

JavaScript

Comprendre les bases

J. Saraydaryan

- I Introduction
- II Les Bases
- III Utilisation du DOM
- IV Introduction à JQuery
- V Les Communications



Introduction

JAVASCRIPT

- **Qu'est ce que le JavaScript ?**

- ❑ Ce n'est pas JAVA!!
- ❑ Un code interprété (!= compilé comme java ou C)
- ❑ Utilisé pour effectuer du traitement d'information sur les navigateurs clients
- ❑ S'intègre dans les fichiers html
- ❑ Possibilité d'interactions avec l'utilisateur plus grande



JAVASCRIPT

• A quoi ça peut servir ?

- ❑ Ecrire à l'intérieur du document html

```
document.write("<h1>This is a heading</h1>");
```

- ❑ Réagir à des évènements

```
<button type="button" onclick="alert('Welcome!')">Click  
Me!</button>
```

- ❑ Changer du contenu HTML

```
x=document.getElementById("demo")  
x.innerHTML="Hello JavaScript";
```

- ❑ Changer le style d'un élément HTML

```
x=document.getElementById("demo")  
x.style.color="#ff0000";
```

- ❑ Valider des entrées utilisateurs

```
if isNaN(x) {alert("Not Numeric");}
```



JAVASCRIPT

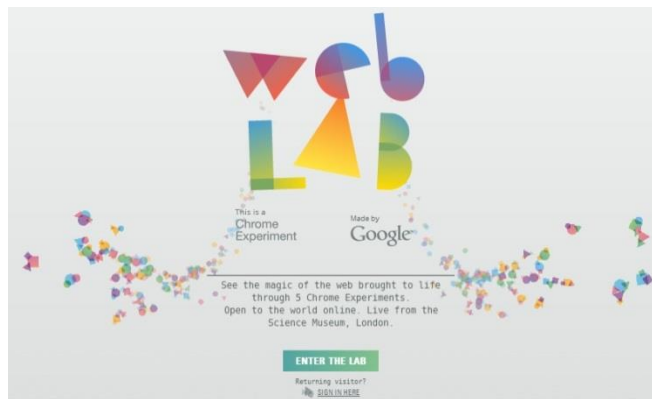
- Et à bien d'autres choses....



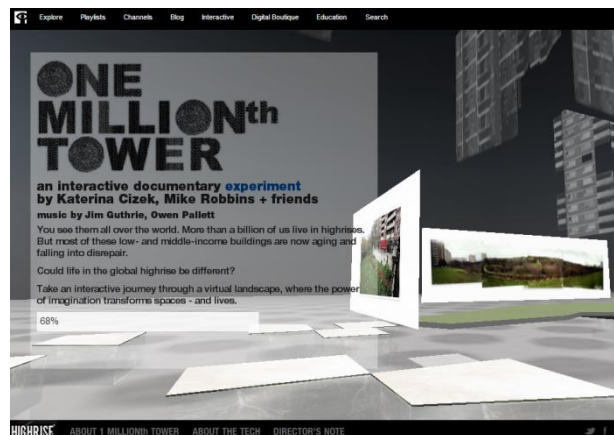
<http://www.goengine.com/demofiles/pearl-boy/index.html>

JAVASCRIPT

- Et à bien d'autres choses....



<http://www.chromeweb.com/>



http://highrise.nfb.ca/onemillionthtower/1mt_webgl.php

Source: <http://blog.tangcs.com/2011/04/15/html-css-tutorials-intro/>

Copyright © 2012-2013 Jacques Saraydaryan

Les Bases

JAVASCRIPT

• Comment ça marche ?

- ❑ Utilisation de variables non typées
- ❑ La portée d'une variable peut être locale ou globale
- ❑ Création de fonction avec arguments et valeur de retour

```
<script>  
  var threshold=500;  
  
  /*  Multiline  
      Comment */  
  
  function somme(a,b) {  
    //simple line comment  
    var sum=a+b;  
    if(sum> threshold){  
      alert("Threshold Exceeded");  
    }  
    return sum;  
  }  
</script>
```

Variable globale

Commentaires

Fonction

Variable locale

JAVASCRIPT

• Utilisation

□ Interagir avec le document HTML

- 1 Intégration des scripts directement dans le HTML
- 2 Intégration d'un fichier de script extérieur au document HTML
- 3 Appel de fonctions dans le code HTML par le biais d'évènements
- 4 Utilisation des objets HTML par le script
- 5 Modification du document HTML par le script



JAVASCRIPT

• Utilisation

```
<!DOCTYPE html>
<html>
  <header>
    <script type="text/javascript">
      document.write("Hello World!");
    </script>
  </header>
  <body>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <header>
    <script type="text/javascript"
      src="MonFichier.js">
    </script>
  </header>
  <body>
  </body>
</html>
```

1

Intégration des scripts
directement dans le
HTML

2

Intégration d'un fichier de
script extérieur au
document HTML

JAVASCRIPT

- Utilisation

```
<!DOCTYPE html>
<html>
  <header>
    <title>My HTML and Script</title>
    <script type="text/javascript">
      function helloWorldA() {
        window.alert("Hello World!A");
      }

      function helloWorldClick() {
        window.alert("Hello World on Click !");
      }

      function helloWorldMouseOver() {
        window.alert("Hello World MouseOver!");
      }
    </script>
  </header>
  <body onload="helloWorldA()">
    <form>
      <input type="button" value="B" onclick="helloWorldClick()" />
    </form>
    <div onmouseover="helloWorldMouseOver()">
      <p>
        Just come over this section
      </p>
    </div>
  </body>
</html>
```

3

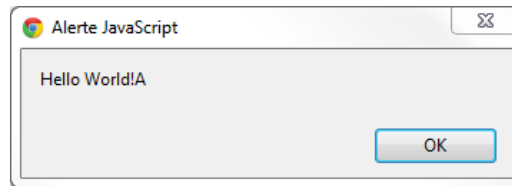
Appel de fonctions dans le code HTML par le biais d'évènements

JAVASCRIPT

- **Scripts (3/8)**

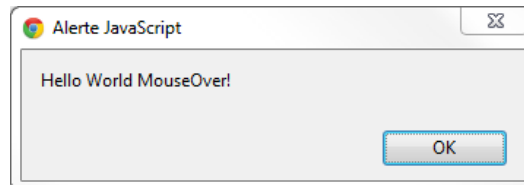
B

Just come over this section



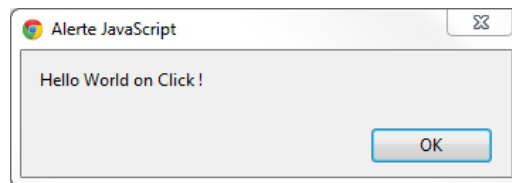
B

Just come over this section



B

Just come over this section



3

Appel de fonctions dans
le code HTML par le biais
d'évènements

JAVASCRIPT

• Les Variables

❑ Type non défini

Attention var i;

i non déterminée i=Undefined

❑ Tableau de données

❑ Structures de données

Les données sont accessibles soit

Variable["attribut"] e.g person["lastname"]

Soit

Variable.attribut e.g person.lastname

```
var i=5; // Integer
var j="jDoe "; // String
var k=5.124; // Float
var x=true; // Boolean
var alpha= 123e5; // notation scientifique 12300000
```

```
var cars=new Array();
cars[0]="Saab";
cars[1]="Volvo";
cars[2]="BMW";
```

Ou

```
var cars=new Array("Saab","Volvo","BMW");
```

Ou

```
var cars=["Saab","Volvo","BMW"];
```

```
var person={firstname:"John", lastname:"Doe", id:5566};
```

Ou

```
var person={
  firstname : "John",
  lastname  : "Doe",
  id       : 5566
};
```

JAVASCRIPT

• Les Variables: Operateurs

Operator	Description	Example	Result of x	Result of y
+	Addition	$x=y+2$	7	5
-	Subtraction	$x=y-2$	3	5
*	Multiplication	$x=y*2$	10	5
/	Division	$x=y/2$	2.5	5
%	Modulus (division remainder)	$x=y\%2$	1	5
++	Increment	$x=++y$	6	6
		$x=y++$	5	6
--	Decrement	$x=--y$	4	4
		$x=y--$	5	4

Operator	Example	Same As	Result
=	$x=y$		$x=5$
+=	$x+=y$	$x=x+y$	$x=15$
-=	$x-=y$	$x=x-y$	$x=5$
=	$x=y$	$x=x*y$	$x=50$
/=	$x/=y$	$x=x/y$	$x=2$
%=	$x\%=y$	$x=x\%y$	$x=0$

JAVASCRIPT

• Les Conditions

□ Expression conditionnelle

- && -> ET logique
- // -> OU logique
- == null → est égale à null
- isNaN() → is not a number

□ Switch

switch(n)

