

Internet

jQuery jQuery Mobile Partie 1

Olivier Pons / 2013



Objectif

Rappel Web / Json / Ajax

Sommaire

1. Fondamentaux
2. Hotes virtuels
3. Règles de réécriture
4. Principes Json / Ajax
5. Présentation jQuery

1. Fondamentaux - DNS

fsf.com => 208.73.210.29

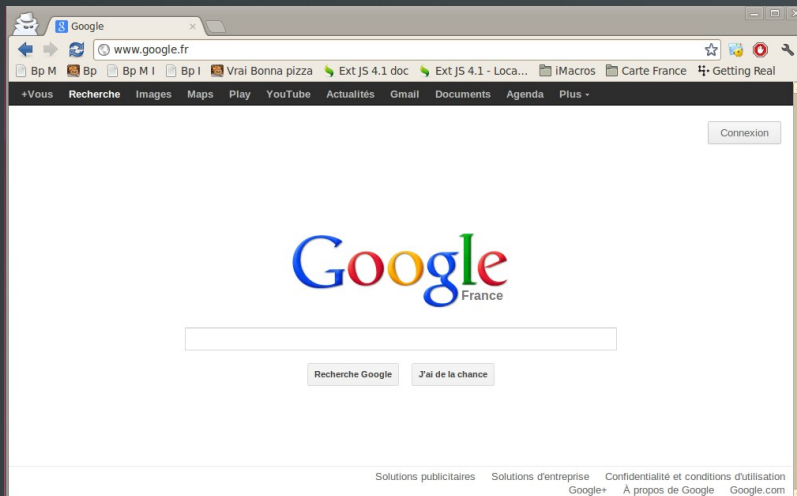
gnu.org => 140.186.70.148

...



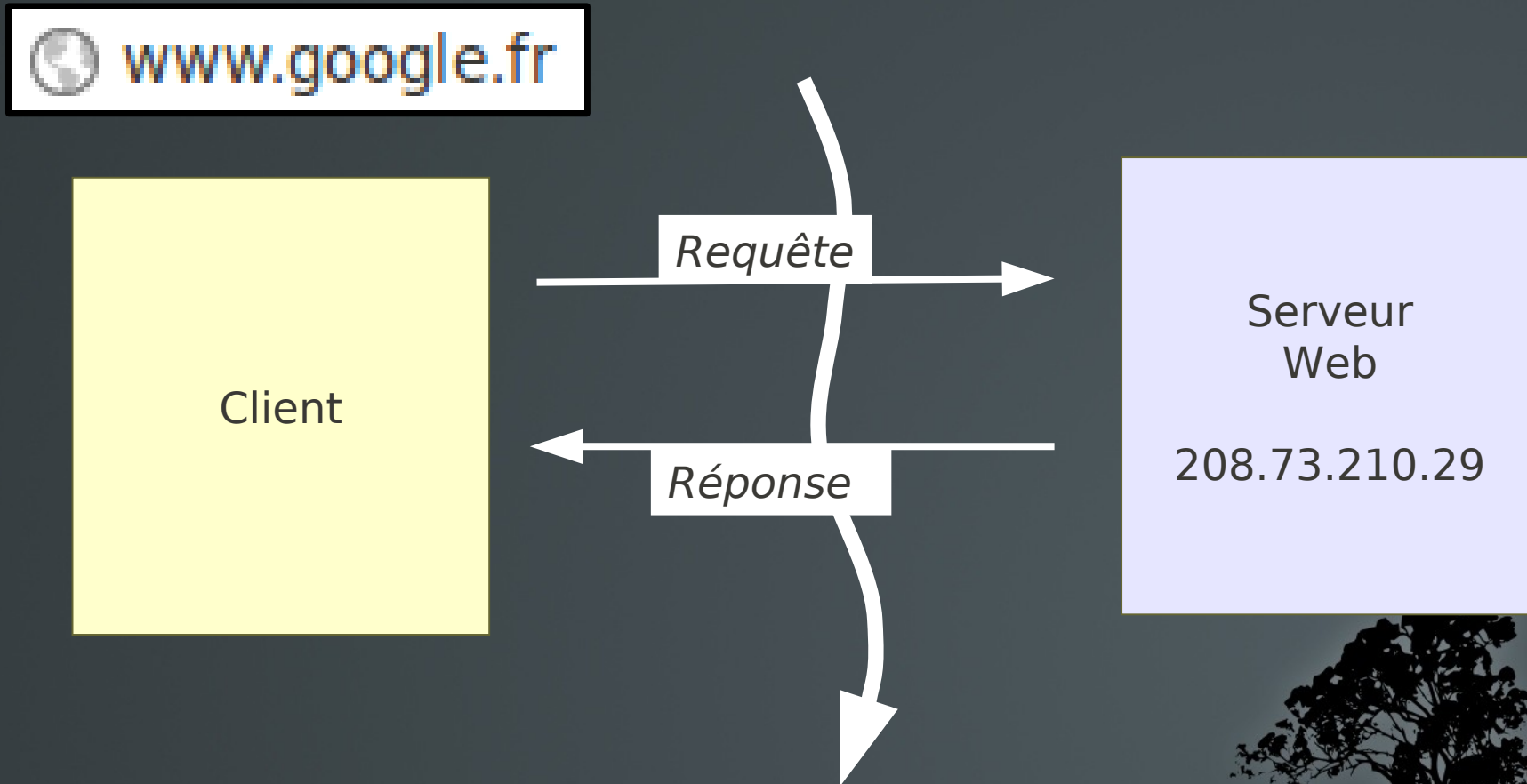
Internet : jQuery / jQuery Mobile / Part. 1

