

Internet

jQuery jQuery Mobile

Olivier Pons / 2019 - 2020



Internet : jQuery / jQuery Mobile

Objectif

Json / Ajax / jQuery / Mobile

Sommaire

1. Fondamentaux + Json / Ajax
2. jQuery
3. jQuery mobile



1. Fondamentaux - Protocole

- HTTP: HyperText Transfer Protocol
- HTTP: les principales méthodes
 - GET URL : demander le contenu de la ressource
 - POST URL : envoi de données vers une application
- HTTP: le transport
 - Architecture Client-Serveur, mode « Pull »
 - Connexions courtes, « Sans état » (stateless)

1. Fondamentaux – Client Serveur

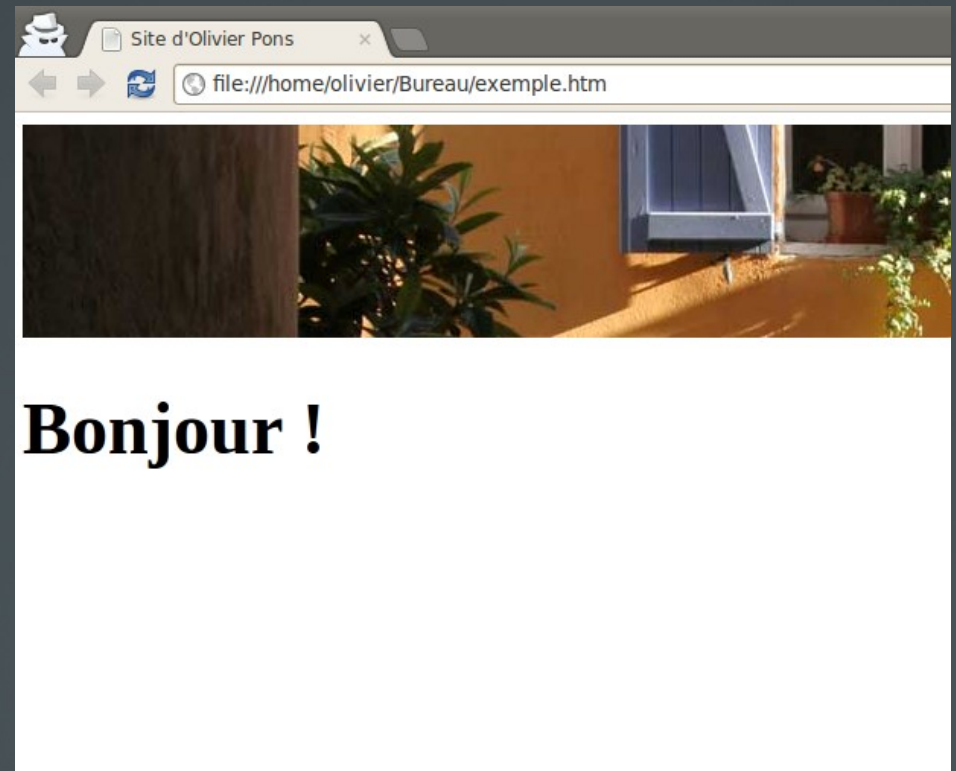


1. Fondamentaux – Client Serveur



1. Fondamentaux – HTML

```
<html>
  <head>
    ...
    <title>Site d'Olivier Pons</title>
    ...
  </head>
<body>
  <div>
    
  </div>
  <h1>Bonjour !</h1>
</div>
</body>
</html>
```



1. Fondamentaux – Echanges

- (1) Client demande une page**
- (2) Serveur renvoie la page**

(Boucle)

**Client demande ressource
nécessaire à la page
Serveur renvoie la ressource**

(Fin boucle)



1. Json / Ajax – Json

JSON (JavaScript Object Notation) est un format de données textuel, générique, dérivé de la notation des objets du langage ECMAScript. Il permet de représenter de l'information structurée. Créé par Douglas Crockford, il est décrit par la RFC 4627 de l'IETF.