{

case 1:

execute code block 1

break;

case 2:

execute code block 2

break;

default:

code to be executed if n

is different from case 1 and 2

}

```
If( (!isNaN(age) && (age>18)) || (age===-1){  
    // your code  
}else{  
    // your code  
}
```

```
var day=new Date().getDay();  
switch (day)  
{  
case 6:  
    x="Today it's Saturday";  
    break;  
case 0:  
    x="Today it's Sunday";  
    break;  
default:  
    x="Looking forward to the Weekend";  
}
```


JAVASCRIPT

- Les Conditions: Operateurs

Operator	Description	Comparing	Returns
==	is equal to	x==8	<i>false</i>
		x==5	<i>true</i>
===	is exactly equal to (value and type)	x==="5"	<i>false</i>
		x===5	<i>true</i>
!=	is not equal	x!=8	<i>true</i>
!==	is not equal (neither value nor type)	x!=="5"	<i>true</i>
		x!==5	<i>false</i>
>	is greater than	x>8	<i>false</i>
<	is less than	x<8	<i>true</i>
>=	is greater than or equal to	x>=8	<i>false</i>
<=	is less than or equal to	x<=8	<i>true</i>

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
	or	(x==5 y==5) is false
!	not	!(x==y) is true

JAVASCRIPT

• Les Boucles

□ For

```
for (statement 1; statement 2; statement 3)
{
    the code block to be executed
}
```

□ While

```
while (condition)
{
    code block to be executed
}
```

OU

```
do
{
    code block to be executed
}
while (condition);
```

```
for (var i=0; i<5; i++)
{
    x=x + "The number is " + i + "<br>";
}
```

```
var person={fname:"John",lname:"Doe",age:25};
for (x in person)
{
    txt=txt + person[x];
}
```

```
while (i<5)
{
    x=x + "The number is " + i + "<br>";
    i++;
}
```

```
do
{
    x=x + "The number is " + i + "<br>";
    i++;
}
while (i<5);
```

JAVASCRIPT

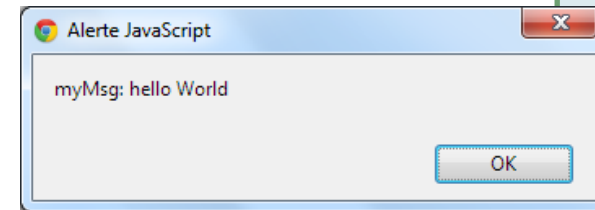
• Popup d'interactions

□ Alert

- Affiche un message d'information à l'utilisateur

```
window.alert(" Hello");
```

Ou
`alert("Hello");`

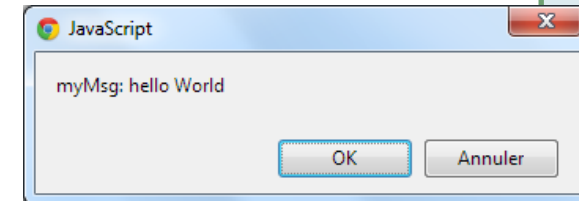


□ Confirm

- Affiche un message de confirmation à l'utilisateur
- Valeur de return = *true* si *ok* *false* si *cancel*

```
window.confirm(" Hello");
```

Ou
`confirm("Hello");`

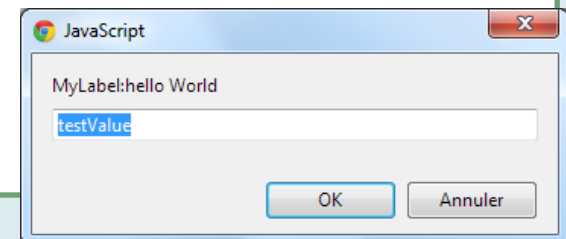


□ Prompt

- Affiche une popup de saisie à l'utilisateur

```
window.prompt("Label", " testValue");
```

Ou
`prompt("Label", " testValue");`



JAVASCRIPT

- A vous de Jouer !

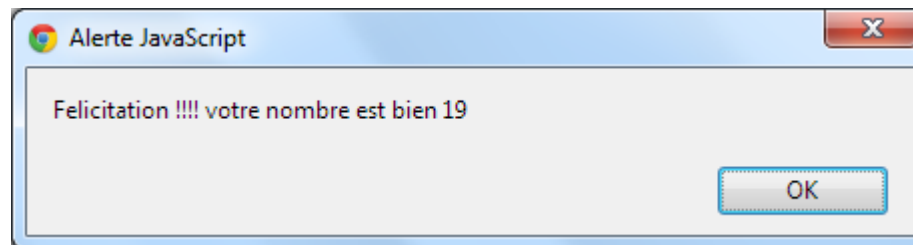
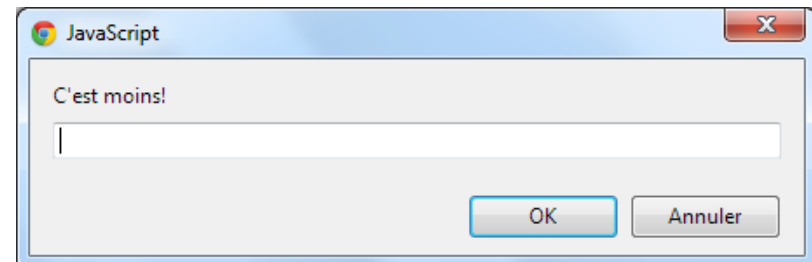
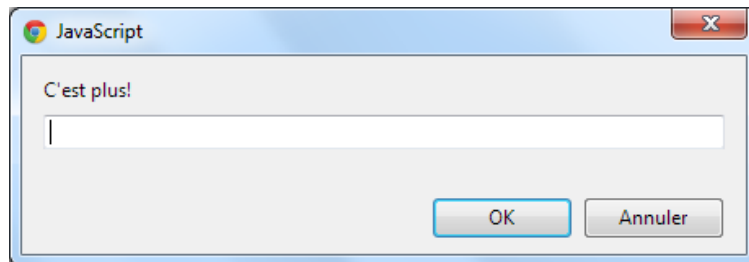
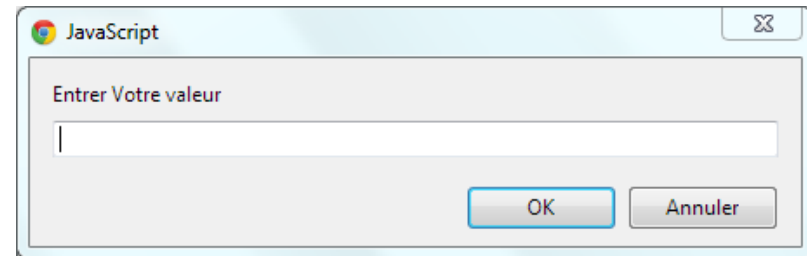
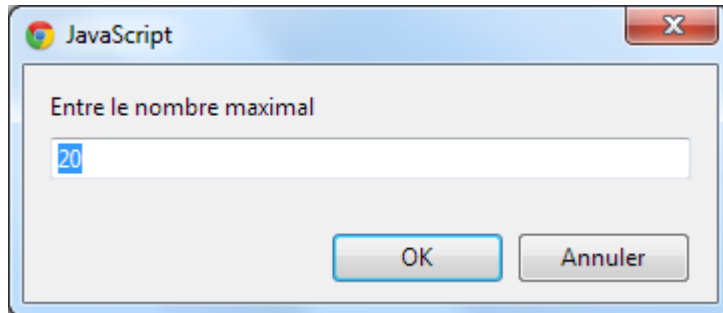
- ❑ Effectuer un jeu permettant de trouver un nombre entre 0 et la valeur maximum entrée par l'utilisateur

- ❑ Utiliser la méthode `Math.floor(Math.random()*n)` permettant de tirer un nombre aléatoire entre 0 et n



JAVASCRIPT

- A vous de Jouer !