1. Fondamentaux - DNS



=> 208.73.210.29

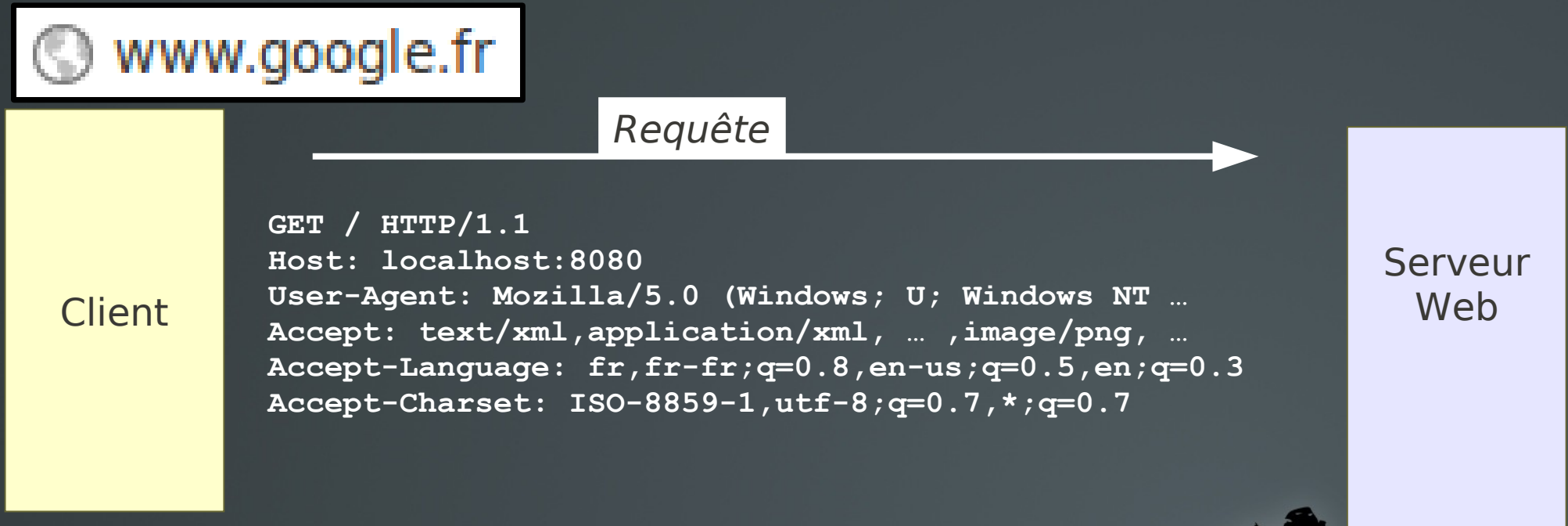
1. Fondamentaux – Client Serveur



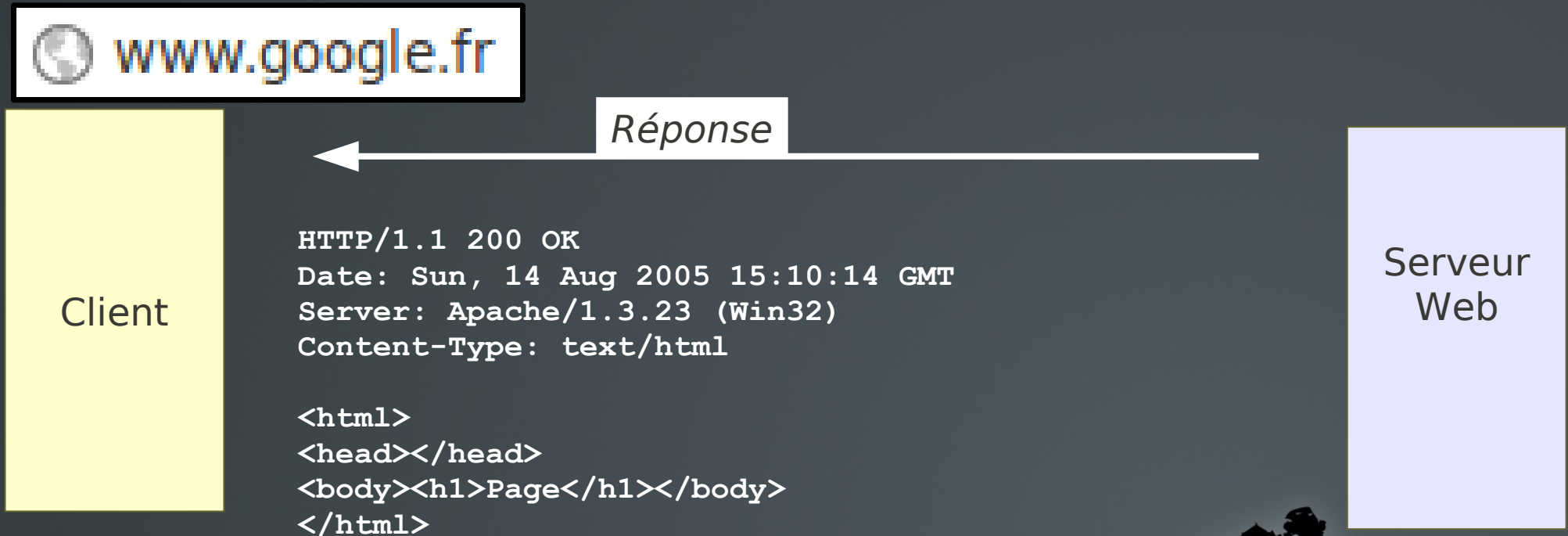
1. Fondamentaux - Protocole

- 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 »
 - Connections courtes, « Sans état » (stateless)

1. Fondamentaux – Client Serveur



1. Fondamentaux – Client Serveur



Internet : jQuery / jQuery Mobile / Part. 1

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)

2. VirtualHosts

```
>ping olivierpons.fr
```

```
PING olivierpons.fr (88.191.136.228)  
56(84) bytes of data.
```

```
>ping keemy.com
```

```
PING keemy.com (88.191.136.228)  
56(84) bytes of data.
```



2. VirtualHosts

Le premier virtualhost est le virtualhost par défaut.

```
<VirtualHost *:80>  
    ServerName keemy.fr  
    ServerAlias www.keemy.fr  
    DocumentRoot /var/www/keemy/  
</VirtualHost>
```

3. Règles de ré-écriture

Expressions régulières : en général

- + Une fois ou plus chu+t : “chut”, “chuut”, “chuuut”, etc.
- * Zéro ou plus chu*t : “cht”, “chut”, “chuut”, etc.
- | “ou” (oli|pons) : oli <=> pons
- ^ Au début ^(a|o)live : “alive”, “olive”
- \$ A la fin (passe)*moilesel\$
- . N'importe quel car.
 - a(.*)ur : “arthur”
 - “ah, oui ! Marlène Sasoeur”



3. Règles de ré-écriture

Expressions régulières : exemple concret

```
http://(.*) test (.*) \.(fr|com)/
```

```
http://un_site_test_qui_est.fr/
```

```
$1 = un_site_
```

```
$2 = _qui_est
```

```
$3 = fr
```