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

1. Json / Ajax – Json

```
{
  "menu":
  {
    "id": "file",
    "value": "File",
    "popup":
    {
      "menuitem":
      [
        { "value": "New", "onclick": "CreateNewDoc()" },
        { "value": "Open", "onclick": "OpenDoc()" },
        { "value": "Close", "onclick": "CloseDoc()" }
      ]
    }
  }
}
```



1. Json / Ajax – Json – Encode - Php

```
string json_encode (  
    mixed $value  
    [, int $options = 0 ]  
)
```

Retourne une chaîne contenant la représentation JSON de la valeur value.



1. Json / Ajax – Json – Encode - Php

```
json_encode(array("Pêche", "Pomme", "Poire"));
```

```
=> ["Pêche", "Pomme", "Poire"]
```

```
json_encode(array(4 => "Mauvais", 18 => "Bon"));
```

```
=> {"4": "Mauvais", "18": "Bon"}
```

```
json_encode(array("IUT" => true, "Fb" => null));
```

```
=> {"IUT": true, "Fb": null}
```



1. Json / Ajax – Json – Decode - Php

```
$string = '{"vive": "Linux", "autre": "chaine"}';
```

```
$result = json_decode($string);  
var_dump($result);
```

```
object(stdClass)#1 (2) {  
    ["vive"]=> string(5) "Linux"  
    ["autre"]=> string(6) "chaine"  
}
```

```
echo $result->vive; // "Linux"  
echo $result->autre; // "chaine"
```



1. Json / Ajax – Json – JavaScript

JSON = JavaScript Object Notation

=> C'est un sous ensemble de JavaScript

=> En JavaScript, on peut écrire directement en JSON

```
var myJSONObject = {  
  "a": [  
    {"b": "c", "d": "e", "r": "^http://.*"},  
    {"g": "h", "i": "j", "r": "^dee.*"},  
    {"k": "l", "m": "n", "r": "^rx.*"}  
  ]  
};  
myJSONObject.a[1].r  
=> "^dee.*"
```



1. Json / Ajax – Ajax

[Ajax \(informatique\) - Wikipédia](#)

[fr.wikipedia.org/wiki/Ajax_\(informatique\)](https://fr.wikipedia.org/wiki/Ajax_(informatique))

L'architecture informatique **Ajax** (acronyme d'**Asynchronous JavaScript and XML**) permet de construire des applications Web et des sites web dynamiques ...

[Le principe - Histoire - Les technologies utilisées - Ajax et les applications Web ...](#)

[jQuery.ajax\(\) | jQuery API Documentation](#)

api.jquery.com/jquery.ajax/ - Traduire cette page

A set of key/value pairs that configure the **Ajax** request. All settings are optional. A default can be set for any option with \$.ajaxSetup(). See jQuery.ajax(settings) ...

[Ajax - jQuery.ajaxSetup\(\) - Ajax Events](#)

1. Json / Ajax – Ajax

Ajax (acronyme d'Asynchronous JavaScript and XML) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005. Il combine JavaScript, les CSS, XML, le DOM et le XMLHttpRequest afin d'améliorer maniabilité et confort d'utilisation des Applications Internet Riches (abr. RIA)^{1,2...}

[http://fr.wikipedia.org/wiki/Ajax_\(informatique\)](http://fr.wikipedia.org/wiki/Ajax_(informatique))

1. Json / Ajax – Ajax

Les échanges de données entre client et serveur peuvent utiliser divers formats, tels que JSON.

Les applications Ajax fonctionnent sur tous les navigateurs Web courants : Mozilla Firefox, Konqueror, Google Chrome, Safari, Opera, Chromium, Internet Explorer, etc.