Utilisation du DOM

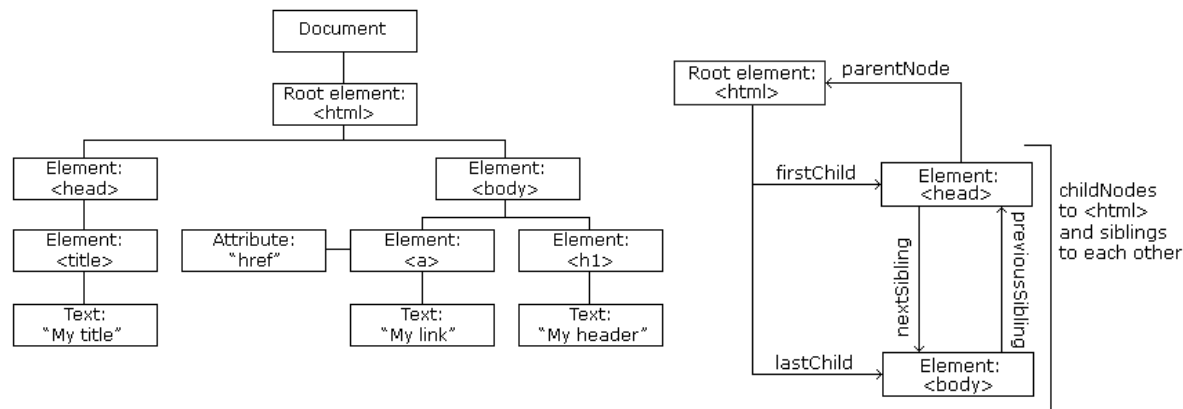
JAVASCRIPT

• Utilisation de DOM Document Model Object

```

<!DOCTYPE html>
<html>
  <header>
    <title>My title</title>
  </header>
  <body >
    <h1>My Header</h1>
    <a href="http://www.w3schools.com">My link</a>
  </body>
</html>

```



4

Utilisation des objets
HTML par le script

a

Description des
données via DOM

JAVASCRIPT

• Utilisation de DOM Document Model Object

Windows object: Fenetre du navigateur (si iframe 1 windows / iframe)

Navigator object: information du navigateur

Screen object: information sur l'écran du visiteur

History object: URLs visitées par l'utilisateur

Location object: Information à propos de l'URL courante

DOM Node: nœud dans le document HTML

DOM NodeList: liste ordonnée de noeuds

DOM NamedNodeMap: liste non ordonnée de nœuds

DOM Document: racine de l'arbre, donne accès aux données du document.

DOM Element: objet représentant un élément du document HTML

DOM Attr: objet représentant un attribut d'un élément HTML



4

Utilisation des objets HTML par le script

a

Description des données via DOM

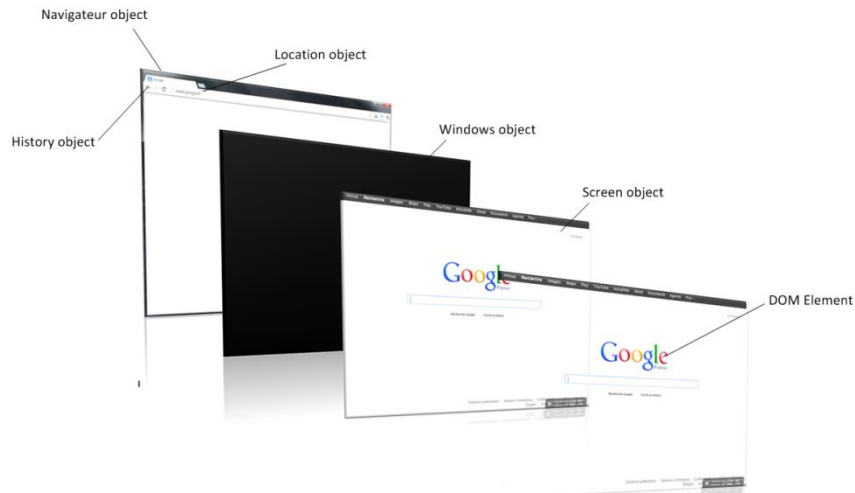
b

Type de données accessibles

JAVASCRIPT

• Utilisation de DOM Document Model Object

Document object	Image object	Link object
Event object	Input Button object	Meta object
HTMLElement object	Input Checkbox object	Object object
Anchor object	Input File object	Option object
Area object	Input Hidden object	Select object
Base object	Input Password object	Style object
Body object	Input Radio object	Table object
Button object	Input Reset object	td / th object
Form object	Input Submit object	tr object
Frame/IFrame object	Input Text object	Textarea object
Frameset object	Link object	



4

Utilisation des objets
HTML par le script

a

Description des
données via DOM

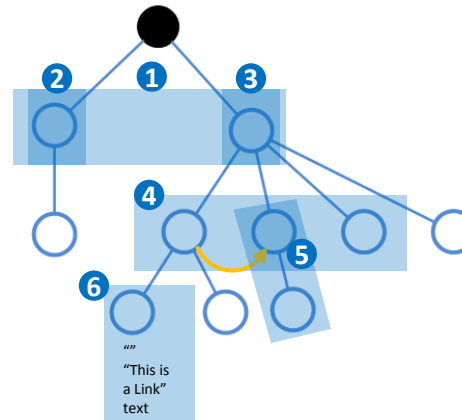
b

Type de données
accessibles

JAVASCRIPT

• Utilisation de DOM Document Model Object

- ❶ **childNodes**: liste des nœuds enfants
 - ❷ **firstChild**: premier enfant du nœud
 - ❸ **lastChild**: dernier enfant du nœud
 - ❹ **nextSibling**: prochain nœud (même niveau)
 - ❺ **parentNode**: nœud parent
 - ❻ **nodeName**: nom du nœud courant
- nodeValue**: valeur/contenu du nœud
- nodeType**: type du nœud



4

Utilisation des objets
HTML par le script

a

Description des
données via DOM

b

Type de données
accessibles

c

Récupérer les données
d'un DOM

JAVASCRIPT

• Utilisation de DOM Document Model Object

```
<html>
  <header>

    <base href="http://www.cpe.fr/" />

    <script type="text/javascript">
      function infoById() {
        var linkVar=document.getElementById("linkId");
        var url=linkVar.href;

        var paraVar=document.getElementById("paraId");
        var url2=paraVar.nextSibling.nextSibling.href;

        window.alert("This is the link URL (direct) "+url
          +"\n\nThis is the link URL (navigation) "+url2);
      }
    </script>

    <title>My First HTML Page</title>
  </header>
  <body onload="infoById()">
    <p id="paraId">
      this is a paragraph of Test
    </p>
    <a id="linkId" href="documents/formation/Specimen_diplome_IRC.pdf">
      this is a link
    </a>
  </body>
</html>
```

4

Utilisation des objets
HTML par le script

a

Description des
données via DOM

b

Type de données
accessibles

c

Récupérer les données
d'un DOM

d

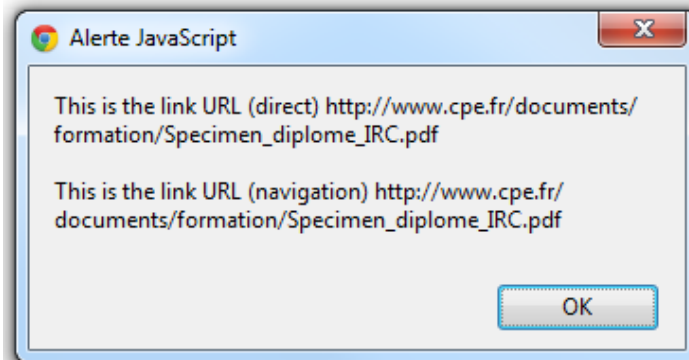
Naviguer dans un DOM

JAVASCRIPT

- Utilisation de DOM Document Model Object

this is a paragraph of Test

[this is a link](#)



4

Utilisation des objets
HTML par le script

a

Description des
données via DOM

b

Type de données
accessibles

c

Récupérer les données
d'un DOM

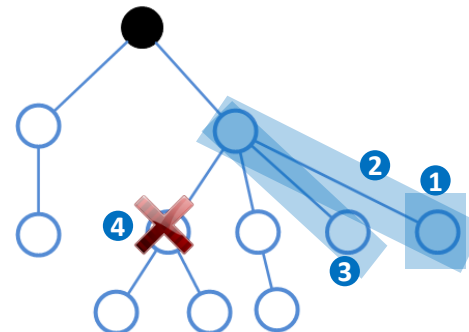
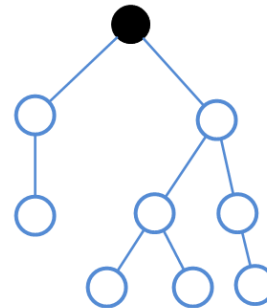
d

Naviguer dans un DOM

JAVASCRIPT

• Utilisation de DOM Document Model Object

- 1 **createElement**: crée un nouvel élément dans le document HTML
createTextNode: crée un noeud de type texte
- 2 **appendChild**: Ajouter l'élément créé (dernière position)
- 3 **insertBefore**: ajoute un élément avant un autre noeud
- 4 **removeChild**: supprimer un noeud



5

Modification du document HTML par le script

JAVASCRIPT

- Modifier la page HTML

- ❑ Ecriture dans le cœur de la page

document.write(your code);

- ❑ Changer la valeur d'un élément

document.getElementById(id).innerHTML

=new HTML

- ❑ Changer la valeur d'un attribut

document.getElementById(id).attribute

=new value

```
document.write(Date());
```

```
<!DOCTYPE html>
<html>
<body>
<h1 id="header">Old Header</h1>
<p id="p1">Hello World!</p>
<script>
document.getElementById("p1").innerHTML="New text!";
var element=document.getElementById("header");
element.innerHTML="New Header";
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<script>
    document.getElementById("image").src="landscape.jpg";
</script>
</body>
</html>
```

JAVASCRIPT

- Modifier la page HTML

- ❑ Ajouter un élément

document.createElement(<elementName>)

→ Crée un nouvel élément (pas encore attaché au document)

- ❑ Ajouter une valeur

document.createTextNode(<value>)

→ Crée un nœud de texte contenant la valeur passée en paramètre

- ❑ Association d'élément

(insertion dans l'objet)

element.appendChild(<elt>)

→ Associe le nœud/élément *elt* à *element*

```

<!DOCTYPE html>
<html>
  <body>
    <div id="d1">
      <p id="p1">This is a paragraph.</p>
      <p id="p2">This is another paragraph.</p>
    </div>
    <script>
      var para=document.createElement("p");
      var node=document.createTextNode("This is new.");
      para.appendChild(node);

      var element=document.getElementById("d1");
      element.appendChild(para);
    </script>
  </body>
</html>

```

This is a paragraph.