3. Règles de ré-écriture

Exemple simple

`http://oom.papdevis.fr/`



`http://papdevis.fr/index.php?p=oom`

`http://ima.papdevis.fr/`

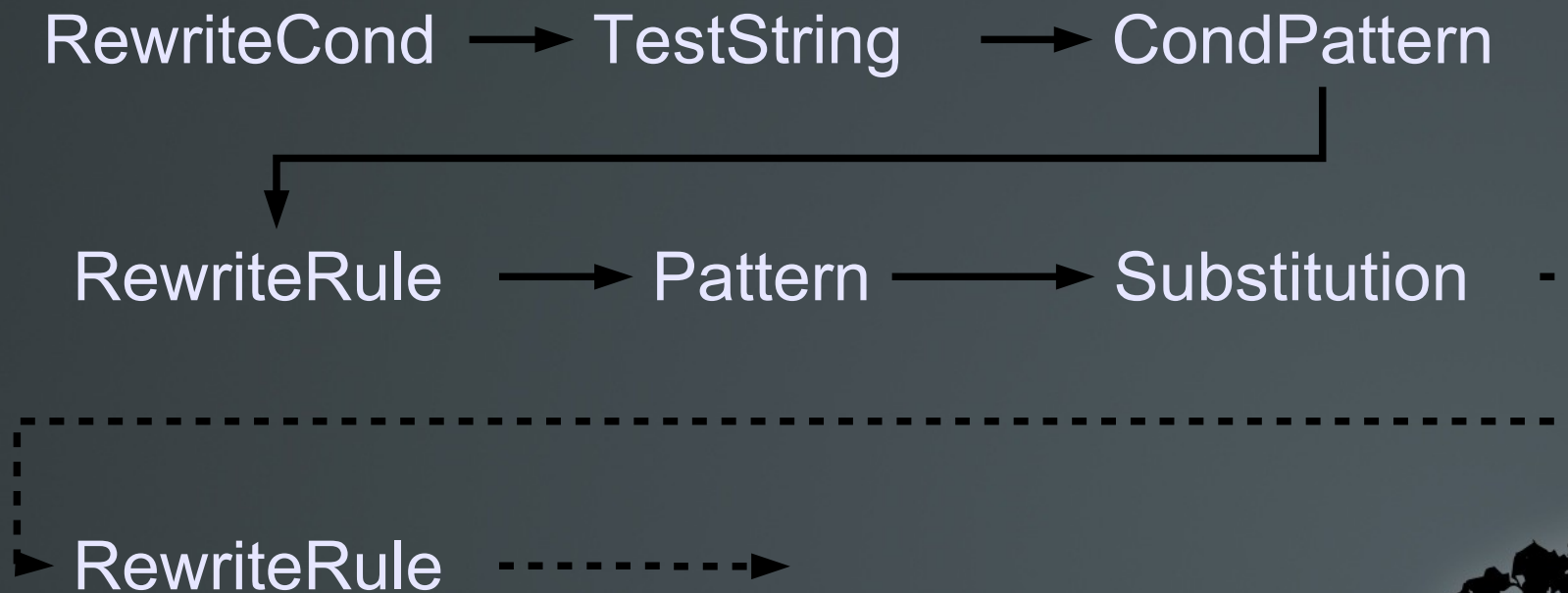


`http://papdevis.fr/index.php?p=ima`



3. Règles de ré-écriture

Principe : en pratique



3. Règles de ré-écriture

Exemple

RewriteCond → %{HTTP_HOST} → http://monsite.uk

RewriteRule (.*) http://us.monsite.com\$1 [R=301,L]

http://monsite.uk/client/



http://us.monsite.com/client/

3. Règles de ré-écriture

Exemple avancé

```
RewriteCond %{HTTP_USER_AGENT} android.+mobile|
avantgo|bada\/|blackberry|blazer|... [NC,OR]
RewriteCond %{HTTP_USER_AGENT} compal|elaine|
fennec|hiptop|iemobile... [NC,OR]
RewriteCond %{HTTP_USER_AGENT} ^(1207|6310|6590|
3gso|4thp|50[1-6]i|770s|802s|a\ wa|a-[2-7]...) [NC]
RewriteCond %{HTTP_HOST} ^(([a-zA-Z0-9\ -]+)\.)
+)([a-zA-Z0-9\ -]+)\.(fr|com|net|org|eu)$
RewriteRule (.*) http://%1m.%4.%5/$1 [QSA,R,L]
```

2. Règles de ré-écriture

Variables Apache internes au serveur

DOCUMENT_ROOT

SERVER_ADMIN

SERVER_NAME

SERVER_ADDR

SERVER_PORT

SERVER_PROTOCOL

SERVER_SOFTWARE

TIME_ (YEAR | MON | DAY | HOUR | MIN | SEC | WDAY)