[http://fr.wikipedia.org/wiki/Ajax_\(informatique\)](http://fr.wikipedia.org/wiki/Ajax_(informatique))

1. Json / Ajax – Ajax

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title></title>
    <link rel="stylesheet" media="screen" href="style.css">
    <script src="jquery-1.x.x.min.js"></script>
    <script src="script.js"></script>
  </head>
  ...
```

[http://fr.wikipedia.org/wiki/Ajax_\(informatique\)](http://fr.wikipedia.org/wiki/Ajax_(informatique))

1. Json / Ajax – Ajax

```
...  
<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>R&eacute;sultat</legend>  
      <p id="result"></p>  
    </fieldset>  
    <p><button>Soumettre</button></p>  
  </form>  
</body>  
</html>
```



Internet : jQuery / jQuery Mobile

1. Json / Ajax – Ajax

```
$(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);
}
```



1. Json / Ajax – Ajax Equivalent plus léger

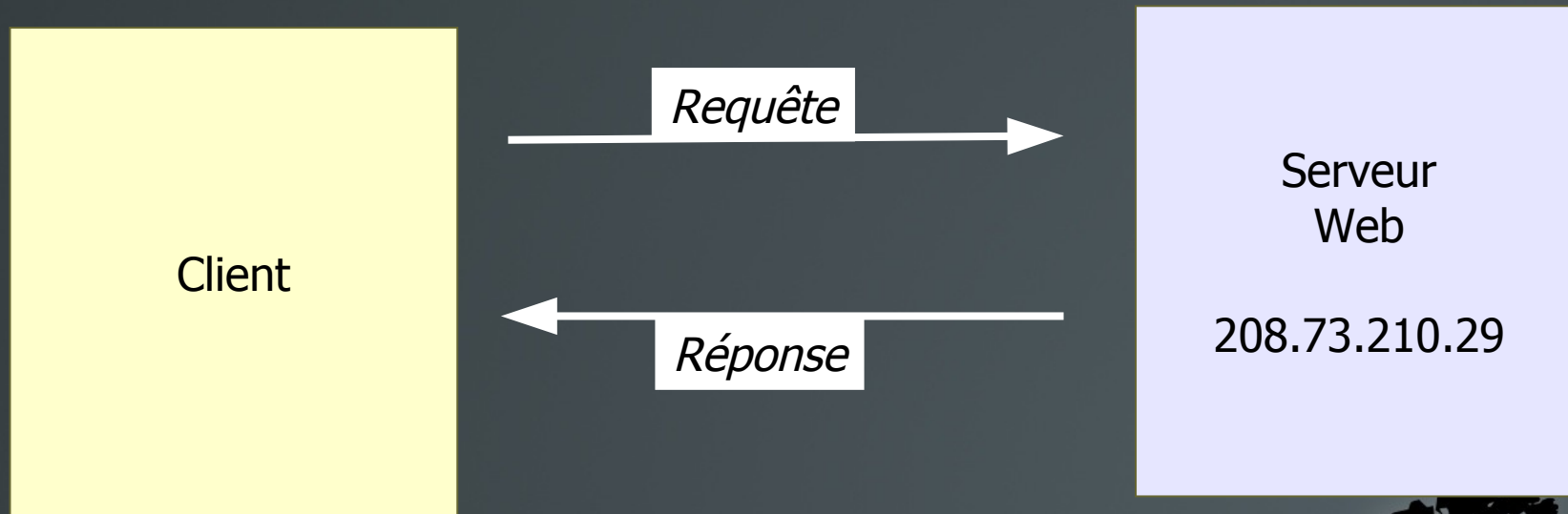
```
$(document).ready(function() {  
    $("form").submit(function (data){  
        $.ajax({  
            type: $(this).attr("method"),  
            url: $(this).attr("action"),  
            data: $(this).serialize(),  
        });  
        return false;  
    }).done(function() {  
        $("#result").html(result);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