This is another paragraph.

This is new.

HTML

JAVASCRIPT

- A vous de Jouer !
 - ❑ Créer une page permettant d'ajouter automatiquement des éléments à la page lorsque une zone est en mouseover
 - ❑ Changer le texte d'un élément permettant d'indiquer le nombre de mouseover effectué

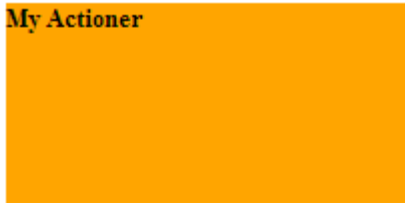


JAVASCRIPT


- A vous de Jouer !

T0

My Actioner



My Container

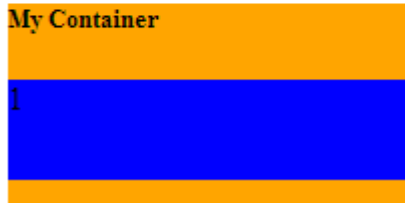


T1

1 Elements Added

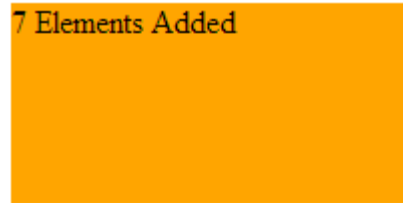


My Container

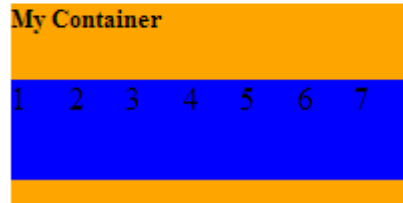


T7

7 Elements Added



My Container



Format de données JSON

JAVASCRIPT

• Format de données et d'échange Javascript

❑ JSON

❑ JavaScript Object Notation

❑ Format de description alternatif à XML

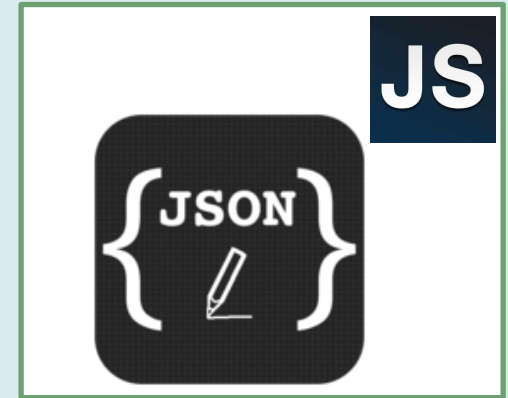
❑ Les avantages de JSON

❑ Vitesse de traitement

❑ Simplicité de mise en oeuvre

❑ Notation

```
{  
  <property1>:<value1>,  
  <property2>:<value2>,  
  <property3>:value3>,  
  <property4>:[<value41>,<value42>,<value43>,<value44>]  
  ...  
}
```



JAVASCRIPT

- **JSON: Notation**

```
{  
  id:45,  
  name: "jdoe" ,  
  age:42,  
  loginName:["jdoe" , "john.Doe" , "john_Doe"],  
  contactList:[  
    {  
      name:"smith",  
      email:"smith@company.com"  
    },  
    {  
      name:"purple",  
      email:"purple @company.com"  
    }  
  ]  
}
```

- 1 Propriétés et valeurs
- 2 Tableau de valeurs
- 3 Tableau d'objets JSON

JAVASCRIPT

- JSON utilisation

UTILISATION JSON

```
function readJson(){
  var titleToWrite='<h1>' + user.name + ' ' + user.age + '</h1>';
  var resultToWrite='<table border="1">';

  for(var i=0; i<user.contactList.length; i++){
    resultToWrite+= "<tr><td>" + user.contactList[i].name + "</td>";
    resultToWrite+= "<td>" + user.contactList[i].email + "</td></tr>";
  }

  resultToWrite+='</table>';
  document.write(titleToWrite+resultToWrite);
}
```

Objet JSON

```
user= {
  id:45,
  name: "jdoe" ,
  age:42,
  loginName:["jdoe" , "john.Doe" , "john_Doe"],
  contactList:[
    {
      name:"smith",
      email:"smith@company.com"
    },
    {
      name:"purple",
      email:"purple @company.com"
    }
  ]
}
```

JAVASCRIPT

- JSON utilisation

```

<!DOCTYPE html>
<html>
  <head>
    <title>My Very Funny Game</title>
    <script>
      var user={
        id:45,
        name: "jdoe" ,
        age:42,
        loginName:["jdoe" , "john.Doe", "john_Doe"],
        contactList:[{ name:"smith",
                       email:"smith@company.com"},
                     { name:"purple",
                       email:"purple@company.com"}]
      }

      function readJson(){
        var titleToWrite='<h1>'+user.name+' '+user.age+'</h1>';
        var resultToWrite='<table border="1">';
        for(var i=0;i<user.contactList.length;i++){
          resultToWrite+= "<tr><td>"+user.contactList[i].name+"</td>";
          resultToWrite+= "<td>"+user.contactList[i].email+"</td></tr>";
        }
        resultToWrite+='</table>';
        document.write(titleToWrite+resultToWrite);
      }
    </script>
  </head>
  <body onload="readJson()" >
  </body>
</html>

```

Résultat

jdoe 42

smith	smith@company.com
purple	purple@company.com

HTML

JAVASCRIPT

- JSON Echange

Coté Client

```
//create a JSON object to send
user={id:5,
name:document.getElementsByName("uName")[0].value,
age:document.getElementsByName("uAge")[0].value,
loginNames:loginNamesTmp,
contactList:contactListTmp
}
if (ws.readyState == 1) {
// Send the msg object as a JSON-formatted string
ws.send(JSON.stringify(user));
} else {
alert("The websocket is not open! try refreshing your
browser");
}
```

Coté Serveur

```
@Override
public void onMessage(String arg0) {
Object user = JSON.parse(arg0);
UserModel userModel=new UserModel();
System.out.println("My json Object "+user);
userModel.setValues((Map<String,Object>)user);
System.out.println("My java Object \n"+userModel);
}
```

JAVASCRIPT

• JSON Sumup

□ Notation

```
object= {  
  <property1>:<value1>,  
  <property2>:<value2>,  
  <property3>:value3,  
  <property4>:[<value41>,<value42>,<value43>,<value44>]  
  ...  
}
```

□ Utilisation

- ***object.property₁***
- ***object.property₄[2]***

□ Fonctions pour le transport

- ***JSON.stringify(object)***
- ***JSON.parse(object)***



Framework JQUERY

JAVASCRIPT

• JQUERY

- ❑ Librairie Javascript
- ❑ **Objectif:** simplification des tâches de javascript
 - ❑ Manipulation HTML/DOM
 - ❑ Manipulation CSS
 - ❑ Gestion des évènements HTML
 - ❑ Effets et animations
 - ❑ Communication AJAX
 - ❑ Compatibilité multi Navigateur

- ❑ Proposition d'une gamme de services
 - ❑ **Jquery** mobile (creation d'UI orienté mobile)
 - ❑ **JqueryUI** (librairie d'object graphique complexe)
 - ❑ **Qunitjs** (test unitaire pour javascript)



JAVASCRIPT

• JQUERY Mise en oeuvre

- ❑ Télécharger la librairie et liée la dans vos pages html

- ❑ <http://code.jquery.com/jquery-1.10.2.min.js>

- ❑ <http://code.jquery.com/jquery-1.10.2.js>

```
<!DOCTYPE html>
<html>
  <head>
    <script src="js/jquery-1.10.2.min.js"></script>
  </head>
  ...
```



- ❑ Liée votre page directement à la source de la librairie

```
<!DOCTYPE html>
<html>
  <head>
    <script
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
  </head>
  ...
```

JAVASCRIPT

• JQUERY Utilisation

Indique l'utilisation
de la librairie Jquery

Sélecteur permettant de « trouver »
les éléments (CSS sélecteur)

