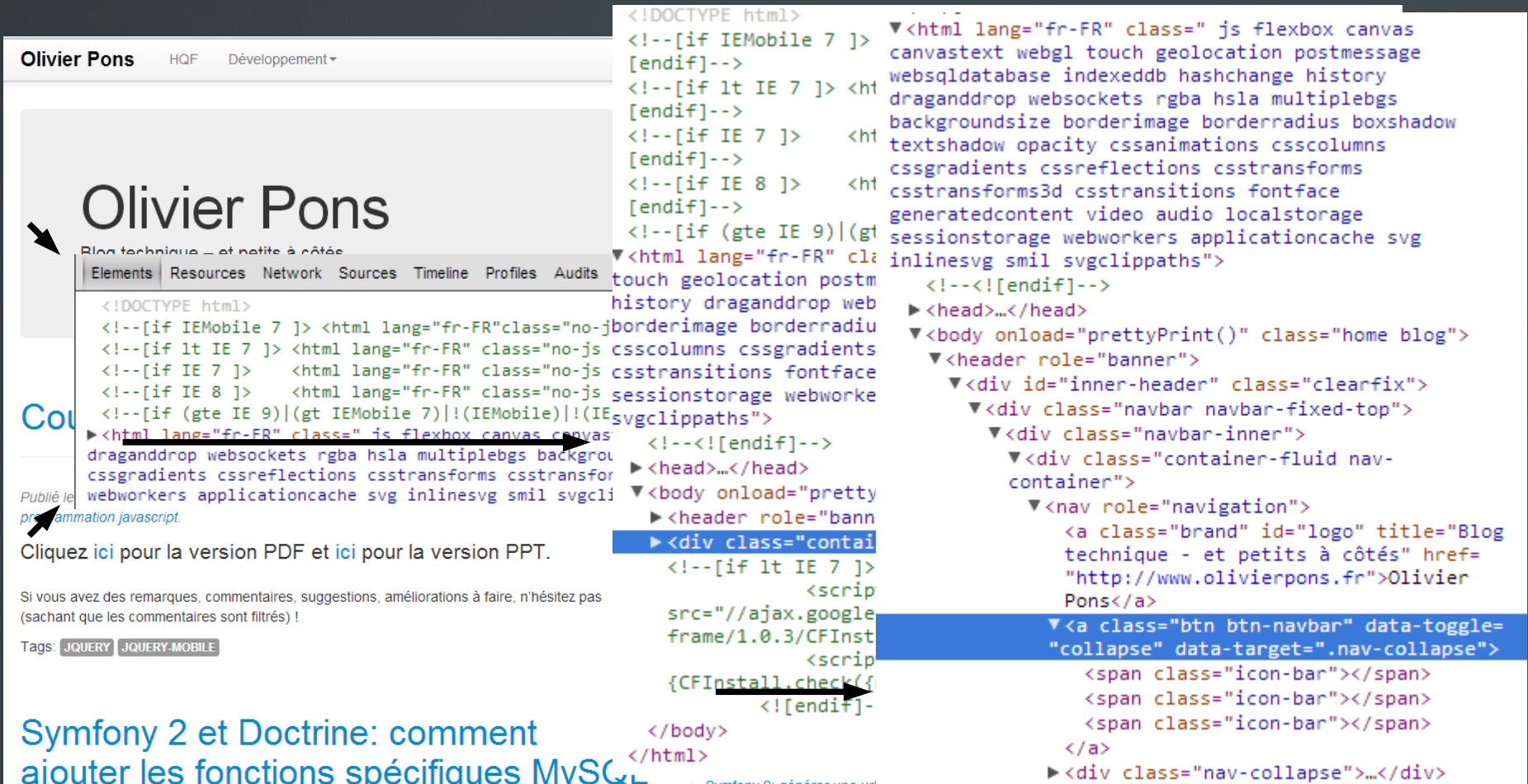
Le troisième niveau a été publié au printemps 2004.

Le groupe officiel de travail a aujourd'hui arrêté le développement du DOM.