Internet : jQuery / jQuery Mobile

1. Json / Ajax – Ajax

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



1. Json / Ajax – Ajax

```
<?php
```

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

```
?>
```



2. jQuery – Présentation

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title></title>
    <link rel="stylesheet" media="screen" href="style.css">
    <script src="jquery-1.x.x.min.js"></script>
    <script src="script.js"></script>
  </head>
  ...
```



2. jQuery – Présentation

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



Internet : jQuery / jQuery Mobile

2. jQuery – Présentation

Showcases jQuery

<http://usejquery.com/sites>

<http://www.exitzeroproject.org/>

<http://like-there-is-no-tomorrow.com/>



Internet : jQuery / jQuery Mobile

2. jQuery – Présentation

Pour les développeurs :
responsive design

<http://getbootstrap.com/>



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

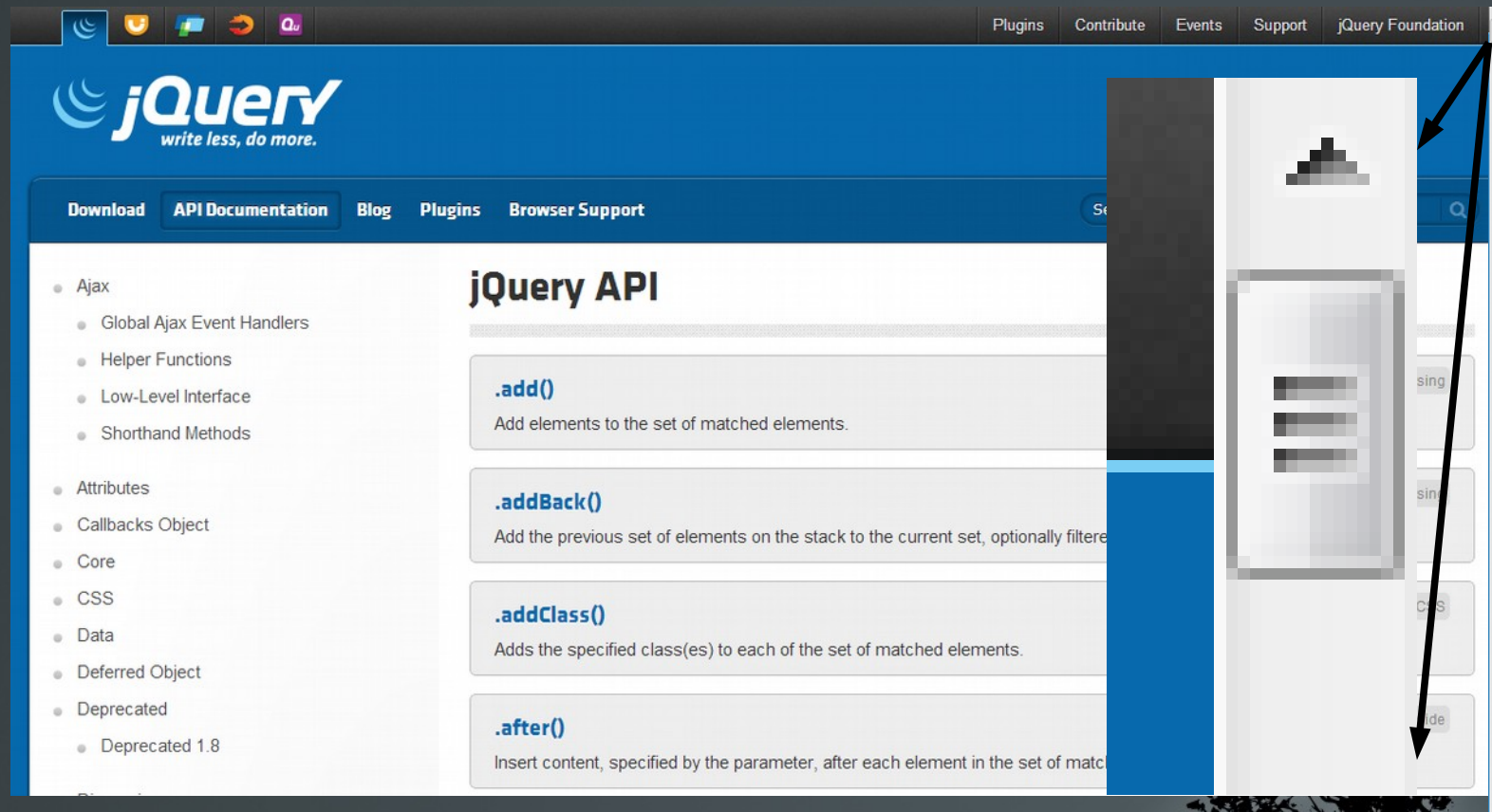
Aujourd'hui : 3 parties

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 has a blue header with the jQuery logo and the 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()". Each method is accompanied by a brief description. A vertical white overlay is positioned on the right side of the page, partially obscuring the content. A black arrow points from the top right of the overlay down to the bottom right, indicating a scroll or transition.

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.

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()
```