TIME

API_VERSION

THE_REQUEST

REQUEST_URI

REQUEST_FILENAME

IS_SUBREQ

HTTPS



3. Règles de ré-écriture

Variables Apache : en-têtes HTTP

HTTP_USER_AGENT
HTTP_COOKIE
HTTP_HOST
HTTP_ACCEPT

HTTP_REFERER
HTTP_FORWARDED
HTTP_PROXY_CONNECTION
REMOTE_ADDR

REMOTE_HOST
REMOTE_USER
REQUEST_METHOD
PATH_INFO
AUTH_TYPE

REMOTE_PORT
REMOTE_IDENT
SCRIPT_FILENAME
QUERY_STRING



4. 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

4. Json / Ajax – Json

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

4. Json / Ajax – Json – Encode - Php

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

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

4. 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}
```

4. 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"
```

4. 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.*"
```

4. Json / Ajax – Ajax

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

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

4. 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))

4. 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))

4. 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))

4. Json / Ajax – Ajax

...

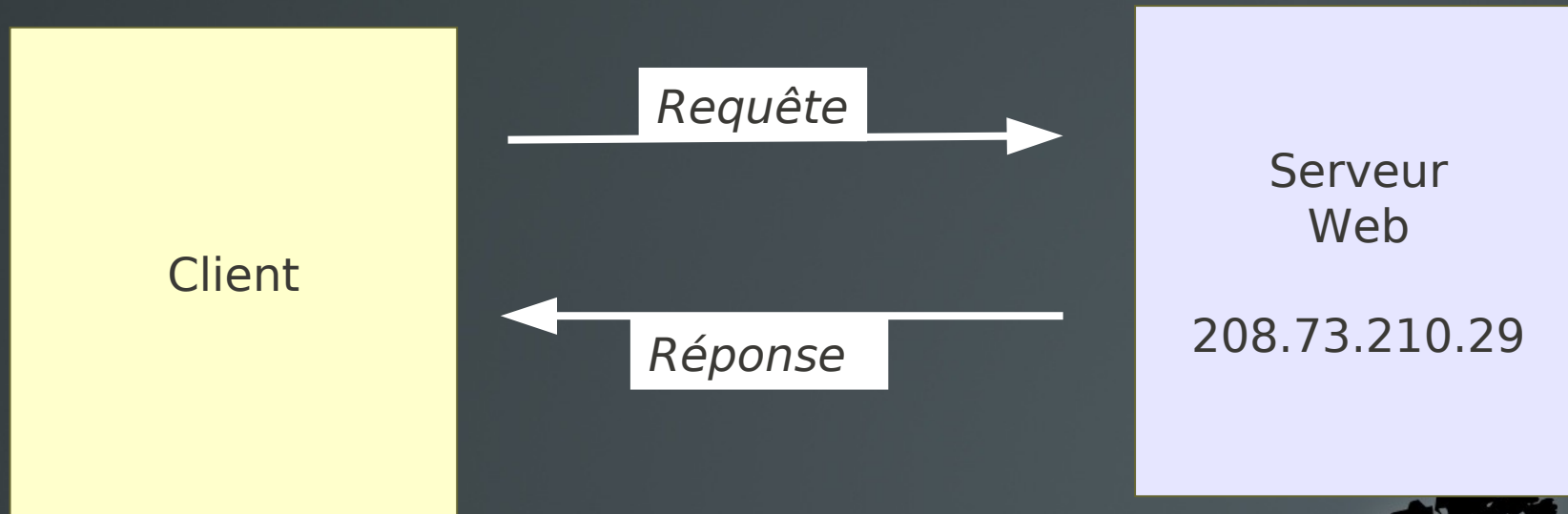
```
<body>
  <form method="post" action="add.php">
    <fieldset>
      <legend>Choisissez deux nombres entiers</legend>
      <p><label>a =
        <input name="a" type="number" required></label></p>
      <p><label>b =
        <input name="b" type="number" required></label></p>
    </fieldset>
   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

4. 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);
}
```

4. Json / Ajax – Ajax

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



4. Json / Ajax – Ajax

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

5. 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>
  ...
```

5. jQuery – Présentation

1) AJAX

2) DOM

- Effets

- Manipulation

- Parcours

3) Gestion des événements



5. jQuery – Présentation

Showcases jQuery

<http://usejquery.com/sites>

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

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



5. jQuery – Présentation

Pour les développeurs :
responsive design

<http://twitter.github.com/bootstrap/>