Internet : jQuery / jQuery Mobile / Part. 2

1. Fondamentaux DOM - Chrome



Olivier Pons HQF Développement

Olivier Pons

Blog technique - et petits à côtés

Elements Resources Network Sources Timeline Profiles Audits

```
<!DOCTYPE html>
<!--[if IEMobile 7 ]>
[endif]-->
<!--[if lt IE 7 ]>
[endif]-->
<!--[if IE 7 ]>
[endif]-->
<!--[if IE 8 ]>
[endif]-->
<!--[if (gte IE 9)|(gt IEMobile 7)]!(IEMobile)!(IE
<html lang="fr-FR" class=" js flexbox canvas
draganddrop websockets rgba hsla multiplebgs backgrou
cssgradients cssreflections csstransforms csstransfor
webworkers applicationcache svg inlinesvg smil svgcli
<!--<![endif]-->
<head>...</head>
<body onload="prettyPrint()" class="home blog">
  <header role="banner">
    <div id="inner-header" class="clearfix">
      <div class="navbar navbar-fixed-top">
        <div class="navbar-inner">
          <div class="container-fluid nav-
container">
            <nav role="navigation">
              <a class="brand" id="logo" title="Blog
technique - et petits à côtés" href=
"http://www.olivierpons.fr">Olivier
Pons</a>
              <a class="btn btn-navbar" data-toggle=
"collapse" data-target=".nav-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
              </a>
            <div class="nav-collapse">...</div>
```

Publié le 10/10/2012 à 10:10

Présentation javascript.

Cliquez [ici](#) pour la version PDF et [ici](#) pour la version PPT.

Si vous avez des remarques, commentaires, suggestions, améliorations à faire, n'hésitez pas (sachant que les commentaires sont filtrés) !

Tags: [JQUERY](#) [JQUERY-MOBILE](#)

Symfony 2 et Doctrine: comment ajouter les fonctions spécifiques MySQL

2. jQuery – Histoire

22 août 2005 : John publie pour la première fois les prémices d'une librairie JavaScript qui utilise des sélecteurs CSS avec une syntaxe plus succincte que les librairies existantes : Selectors