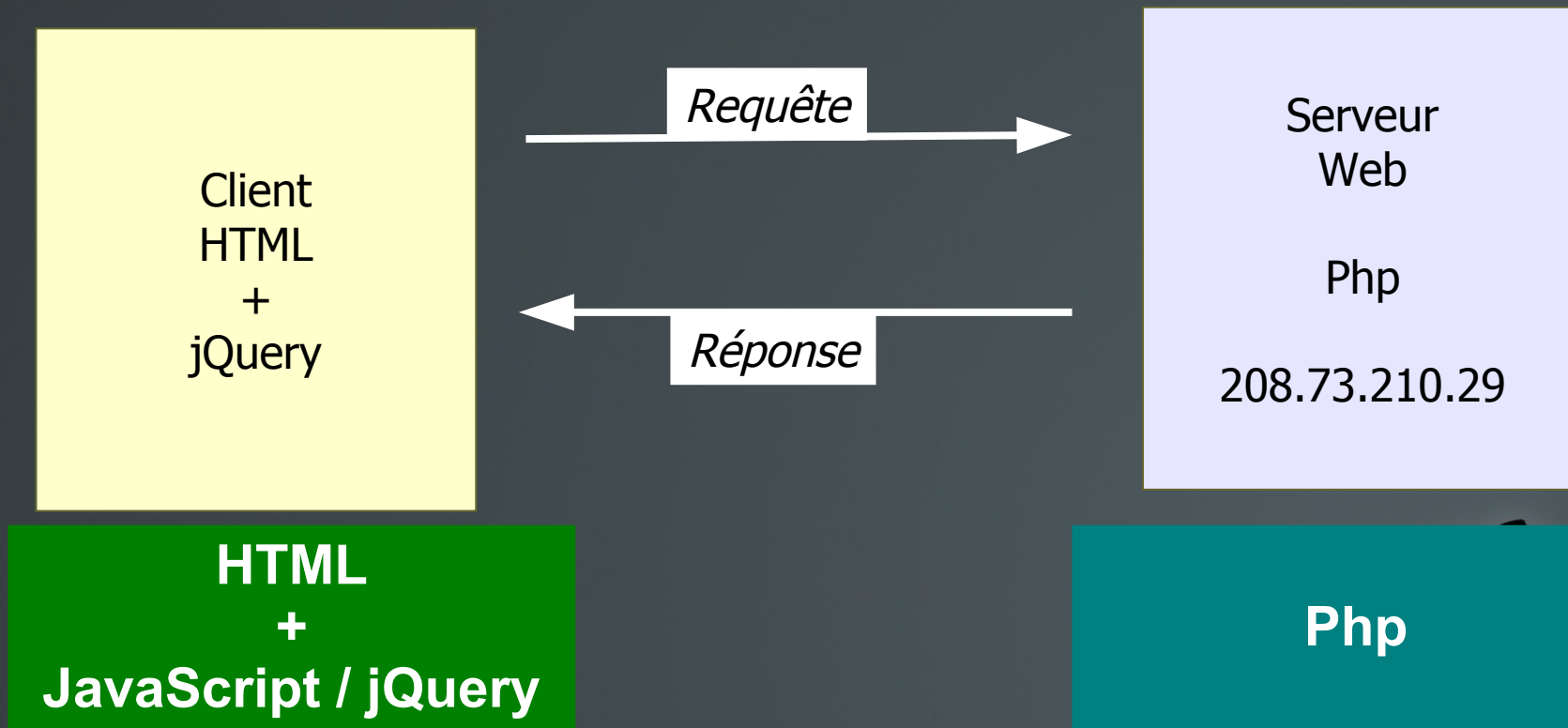
Soit :