`$ (selector) . action ()`

Action à utiliser sur l'objet
sélectionné

```
$(this).hide();
$("p").hide();
$(".test").hide();
$("#test").hide();
```

Masque l'élément courant.
Masque **TOUS** les éléments p
Masque **TOUS** les éléments de class="test".
Masque l'élément d'id="test"

```
$("p.myClass").text("Hello world!");
$("p.myClass").html("<b>Hello world!</b>");
```

Set le texte de p de class="myClass"
Set le contenu de p de class="myClass" avec du
code HTML

JAVASCRIPT

- **JQUERY bonne pratique**

```
$(document).ready(function(){  
  
    // Méthode Jquery ici  
  
});
```

- S'assurer que le document est fini de se charger
- Exemple d'action entraînant des erreurs si le document n'est pas chargé
 - Masquer un élément qui n'est pas créé
 - Récupérer la taille d'une image qui n'est pas créée

JAVASCRIPT

• JQUERY Sélecteurs et évènements

Syntax	Description
<code>\$("*")</code>	Selects all elements
<code>\$(this)</code>	Selects the current HTML element
<code>\$(".p.intro")</code>	Selects all <p> elements with class="intro"
<code>\$(".p:first")</code>	Selects the first <p> element
<code>\$("#ul li:first")</code>	Selects the first element of the first
<code>\$("#ul li:first-child")</code>	Selects the first element of every
<code>("[href]")</code>	Selects all elements with an href attribute
<code>("[a[target='_blank']")</code>	Selects all <a> elements with a target attribute value equal to "_blank"
<code>("[a[target!='_blank']")</code>	Selects all <a> elements with a target attribute value NOT equal to "_blank"
<code>(":button")</code>	Selects all <button> elements and <input> elements of type="button"
<code>("tr:even")</code>	Selects all even <tr> elements
<code>("tr:odd")</code>	Selects all odd <tr> elements

JAVASCRIPT

- JQUERY Sélecteurs et évènements

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

```
$("#button").click(function(){  
    // mon code  
});
```

```
$("#div.myDiv").mouseenter(function(){  
    alert("HELLO! ");  
});
```

```
$("#div.myDiv").mouseleave(function(){  
    alert("GOODBYE :-(");  
});
```

JAVASCRIPT

• Utilisation des effets

❑ Afficher/cacher

❑ show

→ Affiche un élément

❑ hide

→ Masque un élément

❑ Fade

❑ FadeIn/FadeOut

→ Affichage/masquage avec effet

❑ FadeToggle

→ Alternance FadeIn/FadeOut

❑ FadeTo

→ Application d'un effet d'opacité

```
$(selector).show(speed,callback )
```

```
$(selector).hide(speed,callback )
```

```
$("#div.myEffect" ).hide();
```

```
$("#div.myEffect" ).show(1000);
```

```
$("#div.myEffect" ).hide(500,function(){  
    alert(" my div is now hidded");  
});
```

```
$(selector). fadeIn(speed,callback )
```

```
$(selector). fadeOut(speed,callback )
```

```
$(selector). fadeToggle(speed,callback )
```

```
$(selector). fadeTo (speed,opacity, callback )
```


JAVASCRIPT

• Utilisation des effets

❑ Slide

❑ slideUp

→ « Plie » un élément

❑ slideDown

→ « Déplie » un élément

❑ slideToggle

→ Alterne slideUp/SlideDown

```
$(selector).slideUp(speed,callback )
```

```
$(selector).slideDown(speed,callback )
```

```
$(selector).slideToggle(speed,callback )
```

JAVASCRIPT

• Utilisation des effets

[exemple](#)

```
$(document).ready(function(){
$("#hideShow").click(function(){
    $("#divA").toggle();
    $("#divB").toggle("slow");
    $("#divC").toggle(3000) });
$("#fadeOutOut").click(function(){
    $("#div1").fadeToggle();
    $("#div2").fadeToggle("slow");
    $("#div3").fadeToggle(3000); });
$("#slideUpDown").click(function(){
    $("#divA1").slideToggle();
    $("#divA2").slideToggle("slow");
    $("#divA3").slideToggle(3000);});
});
```

```
<body>
<div class="menu">
<p>Demonstration toggle() <br>
avec différentes vitesses.</p>
<button id="hideShow">Click to show/hide boxes</button>
<br><br>
<div id="divA" class="box color1"></div><br>
<div id="divB" class="box color2"></div><br>
<div id="divC" class="box color3"></div>
</div>
<div class="menu">
<p>Demonstration fadeToggle() avec <br>
différentes vitesses.</p>
<button id="fadeOutOut">Click to fade in/out boxes</button>
<br><br>
<div id="div1" class="box color1"></div><br>
<div id="div2" class="box color2"></div><br>
<div id="div3" class="box color3"></div></div>
...
```

JAVASCRIPT

• Utilisation des effets

❑ Animation

❑ animate

- Effectue une suite d'opération (CSS) sur l'objet

❑ enchainement d'animation

- Effectue une suite séquentielle de plusieurs animation

❑ Stop

- Arrêt de l'animation sur l'objet

exemple

```
$("#button").click(function(){
    $("#div").animate({
        left:'250px',
        opacity:'0.5',
        height:'+=150px',
        width:'+=150px'
    });
});
```

```
$("#button").click(function(){
    $("#div").animate({ left:'250px' }, "slow");
    $("#div").animate({opacity:'0.5'}, "slow");
    $("#div").animate ({height:'+=150px'}, "slow");
    $("#div").animate ({width:'+=150px'}, "slow");
});
```

```
$("#div").stop();
```

JAVASCRIPT

• Manipulation du DOM

❑ Récupérer du contenu

❑ .text()

→ Contenu texte des éléments sélectionnés

❑ .html()

→ Contenu des éléments (html)

❑ .val()

→ Valeur d'un champ d'un formulaire

❑ .attr(<attributName>)

→ Valeur d'un attribut d'un élément

```
alert("info div[menu] .text():" +  
$("div.menu").text());
```

```
alert("info div[menu] .html():" +  
$("div.menu").html());
```

```
alert("info input .val():" + $("input").val());
```

```
alert("info input name attribute  
.Attr('name'): "+$("input").attr("name"));
```

JAVASCRIPT

• Manipulation du DOM

[exemple](#)

```

<script>
$(document).ready(function(){
    $("#showDivTxt").click(function(){
        alert("info div[menu] .text(): "+$("#div.menu").text());    });

    $("#showDivHtml").click(function(){
        alert("info div[menu] .html(): "+$("#div.menu").html());    });

    $("#showFormTxt").click(function(){
        alert("info form .text(): "+$("#form").text());    });

    $("#showFormHtml").click(function(){
        alert("info form .html(): "+$("#form").html());    });

    $("#showInputValue").click(function(){
        alert("info input .val():  "+$("#input").val());});

    $("#showInputNameAttr").click(function(){
        alert("info input name attribute .Attr('name'): "+
            $("#input").attr("name"));});
});
</script>

```

```

<body>
<button id="showDivTxt">showDivTxt
</button>
<button id="showDivHtml">showDivHtml
</button>
...
<div class="menu">
    Texte Libre dans le DIV. ICI
    <p> Manipulation du <b>DOM</b> </p>
    <form>
        Nom utilisateur: <input type="text"
            name="user" value="MyDefaultValue"/>
    </form>
    LA
    </div>
</body>

```

JAVASCRIPT

• Manipulation du DOM

❑ Modification

- ❑ .text(<value>)
- ❑ .html(<html content>)
- ❑ .val(<value>)
- ❑ .attr({
 "<nomAtt1>" : "<valueAtt1>,"
 "<nomAtt1>" : "<valueAtt2> });

❑ Ajout

- ❑ .append() insertion en fin de l'élément
- ❑ .prepend() insertion en début de l'élément
- ❑ .after() insertion après l'élément
- ❑ .before() insertion avant l'élément

```
$("#btn2").click(function(){ exemple  
    $("#test2").html("<b>Hello world!</b>");  
});
```

```
$("#btn3").click(function(){  
    $("#test3").val("Dolly Duck");  
});
```

```
$("#w3s").attr({  
    "href" : "http://www.w3schools.com/jquery",  
    "title" : "W3Schools jQuery Tutorial"  
});
```

```
$("#p").append("Some appended text.");
```

```
$("#p").prepend("Some prepended text.");
```

```
$("#img").after("<p>Some text  
<b>after</b></p>");
```

```
$("#img").before("Some text before");
```

JAVASCRIPT

• Manipulation des CSS

❑ Ajout/Suppression de Class

- ❑ `.addClass("<value>")`
- ❑ `.removeClass("<value>")`
- ❑ `.toggleClass("<value>")`