2. jQuery – Histoire

Aujourd'hui : 3 parties

jQuery = coeur

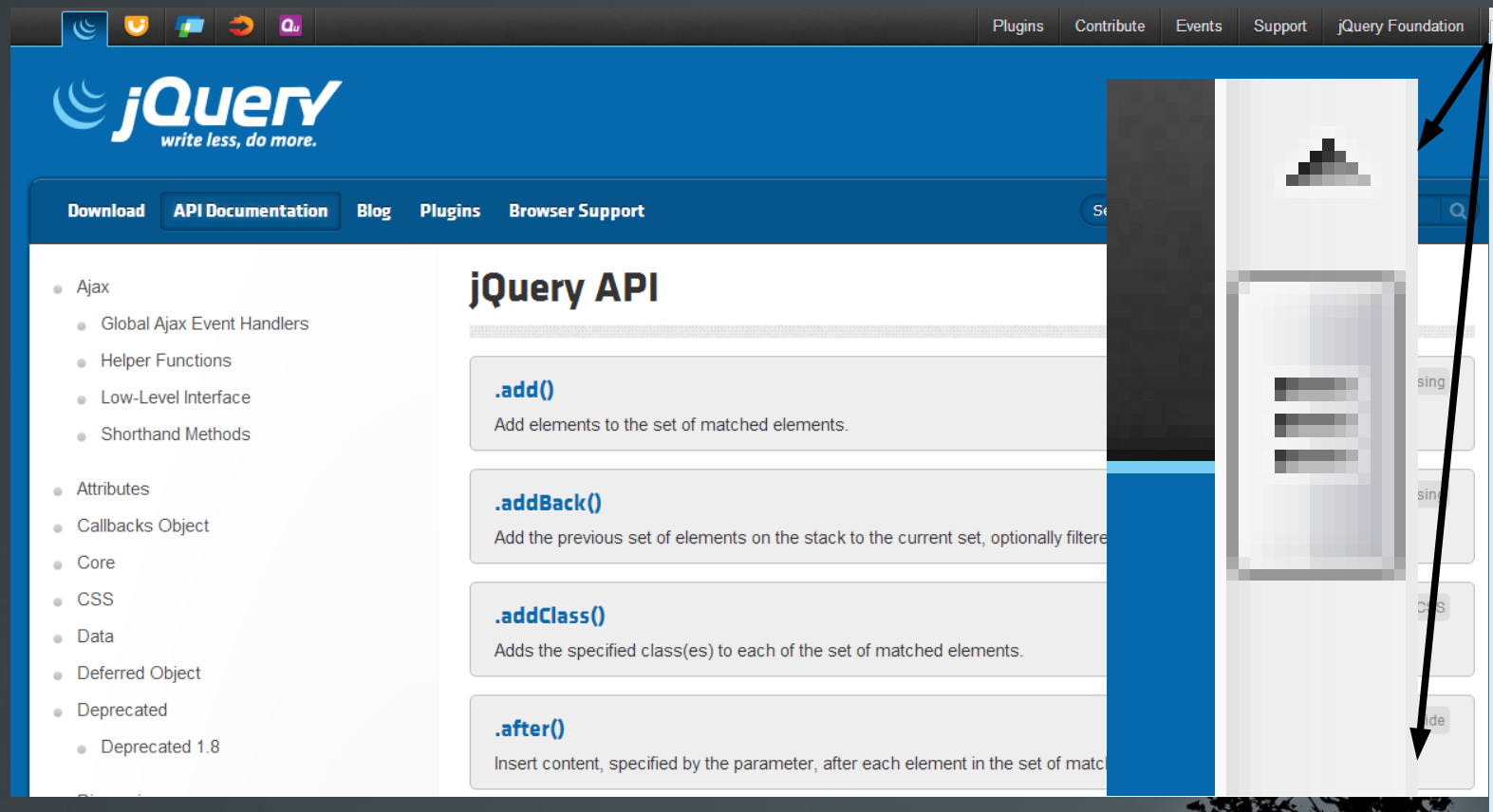
jQueryUI = interface graphique

jQuery Mobile = smartphones
et tablettes



Internet : jQuery / jQuery Mobile / Part. 2

2. jQuery - API



The image shows a screenshot of the jQuery website's API documentation page. The page features a blue header with the jQuery logo and tagline "write less, do more.". Below the header is a navigation bar with links for "Download", "API Documentation", "Blog", "Plugins", and "Browser Support". The main content area is titled "jQuery API" and lists several methods: ".add()", ".addBack()", ".addClass()", and ".after()". A vertical navigation bar is overlaid on the right side of the page, containing a search bar and a list of categories: "sing", "sing", "CSS", and "ide". A black arrow points from the top of the navigation bar to the ".add()" method in the main content area.

jQuery
write less, do more.

Download API Documentation Blog Plugins Browser Support

jQuery API

- .add()**
Add elements to the set of matched elements.
- .addBack()**
Add the previous set of elements on the stack to the current set, optionally filtered.
- .addClass()**
Adds the specified class(es) to each of the set of matched elements.
- .after()**
Insert content, specified by the parameter, after each element in the set of matched elements.

Navigation bar: sing, sing, CSS, ide

2. jQuery – API

- 1) AJAX
- 2) DOM
 - Sélecteurs
 - Effets
 - Manipulation
- 3) Gestion des événements