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



2. jQuery – AJAX – Rappel

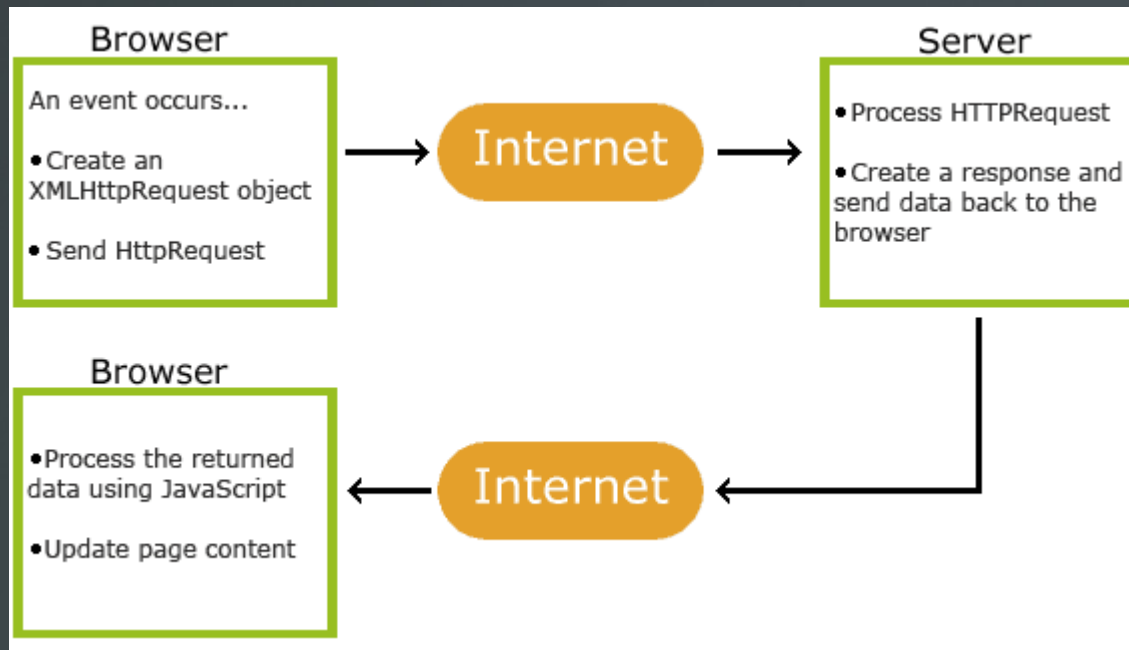
`http://monsiteweb.fr/post.php`



Internet : jQuery / jQuery Mobile / Part. 2

2. jQuery – AJAX – Aide

http://www.w3schools.com/ajax/ajax_intro.asp



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



2. jQuery – AJAX – Deferred

```
var d1 = new $.Deferred();
var d2 = new $.Deferred();
var d3 = new $.Deferred();

$.when( d1, d2, d3 ).done(function ( v1, v2, v3 ) {
    console.log( v1 ); // v1 = undefined
    console.log( v2 ); // v2 = "abc"
    console.log( v3 ); // v3 = array [ 1, 2, 3, 4, 5 ]
});

d1.resolve();
d2.resolve( "abc" );
d3.resolve( 1, 2, 3, 4, 5 );
```