❑ Modification de propriétés CSS

- ❑ `.css("<property>", "<value>")`
- ❑ `.css("<property1>":"<value1>",
"<property2>":"<value2>", ...)`

```
$("#div").addClass("myBlueClass");
```

```
$("#div").removeClass("myBlueClass");
```

```
$("#div").toggleClass("myBlueClass");
```

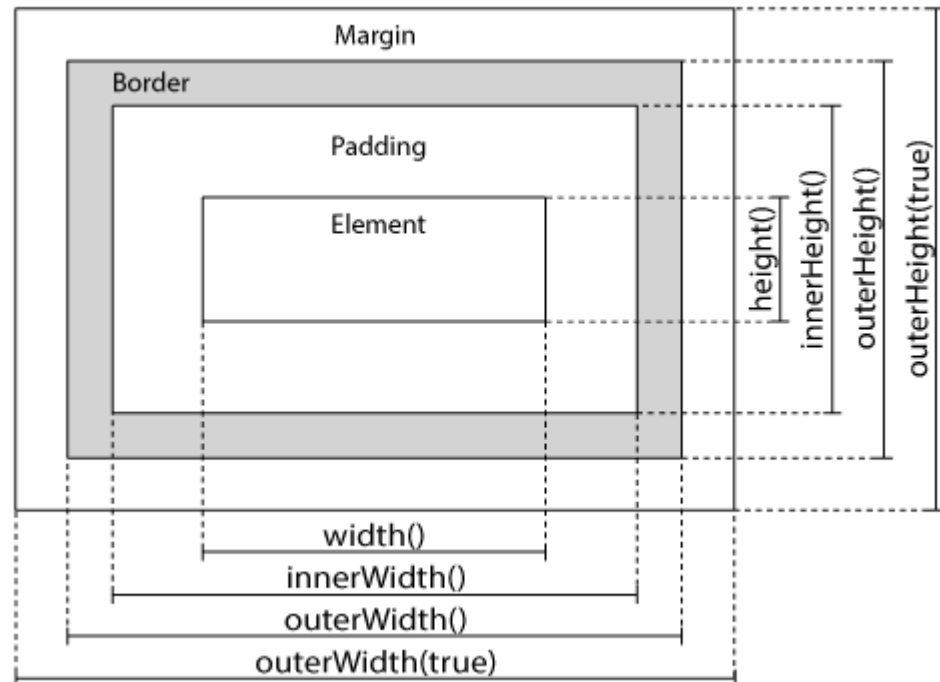
```
$("#p").css("background-color", "yellow");
```

```
$("#div").css("background-color":"yellow",  
"border-width":"5px");
```

JAVASCRIPT

- Manipulation des CSS Dimension

- `width()`
- `height()`
- `innerWidth()`
- `innerHeight()`
- `outerWidth()`
- `outerHeight()`



JAVASCRIPT

- A vous de Jouer !

- Réaliser un double menu déroulant
- 1 une bannière
- 1 click sur bannière → apparition menu
- 1click sur 1 item du menu → apparition detail du menu



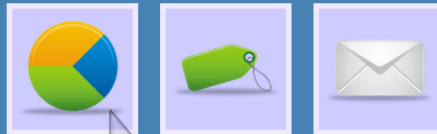
JAVASCRIPT

- A vous de Jouer !

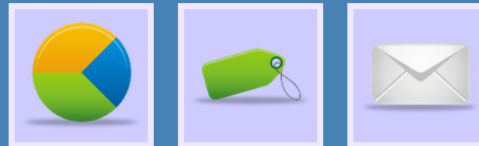
[demo](#)

MENU BOARD

MENU BOARD



MENU BOARD



Hello all the world

Les Communications

JAVASCRIPT

• Communications en Javascript

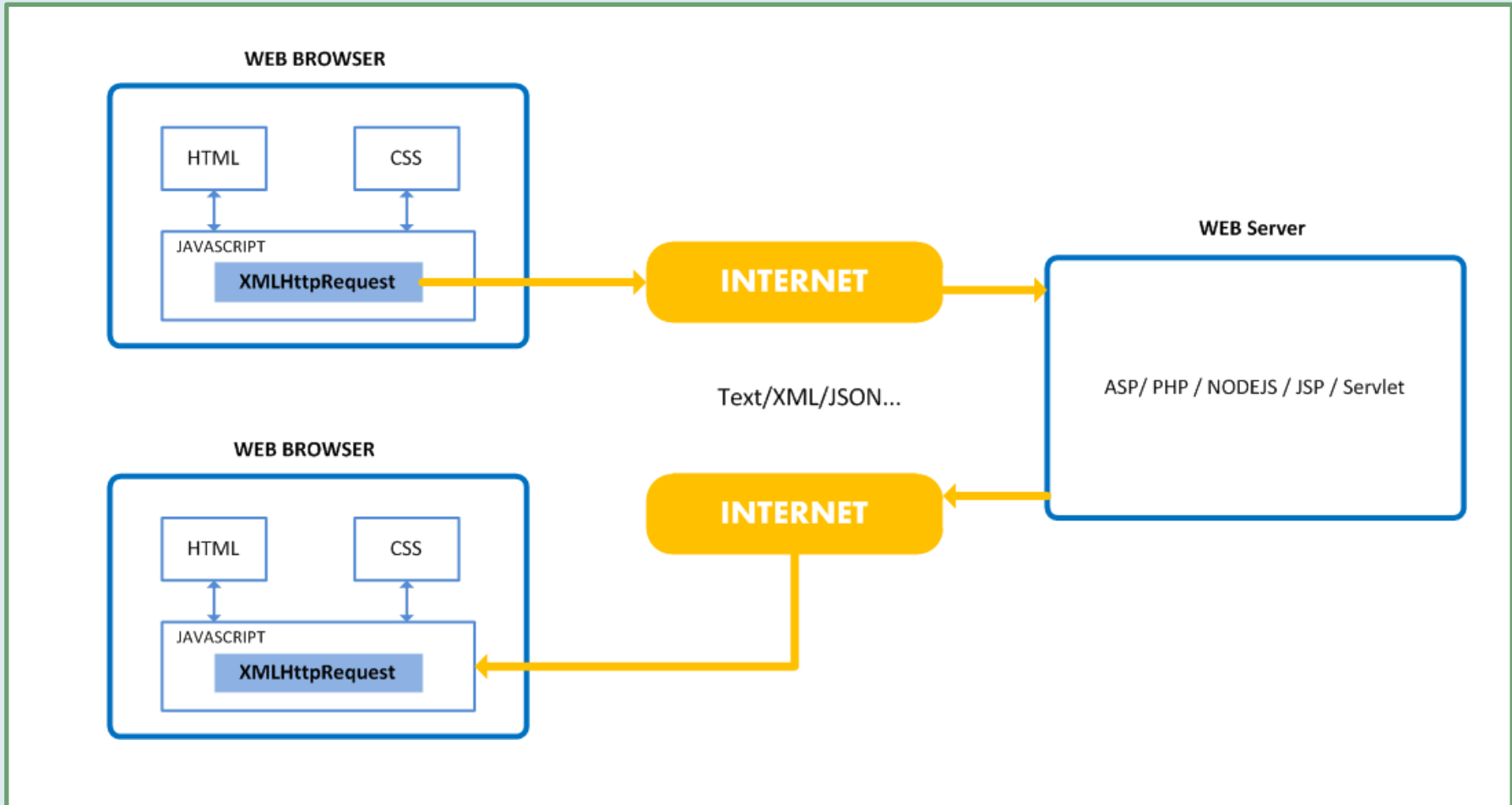
- ❑ Communications traditionnelles
 - ❑ AJAX
 - ❑ Communication unidirectionnelle
 - ❑ Utilisation de XMLHttpRequest
 - ❑ Envoi d'information protocol HTTP Get ou POST

- ❑ Nouvelles communications
 - ❑ WebSocket
 - ❑ Communications bi-directionnelles
 - ❑ Handler sur message reçu du serveur