2. jQuery – API

Si c'est du jQuery c'est

Soit :

```
.fonction()
```

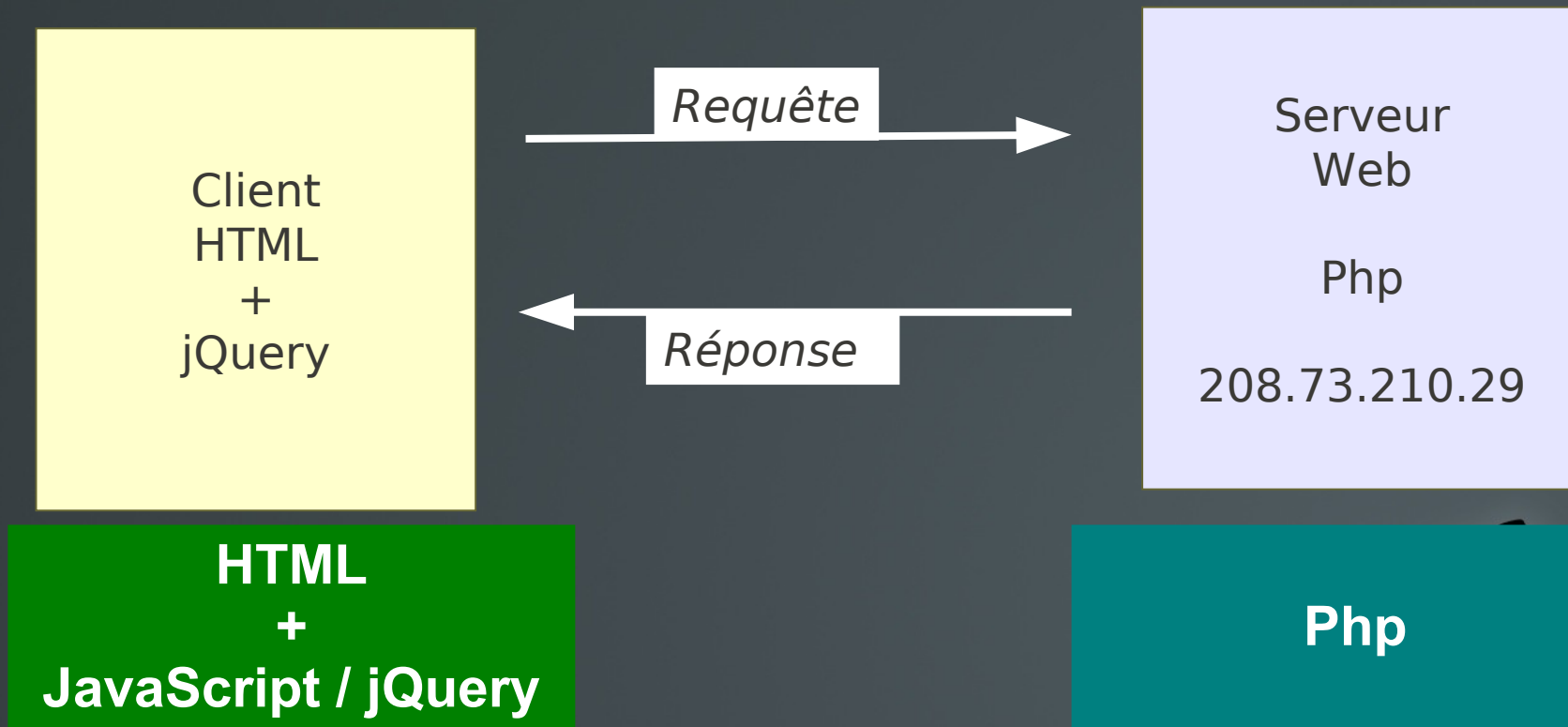
Soit :

```
$(objet)
```

```
$('element')
```

2. jQuery – AJAX – Rappel

<http://monsiteweb.fr/post.php>



2. jQuery – AJAX – Exemple - déjà vu

```
$(document).ready(OnReady);
function OnReady(){
    $("form").submit(OnSubmit);
}
function OnSubmit(data){
    $.ajax({
        type: $(this).attr("method"),
        url: $(this).attr("action"),
        data: $(this).serialize(),
        success: OnSuccess
    });
    return false;
}
function OnSuccess(result){
    $("#result").html(result);
}
```

2. jQuery – AJAX – Exemple - déjà vu

```
...
<body>
  <form method="post" action="add.php">
    <fieldset>
      <legend>Choisissez deux nombres entiers</legend>
      <p><label>
        a = <input name="a" type="number" required>
      </label></p>
      <p><label>
        b = <input name="b" type="number" required>
      </label></p>
    </fieldset>
    <fieldset>
      <legend>Résultat</legend>
      <p id="result"></p>
    </fieldset>
    <p><button>Soumettre</button></p>
  </form>
</body>
</html>
```

2. jQuery – AJAX – Exemple - déjà vu

```
<?php  
  
/* Envoi au client le résultat du calcul de a + b */  
print(  
    intval($_POST["a"]) +  
    intval($_POST["b"])  
);  
  
?>
```

2. jQuery – AJAX – Exemple 2

```
$.ajax( "example.php" )  
  .done(function() {  
    console.log("success");  
  })  
  .fail(function() {  
    console.log("error");  
  })  
  .always(function() {  
    console.log("complete");  
  });
```

2. jQuery – DOM – Sélecteurs

Sélectionner...

...tous les éléments :

```
$( '* ' )
```

```
$( '* ' ).css( 'border', '3px solid red' )
```

...les éléments d'une classe :

```
$( '.maclasse' )
```

```
$( '.maclasse' ).css( 'border', '3px solid red' )
```

...l'élément par son id :

```
$( '#elementid' )
```

```
$( '#idimg' ).css( 'border', '3px solid red' )
```

2. jQuery – DOM – Sélecteurs

Sélectionner...