2. jQuery – AJAX – Deferred

Créer un objet "Promise" = un seul paramètre :

```
$.when( unObjet )
```

Exemple:

```
$.when( $.ajax( "/mapageweb" ) )
```

Y attacher des événements :

```
$.when( $.ajax( "/mapageweb" ) )  
  .then(function( data, textStatus, jqXHR ) {  
    alert( jqXHR.status ); // Alerts 200  
  });
```

2. jQuery – AJAX – Deferred

- Quand l'objet est construit :
`$.then(function() {});`
- Quand tous les objets ont **correctement** fini :
`$.done(function() {});`
- Quand **au moins un** des objets a eu une erreur :
`$.error(function() {});`
- Quand tout est fini, erreur ou pas :
`$.always(function() {});`



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').click();
```

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

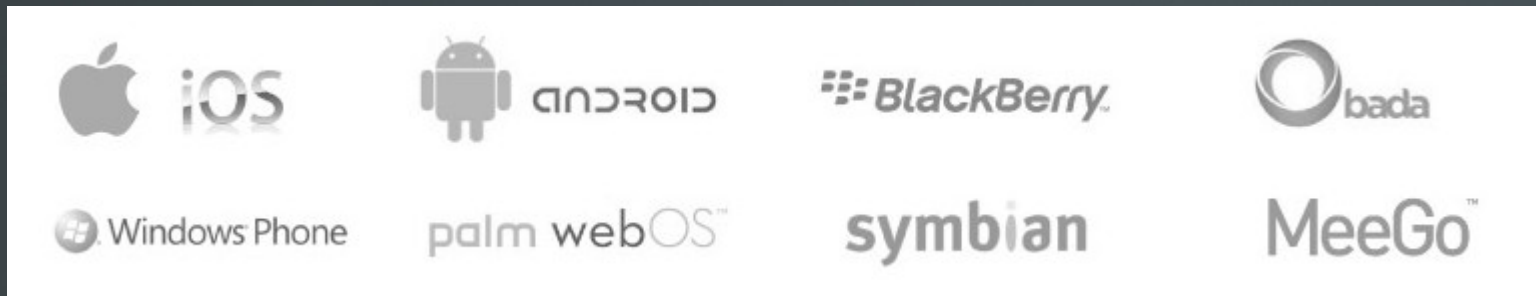
Exemple `empty()` `append()` ... `.click()` :

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