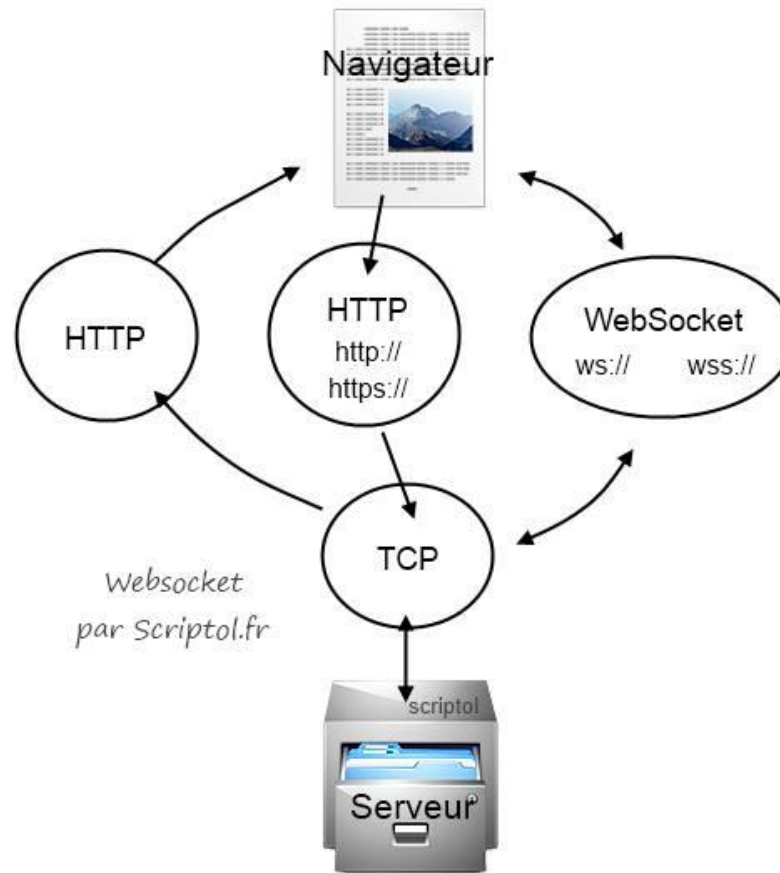
JAVASCRIPT

- **AJAX** Asynchronous Javascript And XML



JAVASCRIPT

- **Communications en Javascript**



Source: <http://www.scriptol.fr/ajax/xhr.php>

Copyright © 2012-2013 Jacques Saraydaryan

JAVASCRIPT

- **Communications en Javascript: XMLHttpRequest**

```
try {  
  xhr = new ActiveXObject("Microsoft.XMLHTTP"); // pour Internet Explorer  
}  
catch(e) // Echec  
{  
  xhr = new XMLHttpRequest(); // pour les autres navigateurs  
}
```

1 Création

//envoi d'information HTTP GET

```
xhr.open('GET', 'http://www.cpe.fr/myUrl ?param=val&param2=val2 ', true);  
xhr.send(Null);
```

2 Envoi de données

//envoi d'information HTTP POST

```
xhr.open('POST', 'http://www.cpe.fr/myUrl ', true);  
xhr.send(" param=val&param2=val2 ");
```

3 Réaction lors de la
réponse du serveur

// reaction lors d'une réponse du serveur
xhr.onreadystatechange = function() { ... };

JAVASCRIPT

- **Communications en Javascript:** XMLHttpRequest exemple

```
function submitForm1()
{
    var req = null;
    document.getElementById("dyn1").value="Started...";
    if (window.XMLHttpRequest) // test si XMLHttpRequest est supporté
    {
        req = new XMLHttpRequest();
    }
    else if (window.ActiveXObject) // test si ActiveXObject est support (IE)
    {
        try { req = new ActiveXObject("Msxml2.XMLHTTP");
        } catch (e)
        {
            try {
                req = new ActiveXObject("Microsoft.XMLHTTP");
            } catch (e) {}
        }
    }
    ....
}
```


- **Communications en Javascript: XMLHttpRequest exemple**

```

req.onreadystatechange = function()
    {
        document.getElementById("dyn1").value="Wait server...";
        if(req.readyState == 4) // réponse reçue
        {
            if(req.status == 200) //destination trouvée
            { document.getElementById("dyn1").value=req.responseText; }

            else
            { document.getElementById("dyn1").value="Error: returned status code "
                + req.status + " " + req.statusText;
            }
        }
    };
req.open("GET", "demo.txt", true);
req.send(null);

```

JAVASCRIPT

- **Communications en Javascript: WebSocket**

```
var connection = new WebSocket('http://www.cpe.fr/myUrl ');
```

1 Création

```
connection.onopen=function()  
{ // your code executed after connection };
```

2 Exécution à l'ouverture

```
connection.onerror =function(e)  
{ // your code executed on errors};
```

3 Exécution en cas d'erreurs

```
connection.send('Message xxx');
```

4 Envoi de données

```
// Données reçu contenu dans e.data  
connection.onmessage=function(e)  
{  
  console.log('Received: ' + e.data);  
};
```

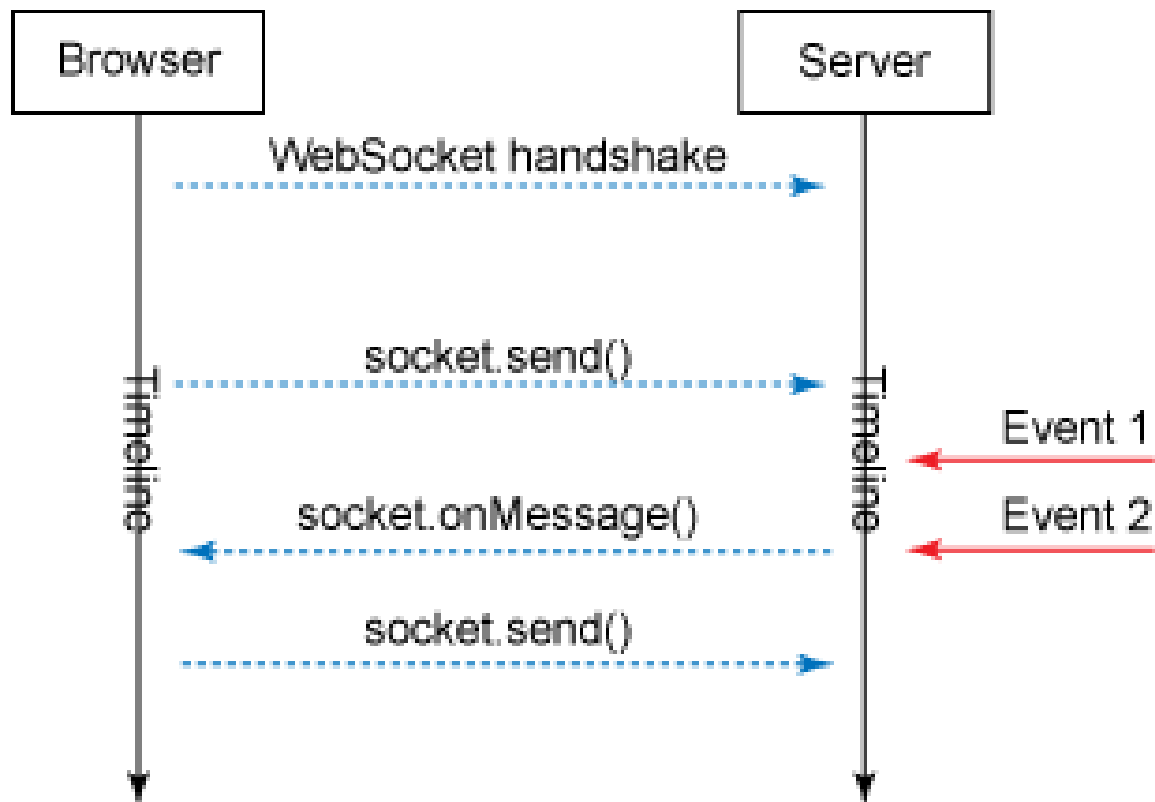
5 Réaction lors de la réponse du serveur

```
connection.onclose=function(e)  
{ // your code executed on close event};
```

6 Exécution à la fermeture

JAVASCRIPT

- **Communications en Javascript: WebSocket**



JAVASCRIPT

- **Communications en Javascript: WebSocket Exemple**

```
var ws; var username; var msg;

function updateStatus(id, message) {
    document.getElementById(id).innerHTML = message;
}

function loadWebSocket() {
    if (window.WebSocket) { // test si le navigateur supporte les web socket
        var url = "ws://localhost:8080/Html5WebSocket/ExempleWebSocket/";
        ws = new WebSocket(url); //creation de la websocket
        updateStatus("wsStatus", "webSocket supported !");
        ws.onopen = function() { // execution à l'ouverture
            updateStatus("wsStatus", "Connected to WebSocket server!");
        }
        ws.onmessage = function(e) { // Reception de messages
            displayMessage(e.data);
        }
        ws.onclose = function() { //execution à la fermeture
            updateStatus("wsStatus", "WebSocket closed!");
        }
        ws.onerror = function(e) { //execution en case d'erreur
            updateStatus("wsStatus", "WebSocket error : " + e.data);
        }
    } else { updateStatus("wsStatus", "Your browser does NOT support webSocket.");
    }
}
```