...en filtrant sur un attribut qui commence par valeur :

```
$( 'a[href|"valeur"] ' )  
    $( 'a[href|"valeur"] ' ) .css ( . . . ) ;
```

...en filtrant sur un attribut qui contient une valeur :

```
.$ ( 'a[name*"en"] ' )  
    $ ( 'a[name*=en] ' ) .css ( . . . )
```

...et ainsi de suite avec commence par, ne contient pas, etc.

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

2. jQuery – DOM – Effets

Que du visuel

Redimensionnement :

`.animate()`

Opacité :

`.fadeIn()` / `.fadeOut()`

Déplacement :

`.slideUp()` / `.slideDown()`



2. jQuery – DOM – Effets

HTML :

```
<div id="clickme">  
  Click here  
</div>  

```

JavaScript / jQuery :

```
$('#clickme').click(function() {  
  $('#livre').slideDown('slow', function() {  
    // Animation terminée.  
  });  
});
```

2. jQuery – DOM – Effets

HTML :

```
<div id="patientez">
  Patientez quelques instants...
</div>
<div id="message">
  Téléchargement terminé !
</div>
```

JavaScript / jQuery :

```
$( '#patientez' ).fadeOut( 'slow', function() {
  $( '#message' ).fadeIn( 'slow' );
});
```

2. jQuery – DOM – Manipulation

Ce sont toutes les méthodes qui changent un des attributs d'un élément, ou les propriétés de style. Les plus courants :

```
.css ()  
.html ()  
.empty ()  
.attr ()
```

2. jQuery – DOM – Manipulation

Exemples :

```
$( "p" ).css ( "color" , "red" ) ;
```

```
$( "div" ).html ( "<b>Super !</b>" ) ;
```

```
$( ' .hello ' ).empty ( ) ;
```

```
$( '#monimage' ).attr (
    'alt' , 'Saut en parachute' ) ;
```

2. jQuery – DOM – Événements

Ces fonctions sont utilisées pour agir en fonction d'événements déclenchés lorsque l'utilisateur interagit avec le navigateur.



2. jQuery – DOM – Événements

En général, deux possibilités :

Déclencher à la main l'événement :

```
.evt()
```

```
$( '#target' ).blur();
```

Définir une fonction à appeler lorsque l'événement est déclenché :

```
.evt(function() {  
    code JavaScript  
});
```

2. jQuery – DOM – Événements

Les plus courants :

- `.bind()`
- `.blur()`
- `.change()`
- `.click()`
- `.hover()`
- `.keydown()`
- `.keypress()`
- `.keyup()`
- `.off()`
- `.on()`
- `.one()`



2. jQuery – DOM – Événements

Exemple `.one()` :

```
$("#foo").one("click", function() {  
    alert("Affiché une seule fois");  
});
```

```
$("body").one("click", "#foo", function() {  
    alert("Affiché que si #foo est le premier  
élément cliqué dans body.");  
});
```

2. jQuery – DOM – Événements

Exemple .off() / .on() :

```
$("#bind").click(function () {
    $("body").on("click", "#monid", aClick)
        .find("#theone").text("Actif");
});
$("#unbind").click(function () {
    $("body").off("click", "#monid", aClick)
        .find("#theone").text("Inactif...");
});
function aClick() {
    $("div").show().fadeOut("slow");
}
```

