

Javascript

Séance 2

Contenu

- ✦ Requete AJAX simple
- ✦ Se souvenir du contexte
- ✦ Requete / cross-domaine
- ✦ JSON & XML

Pour la suite

2 fonctions

```
function ajouterTexteDansBody (texte) {  
    document.getElementsByTagName("body")[0].  
        appendChild(document.createTextNode(texte));  
}  
  
function ajouterEolDansBody() {  
    document.getElementsByTagName("body")[0].  
        appendChild(document.createElement("BR"));  
}
```

+ Utilisation de l'API prototype.js

Requete AJAX simple

Somme.js

```
ajouterTexteDansBody(" 2 + 3 = ");  
new Ajax.Request ("Somme.php", // prototype.js  
    {  
        method : "post", // type de requête HTTP  
        parameters : { a : 2, b : 3}, // paramètres de la requête HTTP  
        onSuccess : function (transport) {  
            // function invoquée quand une réponse  
            // est retournée par le serveur  
            ajouterTexteDansBody(transport.responseText);  
        },  
        onFailure : function () { ajouterTexteDansBody("Erreur");}  
        // function invoquée quand le serveur émet une erreur  
    }  
);
```

Somme.php

```
<?php  
$a=$_POST['a'];  
$b=$_POST['b'];  
echo ($a+$b);  
?>
```

Affichage

Somme en AJAX
 $2 + 3 = 5$

L'objet transport

```
headerJSON: null
readyState: 4
request: klass
  _complete: true
  body: ""
  method: "post"
  options: Object
  parameters: Object
  transport: XMLHttpRequest
  url: "Somme.php"
  __proto__: subclass
responseJSON: null
responseText: "5"
responseXML: null
status: 200
statusText: "OK"
transport: XMLHttpRequest
```

Plusieurs d'affilées

ou «le problème de la perte de contexte»

9 pour le prix d'1

Somme2.js

```
var b=5;
for (var a=1;a<10;a++) {
    ajouterTexteDansBody(" " + a + " + " + b + " = ");
    new Ajax.Request ("Somme.php",
        {
            method : "post",
            parameters : { a : a, b : b},
            onSuccess : function (transport) {
                ajouterTexteDansBody(transport.responseText);
                ajouterEolDansBody();
            },
            onFailure : function () {
                ajouterTexteDansBody("Erreur");
                ajouterEolDansBody();
            }
        }
    );
}
```


Affichage (aïe)

Somme en AJAX V2

1 + 5 = 2 + 5 = 3 + 5 = 4 + 5 = 5 + 5 = 6 + 5 = 7 + 5 = 8 + 5 = 9 + 5 = 6

7

8

9

10

11

12

13

14

En même temps c'est logique => asynchrone !

Le label au retour ?

Somme3.js

```
var b=5;
for (var a=1;a<10;a++) {
  var label=" " + a + " + " + b + " = ";
  new Ajax.Request ("Somme.php",
    {
      method : "post",
      parameters : { a : a, b : b},
      onSuccess : function (transport) {
        ajouterTexteDansBody(label+transport.responseText);
        ajouterEolDansBody();
      },
      onFailure : function () {
        ajouterTexteDansBody(label+"Erreur");
        ajouterEolDansBody();
      }
    }
  );
}
```

Affichage (re-aïe)

Somme en AJAX V3

$$9 + 5 = 6$$

$$9 + 5 = 7$$

$$9 + 5 = 9$$

$$9 + 5 = 10$$

$$9 + 5 = 11$$

$$9 + 5 = 12$$

$$9 + 5 = 8$$

$$9 + 5 = 13$$

$$9 + 5 = 14$$

C'est pas dans l'ordre et en plus le label est incorrect

Solution ?

Scope forever

Pointer sur des variables volatiles

Somme4.js

```
function somme (_a,_b) {  
    var label=" " + _a + " + " + _b + " = ";  
    new Ajax.Request ("Somme.php",  
        {  
            method : "post",  
            parameters : { a : _a, b : _b},  
            onSuccess : function (transport) {  
                ajouterTexteDansBody(label+transport.responseText);  
                ajouterEolDansBody();  
            },  
            onFailure : function () {  
                ajouterTexteDansBody(label+"Erreur");  
                ajouterEolDansBody();  
            }  
        }  
    );  
}  
var b=5;  
for (var a=1;a<10;a++) {  
    somme(a,b);  
}
```

Affichage (yes !)

Somme en AJAX V4

$$2 + 5 = 7$$

$$3 + 5 = 8$$

$$1 + 5 = 6$$

$$4 + 5 = 9$$

$$5 + 5 = 10$$

$$6 + 5 = 11$$

$$7 + 5 = 12$$

$$8 + 5 = 13$$

$$9 + 5 = 14$$

C'est pas dans l'ordre mais le label est bon :
au retour de la requête les variables correspondaient

Explication...

Somme4.js

```
function somme (_a,_b) {  
    var label=" " + _a + " + " + _b + " = ";  
    // _a, _b et label sont des variables créées à chaque invocation  
    new Ajax.Request ("Somme.php",  
        {  
            method : "post",  
            parameters : { a : _a, b : _b},  
            onSuccess : function (transport) {  
                ajouterTexteDansBody(label+transport.responseText);  
                // cette fonction va garder un pointeur sur la variable  
                // label créée pendant cette invocation  
                ajouterEolDansBody();  
            },  
            onFailure : function () {  
                ajouterTexteDansBody(label+"Erreur");  
                ajouterEolDansBody();  
            }  
        }  
    );  
}
```

Et si le script php est
sur un autre serveur ?

CROSS-DOMAIN TIME!

