



**3.4.3.1 - Définition XML**

**3.4.3.2 - Définition XSLT**

**3.4.3.3 - XML : Exemple**

**3.4.3.4 - XSLT : Exemple**

**3.4.3.5 - Élément XSLT**



## ➤ XML (*Extensible Markup Language*)

Langage informatique de balisage **générique**. Le W3C recommande XML pour exprimer des langages de balisages **spécifiques** (exemples : XHTML, SVG, XSLT).

XML est un sous-ensemble de SGML dont il retient plusieurs principes dont : la structure d'un document XML est définissable et validable par un schéma.

**Objectif** initial est de faciliter l'**échange** automatisé de contenus entre **systèmes d'informations** hétérogènes, notamment, sur Internet.



### 3.4.3.2 - Définition XSLT



## ➤ XSLT (eXtended Stylesheet Language Transformations)

XSLT est un langage pour transformer des documents XML en d'autres documents XML. Développé par le W3C, initialement pour le langage de formatage XSL

- Recommandation W3C depuis novembre 1999
- XSLT 2.0 en cours de développement
- Utilise la syntaxe XML
- Utilise le langage d'expressions XPath
- Même type de document ou types différents en entrée et en sortie
- XSLT peut être utilisé dans une chaîne de traitement qui utilise aussi Java, Javascript, ASP, PHP, etc.



## 3.4.3.3a - XML : Exemple



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE tagxhtml SYSTEM "../dtd/tag_xhtml.dtd">

<tagxhtml>
  <tag>
    <balise> <![CDATA[<a>]]> </balise>
    <definition>Lien hypertexte</definition>
    <exemple> <![CDATA[
<p> lien vers W3Schools: <a href="http://www.w3schools.com"> W3Schools</a> </p>
]]>
    </exemple>
  </tag>
  <tag>
    <balise> <![CDATA[<acronyme>]]> </balise>
    <definition>Acronyme</definition>
    <exemple><![CDATA[]]></exemple>
  </tag>
</tagxhtml>
```



### 3.4.3.3b - XML : Exemple



```
<!ELEMENT tagxhtml (tag+)>  
<!ELEMENT tag (balise,definition,exemple)>  
<!ELEMENT balise (#PCDATA)>  
<!ELEMENT definition (#PCDATA)>  
<!ELEMENT exemple (#PCDATA)>
```

Ce fichier XML ne semble pas avoir d'information de style lui étant associé. L'arbre du document est montré ci-dessous.

```
- <tag_xhtml>
  - <tag>
    <balise> <a> </balise>
    <definition>Lien hypertexte</definition>
    - <exemple>
      <p> lien vers W3Schools: <a href="http://www.w3schools.com"> W3Schools</a> </p>
    </exemple>
  </tag>
  - <tag>
    <balise> <acronyme> </balise>
    <definition>Acronyme</definition>
    <exemple></exemple>
  </tag>
</tag_xhtml>
```



## 3.4.3.4a - XSLT : Exemple



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="../xsl/tag.xsl"?>
<!DOCTYPE tagxhtml SYSTEM "../dtd/tag_xhtml.dtd">

<tagxhtml>
  <tag>
    <balise> <![CDATA[<a>]]> </balise>
    <definition>Lien hypertexte</definition>
    <exemple> <![CDATA[
<p> lien vers W3Schools: <a href="http://www.w3schools.com"> W3Schools</a> </p>
]]>
    </exemple>
  </tag>
  <tag>
    <balise> <![CDATA[<acronyme>]]> </balise>
    <definition>Acronyme</definition>
    <exemple><![CDATA[]]></exemple>
  </tag>
</tagxhtml>
```