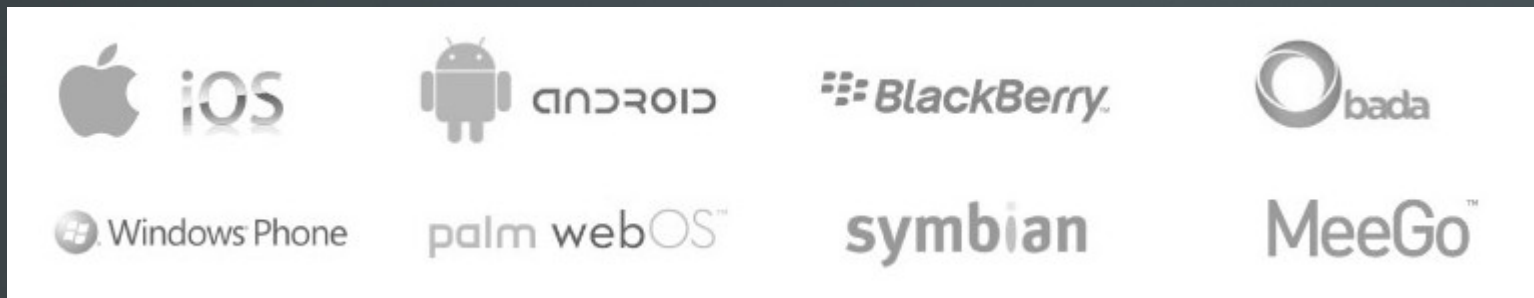
2. jQuery – DOM – Événements

Exemple empty() append()click() :

```
$( '#panier' )  
  .empty()  
  .append(  
    $( '<div>' ).html( 'Cliquez ici...' )  
  )  
  .unbind( 'click' )  
  .click( function( event ) {  
    event.preventDefault();  
    montrePanier();  
  } );
```

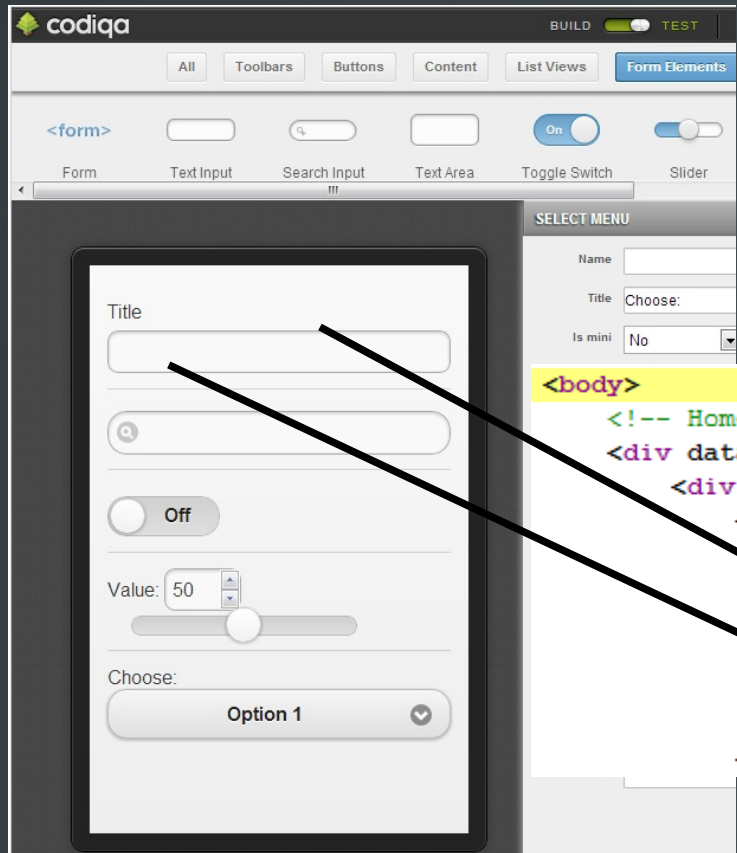
3. jQuery Mobile – API

Mot-clé : cross-platform



3. jQuery Mobile – API

Test direct live



```
<body>
  <!-- Home -->
  <div data-role="page" id="page1">
    <div data-role="content">
      <div data-role="fieldcontain">
        <fieldset data-role="controlgroup">
          <label for="textinput1">
            Title
          </label>
          <input name="" id="textinput1" placeholder="" value="" type="text" />
        </fieldset>
      </div>
    </div>
  </div>
```