- **Communications en Javascript: WebSocket Exemple**

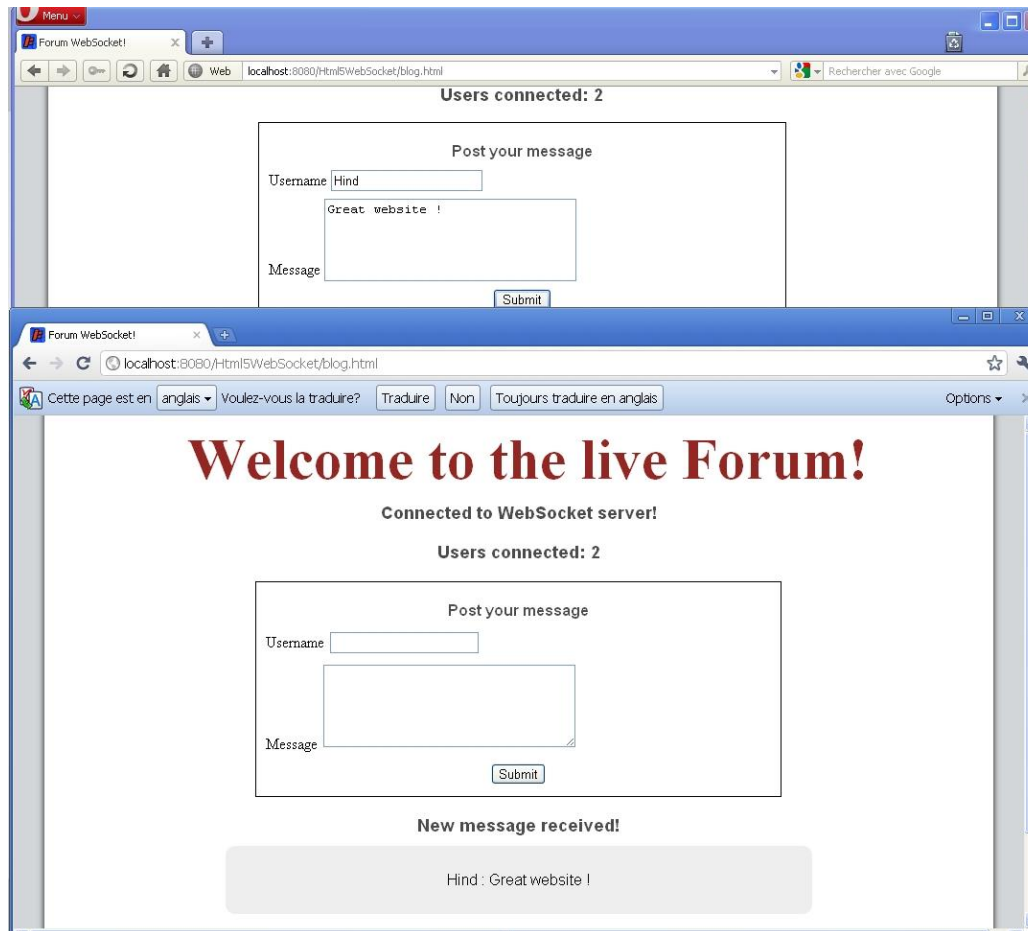
```

//fonction qui envoie le texte saisi et validé par l'utilisateur.
function sendMyPost(newPost) {
    username = document.forms["myform"].name.value;
    msg = document.forms["myform"].msg.value;
    if (msg) {
        var post = username + " : " + msg;
        if (ws.readyState == 1) {
            ws.send(post); //envoi du message
        } else {
            alert("The websocket is not open! try refreshing your browser");
        }
    }
}

function displayMessage(message) {
    //afficher le message en dessous du formulaire
    //...
}
    
```

JAVASCRIPT

- **Communications en Javascript: WebSocket Exemple**



Source: <http://blog.zenika.com/index.php?post/2011/02/25/Html5-et-les-webSockets>

JAVASCRIPT

• Communications en Javascript : Web Socket

❑ Compatibilité:

- Chrome : supporté version 4+
- Firefox : supporté mais désactivé version 4+
- IE : non supporté
- Opéra : supporté mais désactivé version 11+
- Safari : supporté version 5+

❑ Librairie Coté serveur

- Kaazing WebSocket Gateway
- Socket.IO-Node (NodeJS)
- Jetty (Java)
- Netty (Framework Java client serveur)
- JWebSocket (Java)
- Web Socket Ruby (Ruby)
- mod_pyWebSocket (extension en Python pour le serveur Apache HTTP)
- Websocket (Python)



JAVASCRIPT

- A vous de Jouer !
 - ❑ Créer un chat dynamique en vous servant des exemples présentés précédemment.
 - ❑ Sauvegarder sur le client les messages échangés afin de les retrouver lors d'une prochaine connexion.



USING JETTY IMPLEMENTATION

```
/**
 * Jetty WebSocketServlet implementation class ServerChatWebSocket
 */
@WebServlet("/ServerChatWebSocket")
public class ServerChatWebSocket extends WebSocketServlet {
    private static final long serialVersionUID = 1L;
    public final Set<ChatWebSocket> users = new CopyOnWriteArraySet<ChatWebSocket>();

    protected void doGet(HttpServletRequest request,
                          HttpServletResponse response) throws ServletException, IOException {
        getServletContext().getNamedDispatcher("default").forward(request,
                                                                    response);
    }

    @Override
    public WebSocket doWebSocketConnect(HttpServletRequest arg0, String arg1) {
        return new ChatWebSocket(users);
    }
}
```

```
import org.eclipse.jetty.websocket.WebSocket.OnTextMessage;
public class ChatWebSocket implements OnTextMessage {

    private Connection connection;
    private Set<ChatWebSocket> users;
    public ChatWebSocket() {
    }
    public ChatWebSocket(Set<ChatWebSocket> users ) {
        this.users = users;
    }
    @Override
    public void onMessage(String data) {
        for (ChatWebSocket user : users) {
            try {
                user.connection.sendMessage(data);
            } catch (Exception e) {
            }
        }
    }
    @Override
    public void onOpen(Connection connection) {
        this.connection = connection;
        users.add(this);
    }
    @Override
    public void onClose(int closeCode, String message) {
        users.remove(this);
    }
}
```

USING APACHE 7 IMPLEMENTATION

```
import javax.servlet.http.HttpServletRequest;
import org.apache.catalina.websocket.StreamInbound;
import org.apache.catalina.websocket.WebSocketServlet;
import org.apache.catalina.websocket.WsOutbound;
import communication.ChatMessageInbound;

public class ChatWebSocketServlet extends WebSocketServlet{
    private ArrayList<WsOutbound> connections;

    @Override
    protected StreamInbound createWebSocketInbound(String arg0,
        HttpServletRequest arg1) {
        return new ChatMessageInbound(getConnection());
    }

    private ArrayList<WsOutbound> getConnection() {
        if(this.connections==null){
            this.connections=new ArrayList<>();
        }
        return this.connections;
    }
}
```

```
public class ChatMessageInbound extends MessageInbound {
    private ArrayList<WsOutbound> connections;
    private WsOutbound outbound;

    public ChatMessageInbound(ArrayList<WsOutbound> connections) {
        this.connections = connections;
    }
    @Override
    protected void onBinaryMessage(ByteBuffer arg0) throws IOException {}
    @Override
    protected void onTextMessage(CharBuffer message) throws IOException {
        String msg = message.toString();
        broadcast(msg);
    }
    private void broadcast(String message) {
        try {
            for (WsOutbound out : this.connections) {
                CharBuffer buffer = CharBuffer.wrap(message);
                out.writeTextMessage(buffer);
            } catch (IOException e) { e.printStackTrace();}
        }
    }

    @Override
    protected void onOpen(WsOutbound outbound) {
        this.outbound = outbound;
        this.connections.add(outbound);
    }
    @Override
    protected void onClose(int status) { this.connections.remove(this); }
}
```

JAVASCRIPT

• Communications Avec JQuery

- ❑ Chargement d'information
 - ❑ **load** récupération d'information depuis une ressource (1 fichier)
 - ❑ **Get**: requête HTTP Get sur une URL spécifiée
 - ❑ **Post**: requête HTTP POST sur une URL spécifiée
 - ❑ **Websocket** ? Des implémentations existent
 - ❑ jquery-websocket
 - ❑ jquery-graceful-websocket
- Pas vraiment plus simple qu'en javascript



JAVASCRIPT

• Communications Avec JQuery

❑ load

- Insérer les éléments contenus dans l'URL dans les objets pointés par le selecteur

```
$(selector).load(URL,parameters,callback);
```

❑ Get

- ❑ Appel d'un contenu distant en HTTP GET

```
$.get(URL,callback);
```

❑ Post

- ❑ Appel d'un contenu distant en HTTP Post, données passées en paramètre au format JSON

```
$.post(URL,data,callback);
```

```
$("#volatileContent").load("content/item1Content.txt");
```

item1Content.txt

```
<div class="myClass">  
  <p>Hello <b>World</b><p>  
</div>
```

```
$.get("http://localhost/index.jsp",function(data,status){  
  alert("Data: " + data + "\nStatus: " + status);  
});
```

```
$.post("http://localhost/submit.jsp",  
{  
  name:"Donald Duck",  
  city:"Duckburg"  
},  
function(data,status){  
  alert("Data: " + data + "\nStatus: " + status);  
});
```

QUESTIONS ?