Passons par <script>

Somme5.js

```
var b=5;
for (var a=1;a<10;a++) {

    // création d'une balise script
    var baliseScript = document.createElement("script");

    // on fixe la src vers le script php
    // avec les bons arguments (donc mode GET)
    baliseScript.setAttribute("src",
        "http://webtp.fil.univ-lille1.fr/~lepallec/Somme5.php?a="+a+"&b="+b);

    // ajout de la balise script créé à la balise body
    document.getElementsByTagName("body")[0].appendChild(baliseScript);
}
```

Et le php ???

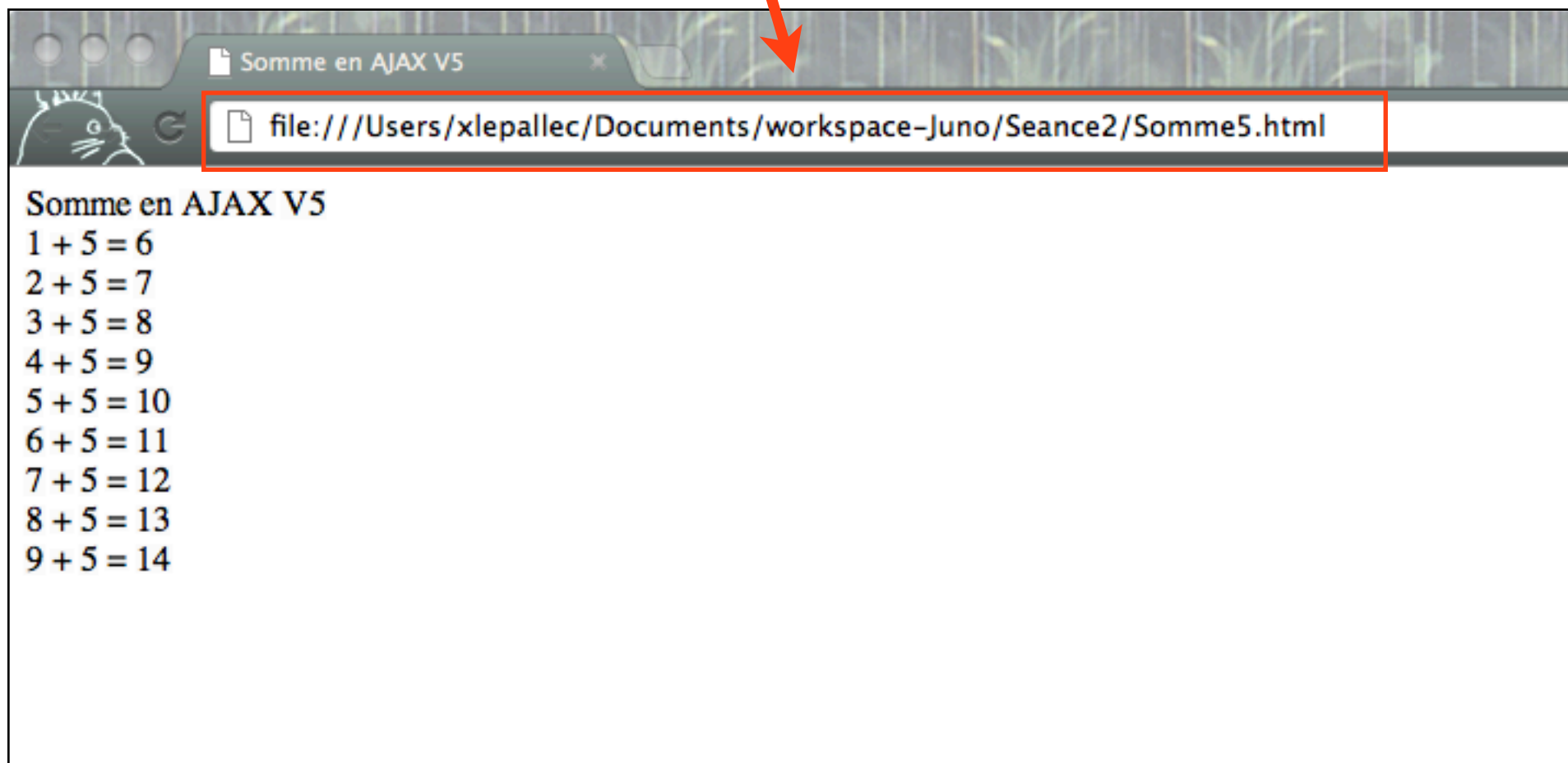
On renvoie du javascript

Somme5.php

```
<?php
$a=$_GET['a']; // MODE GET
$b=$_GET['b']; // MODE GET
$label=$a." + ".$b." = ";
$somme=$a+$b;
header("Content-Type: application/x-javascript");
echo ("ajouterTexteDansBody('$label $somme');ajouterEolDansBody());";
?>
```

Simple mais efficace

Affichage



Même plus besoin de mettre les fichiers html
et js sur un serveur !

JSON vs XML

JSON wins!

JSON

Somme6.js

```
...  
onSuccess : function (transport,json) {  
    ajouterTexteDansBody(label+json.somme);  
    ajouterEolDansBody();  
}  
...
```

JSON

Somme6.php

```
<?php
$a=$_POST['a'];
$b=$_POST['b'];
header("X-JSON: " . json_encode(array ('somme' => $a+$b)));
?>
```

XML

Somme7.js

```
...  
onSuccess : function (transport) {  
    ajouterTexteDansBody(  
        label+  
        transport.responseXML.getElementsByTagName("somme")[0].firstChild.data  
    );  
    ajouterEolDansBody();  
}  
...
```

XML

Somme7.php

```
<?php
$a=$_POST['a'];
$b=$_POST['b'];
$somme=$a+$b;
header ("Content-Type:text/xml");
echo("<rien><somme>$somme</somme></rien>");
?>
```