## 3.4.3.4b - XSLT : Exemple



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
  <html>
    <head>
      <title> Cours Web 2.0 </title>
      <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
      <link rel="stylesheet" media="screen" type="text/css" title="ExempleCSS" href=" ../css/design.css" />
    </head>
    <body id="CorpsPages">
      <h1 id="Titre1" >Cours Web 2.0 </h1>
      <h2 id="Titre2" >Les balises XHTML</h2>
      <ol>
        <li>Balises de premier niveau et en-tête </li><li>Balises de structuration du texte
          <table class="Tableau" border="1">
            <caption>Tableau des balises de structuration du texte</caption>
            <thead><tr><th>Balise</th><th>Definition</th><th>Exemple</th></tr></thead>
            <tfoot> <tr> <th colspan="3"><small><a href="http://www.w3schools.com">W3Schools </a></small></th></tr></tfoot>
            <tbody> <xsl:for-each select="//tag"> <tr>
              <th><code> <xsl:value-of select="balise"/> </code></th>
              <th><xsl:value-of select="definition"/> </th>
              <th><samp> <xsl:value-of select="exemple"/> </samp></th>
            </tr> </xsl:for-each> </tbody>
          </table>
        </li><li>Balises de liste </li></ol></body>
      </html>
    </xsl:template>
  </xsl:stylesheet>
```



### Cours Web 2.0

#### Les balises XHTML

- i. Balises de premier niveau et en-tête
- ii. Balises de structuration du texte

Tableau des balises de structuration du texte

Balise	Définition	Exemple
<code>&lt;a&gt;</code>	Lien hypertexte	<code>&lt;p&gt; lien vers W3Schools: &lt;a href="http://www.w3schools.com"&gt; W3Schools&lt;/a&gt; &lt;/p&gt;</code>
<code>&lt;acronym&gt;</code>	Acronyme	<code>&lt;acronym&gt;W3Schools&lt;/acronym&gt;</code>

- iii. Balises de liste



## 3.4.3.5 - Élément XSLT



- Élément template:
  - <xsl:apply-template>
  - <xsl:template>
  - <xsl:call-template>
  
- Element définition de variable
  - <xsl:param>
  - <xsl:with-param>
  - <xsl:variable>
  
- Élément de structure:
  - <xsl:for-each>
  - <xsl:value-of>
  - <xsl:sort>
  - <xsl:if>
  - <xsl:choose>
  - <xsl:when>

Allez voir :

[http://www.w3schools.com/xsl/xsl\\_w3celementref.asp](http://www.w3schools.com/xsl/xsl_w3celementref.asp)



## 3.4.4 - DOM/SAX

### ➤ **DOM** (*Document Object Model*)

spécification du W3C (*World Wide Web Consortium*) qui consiste à décomposer le contenu d'un document HTML ou XML en une arborescence de nœuds (chaque éléments du document est un nœud).

### ➤ **SAX** (*Simple API for XML*)

Interface de programmation pour de nombreux langages permettant de lire et de traiter des documents XML.

C'est une API basée sur un modèle événementiel, cela signifie que SAX permet de déclencher des événements au cours de l'analyse du document XML.

### ➤ **Différences DOM/SAX**

DOM charge l'intégralité d'un document XML dans une structure de données, qui peut alors être manipulée puis reconvertie en XML.

SAX traite les documents élément par élément au fur et à mesure qu'ils sont rencontrés



## 3.4.5 - JavaScript



### 3.4.5.1 - Définition

### 3.4.5.2 - L 'élément HTML <Script>

### 3.4.5.3 - document.write(...)

### 3.4.5.4 - Boucle while ( ... ) { ... }

### 3.4.5.5 - Boucle do { ... } while( ... )

### 3.4.5.6 - Boucle for(... ; ... ; ... ) { ... }

### 3.4.5.7 - Boucle for ( ... in ... ) { ... }

### 3.4.5.8 - Tests switch ( ... ) { case : ...; break; default: ...}

### 3.4.5.9 - Tests if (...) { ... } else { ...}

### 3.4.5.10 - try { ...} ; catch (...) { ... }

### 3.4.5.11 - Les 4 Types de bases

### 3.4.5.12 - Déclaration de variables

### 3.4.5.13 - Les Expressions

### 3.4.5.14 - Les Opérateurs

### 3.4.5.15 - Les Fonctions

### 3.4.5.16 - Les Objets

### 3.4.5.17 - Les Objets prédéfinis

### 3.4.5.18 - Les Objets prédéfinis du navigateur

### 3.4.5.19 - Bibliographie



### 3.4.5.1 - Définition



Javascript est un langage interprété par le navigateur.

Le Javascript est un langage « client »  $\Rightarrow$  il est exécuté sur la machine de l'utilisateur lorsque la page Web est chargée.

Son but est de dynamiser les sites Internet

**Javascript n'est pas java**



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" >
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
  <head>
    <title> Cours Web 2.0 : JavaScript</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <link rel="stylesheet" media="screen" type="text/css" title="ExempleCSS" href="css/design.css" />
    <script type="text/javascript" src="js/jstest.js" >
    </script>
  </head>

  <body id="CorpsPages">
    <script type="text/javascript">
      doctitre1="Cours Web 2.0";
      doctitre2="JavaScript";
      doctitre3="document.write(...)";
      document.write("<h1 id=\"Titre1\">"+doctitre1+"</h1>");
      document.write("<h2 id=\"Titre2\">"+doctitre2+"</h2>");
      document.write("<h3 id=\"Titre3\">"+doctitre3+"</h3>");
    </script>
  </body>
</html>
```