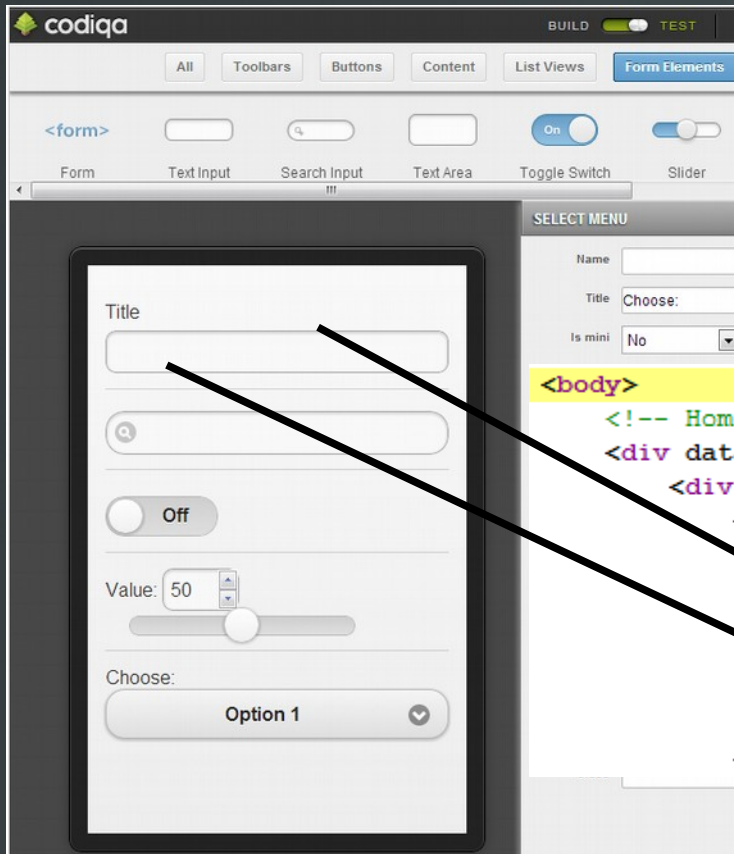
3. jQuery Mobile – API

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

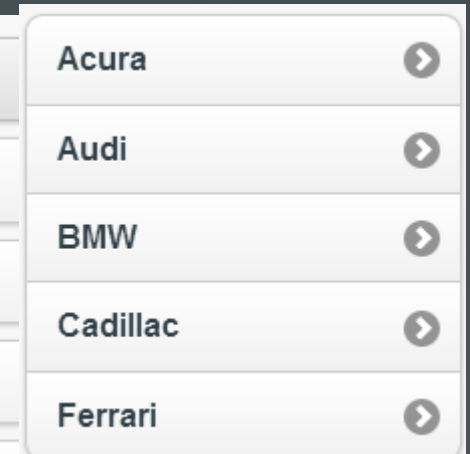
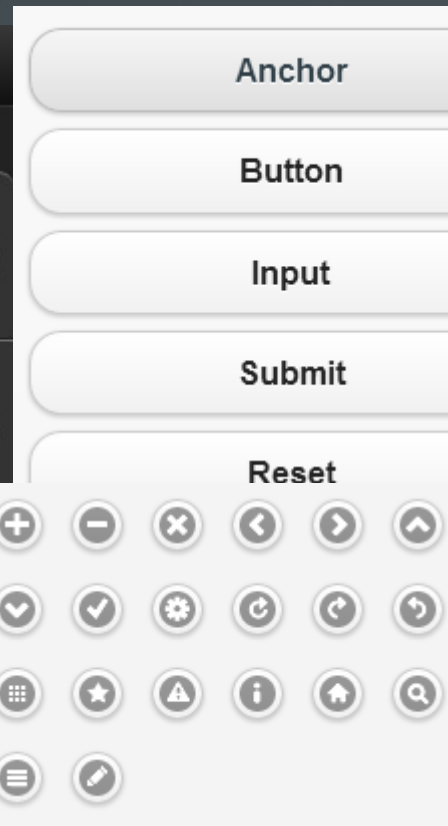
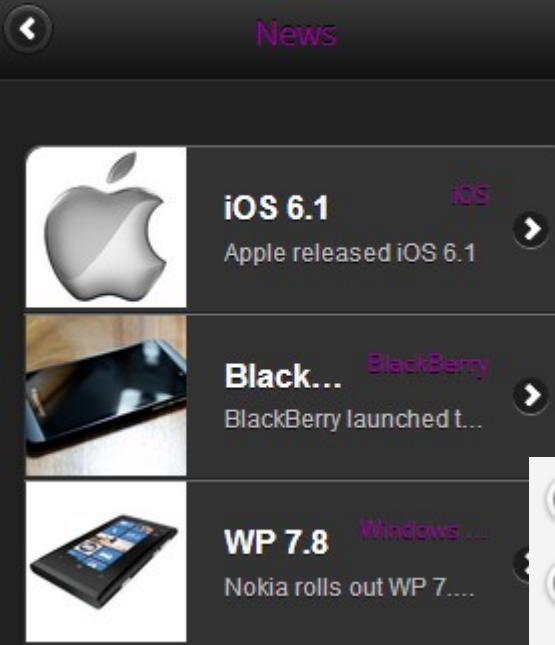
Range slider offer two han
max value along a numeric

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-min="0" data-max="100" data-width="30" data-height="30" data-rel="external" data-rel-external="true" data-rel-popup="true">
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-min="0" data-max="100" data-width="30" data-height="30" data-rel="external" data-rel-external="true" data-rel-popup="true">
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-min="0" data-max="100" data-width="30" data-height="30" data-rel="external" data-rel-external="true" data-rel-popup="true">
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>
```