3. jQuery Mobile – API

Des tonnes de widgets

Range slider

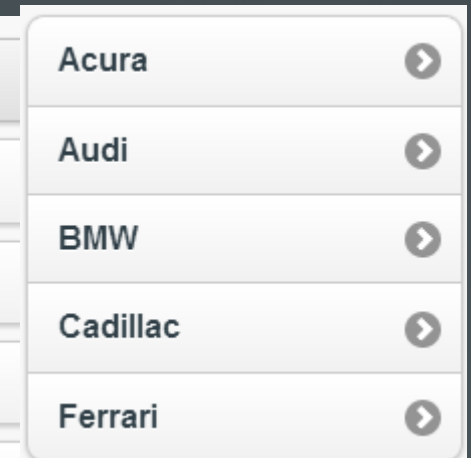
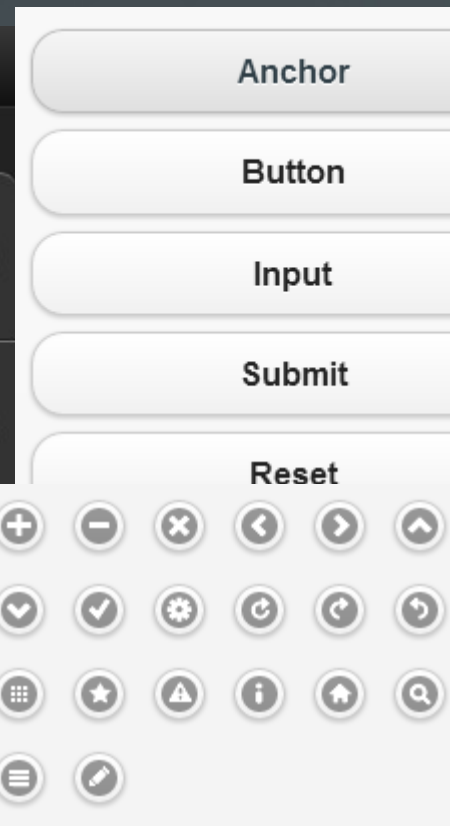
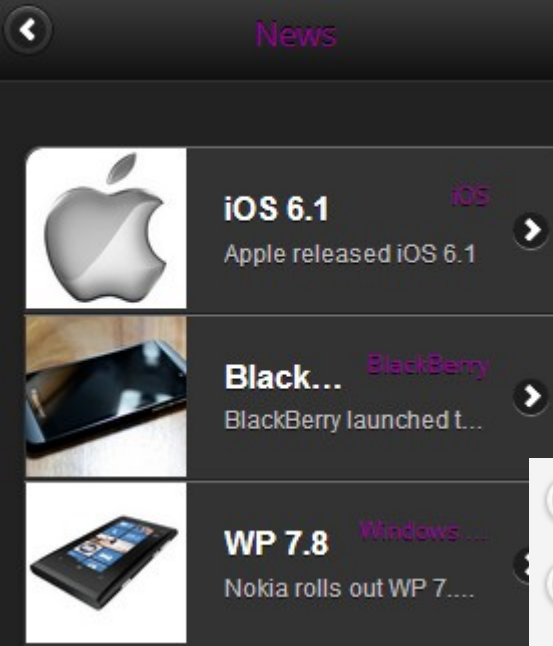
Range slider offer two han
max value along a numeric

Basic range slider

Rangeslider:

40

80



3. jQuery Mobile – API

Ce n'est que de l'habillage !

Dynamic collapsible

This demo shows how you can dynamically add a new section to a collapsible set (accordion). It also shows how to use the expand and collapse events to dynamically open or close a collapsible.

[Add ⚙](#) [Expand last +](#) [Collapse last -](#)

[-] Section 1

I'm the collapsible content.

[+] Section 2

[+] Section 3

[View Source](#)

HTML

```
1 <button type="button" data-icon="gear" data-theme="b" data-iconpos="top" data-corners="true" data-shadow="true" data-textshadow="false" data-focus="false" data-no-close="true" data-no-focus="true" data-no-hover="true" data-no-outline="true" data-no-rotate="true" data-no-scale="true" data-no-shrink="true" data-no-zoom="true" data-role="button"></button>
2 <button type="button" data-icon="plus" data-theme="b" data-iconpos="top" data-corners="true" data-shadow="true" data-textshadow="false" data-focus="false" data-no-close="true" data-no-focus="true" data-no-hover="true" data-no-outline="true" data-no-rotate="true" data-no-scale="true" data-no-shrink="true" data-no-zoom="true" data-role="button"></button>
3 <button type="button" data-icon="minus" data-theme="b" data-iconpos="top" data-corners="true" data-shadow="true" data-textshadow="false" data-focus="false" data-no-close="true" data-no-focus="true" data-no-hover="true" data-no-outline="true" data-no-rotate="true" data-no-scale="true" data-no-shrink="true" data-no-zoom="true" data-role="button"></button>
4 <div data-role="collapsible-set" data-content-theme="d" id="set">
5   <div data-role="collapsible" id="set1" data-collapsed="true">
6     <h3>Section 1</h3>
7     <p>I'm the collapsible content.</p>
8   </div>
9 </div>
```