### 3.4.5.3 - document.write(...)



```
<script type="text/javascript">
```

```
/* Variables globales */
```

```
doctitre1="Cours Web 2.0";
```

```
doctitre2="JavaScript";
```

```
doctitre3="document.write(...)";
```

```
/* Write méthode de la classe document permettant d'écrire dans le document */
```

```
document.write("<h1 id=\"Titre1\">"+doctitre1+"</h1>");
```

```
document.write("<h2 id=\"Titre2\">"+doctitre2+"</h2>");
```

```
document.write("<h3 id=\"Titre3\">"+doctitre3+"</h3>");
```

```
</script>
```



## 3.4.5.3a - document.write(...)





```
<script type="text/javascript">
  /* Variables globales */
  doctitre1="Cours Web 2.0";
  doctitre2="JavaScript";
  /* Variables locales */
  var tab= new Array();
  var i=0;
  tab[0] = "document.write(...)";
  tab[1] = "boucle while {}";
  /* Write méthode de la classe document permettant d'écrire dans le document*/
  document.write("<h1 id=\"Titre1\">"+doctitre1+"</h1>");
  document.write("<h2 id=\"Titre2\">"+doctitre2+"</h2>");
  document.write("<table border=\"1\">");
  while (i < 2)
  {
    i++;
    document.write("<tr><th>"+i+"</th><th>"+tab[i-1]+"</th></tr>");
  }
  document.write("</table>");
</script>
```



## 3.4.5.4a - Boucle while ( ... ) { ... }

**Cours Web 2.0**

**JavaScript**

1	document.write(...)	
2	boucle while	

```
<script type="text/javascript">
  /* Variables globales */
  doctitre1="Cours Web 2.0";
  doctitre2="JavaScript";
  /* Variables locales */
  var tab= new Array();
  var i=0;
  tab[0] = "document.write(...)";
  tab[1] = "boucle while {}";
  tab[2] = "boucle do {} while {}";
  /* Write methode de la classe document permettant d'ecrire dans le document*/
  document.write("<h1 id=\"Titre1\">"+doctitre1+"</h1>");
  document.write("<h2 id=\"Titre2\">"+doctitre2+"</h2>");
  document.write("<table border=\"1\">");
  do {
    i++;
    document.write("<tr><th>"+i+"</th><th>"+tab[i-1]+"</th></tr>");
  } while (i < 3)
  document.write("</table>");
</script>
```



## 3.4.5.5a - Boucle do { ... } while( ... )

**Cours Web 2.0**

**JavaScript**

1	<code>document.write(...)</code>	
2	boucle while	





## 3.4.5.6 - Boucle for(... ; ... ; ... ) { ... }



```
<script type="text/javascript">
  /* Variables globales */
  doctitre1="Cours Web 2.0";
  doctitre2="JavaScript";
  /* Variables locales */
  var tab= new Array();
  var i=0;
  tab[0] = "document.write(...)";
  tab[1] = "boucle while {}";
  tab[2] = "boucle do {} while {}";
  tab[3] = "for(... ; ... ; ... ) {}";
  /* Write methode de la classe document permettant d'ecrire dans le document*/
  document.write("<h1 id=\"Titre1\">"+doctitre1+"</h1>");
  document.write("<h2 id=\"Titre2\">"+doctitre2+"</h2>");
  document.write("<table border=\"1\">");
  for (i=1;i <5;i++) {
    document.write("<tr><th>"+i+"</th><th>"+tab[i-1]+"</th></tr>");
  }
  document.write("</table>");
</script>
</body>
</html>
```

**Cours Web 2.0**

**JavaScript**

1	document.write(...)	
2	boucle while {}	
3	boucle do {} while {}	
4	for(... ; ... ; ... ) {}	

```
<script type="text/javascript">
  /* Variables globales */
  doctitre1="Cours Web 2.0";
  doctitre2="JavaScript";
  /* Variables locales */
  var prop;
  /* Write methode de la classe document permettant d'ecrire dans le document*/
  document.write("<h1 id=\"Titre1\">"+doctitre1+"</h1>");
  document.write("<h2 id=\"Titre2\">"+doctitre2+"</h2>");
  document.write("<h3 id=\"Titre3\"> Propriété de l'objet windows</h2>");
  for (prop in window) {
    document.write("<b style=\"color:white\">" + prop + " = " + "</b>");
    document.write("<i>" + window[prop] + " </i>");
  }
  document.write("<h3 id=\"Titre3\"> Propriété de l'objet document</h2>");
  for (var prop1 in document) {
    document.write("<b style=\"color:white\">" + prop1 + " = </b>");
    document.write("<i>" + document[prop1] + " </i>");
  }
</script>
```



## 3.4.5.7a - Boucle for (... in ... ) { ... }

### Cours Web 2.0

#### JavaScript

##### Propriété de l'objet windows

```
getInterface = function getInterface() { [native code] } doctitle2 = JavaScript doctitle1 = Cours Web 2.0 prop1 = undefined prop = prop _firebug_ = [object Object] window =
[object Window] console = [object Object] addEventListener = function addEventListener() { [native code] } location =
file:///C:/Documents%20and%20Settings/David/Bureau/Miages-Exemple/Figure16.html document = [object HTMLDocument] navigator = [object Navigator] packages =
[JavaPackage] sun = [JavaPackage sun] java = [JavaPackage java] xpcNativeWrapper = function xpcNativeWrapper() { [native code] }
GeckoActiveXObject = function GeckoActiveXObject() { [native code] } Components = [object nsXPCComponents] parent = [object Window] op = [object Window] scrollbar =
[object BarProp] name = scrollX = 0 scrollY = 0 scrollTo = function scrollTo() { [native code] } scrollBy = function scrollBy() { [native code] } getSelection = function getSelection() {
[native code] } scrollByLines = function scrollByLines() { [native code] } scrollByPages = function scrollByPages() { [native code] } moveToElement = function moveToElement() {
[native code] } setInterval = function setInterval() { [native code] } setTimeout = function setTimeout() { [native code] } clearInterval = function clearInterval() { [native code] }
clearTimeout = function clearTimeout() { [native code] } clearInterval = function clearInterval() { [native code] } setResizable = function setResizable() {
[native code] } captureEvents = function captureEvents() { [native code] } releaseEvents = function releaseEvents() { [native code] } routeEvent = function routeEvent() { [native code] }
enableExternalCapture = function enableExternalCapture() { [native code] } disableExternalCapture = function disableExternalCapture() { [native code] } open = function open() {
[native code] } openDialog = function openDialog() { [native code] } frames = [object Window] find = function find() { [native code] } self = [object Window] screen = [object Screen]
history = [object History] content = [object Window] toolbar = [object BarProp] locationbar = [object BarProp] personalbar = [object BarProp]
statusbar = [object BarProp] directories = [object BarProp] closed = false crypto = [object Crypto] pkcs11 = [object Pkcs11] controllers = [object XMLHttpRequest] opener = null
status = defaultStatus = innerWidth = 316 innerHeight = 385 outerWidth = 927 outerHeight = 850 screenX = 343 screenY = -20 pageXOffset = 0 pageYOffset = 0
scrollMaxX = 0 scrollMaxY = 462 length = 6 fullScreen = false alert = function alert() { [native code] } confirm = function confirm() { [native code] } focus = function focus() { [native
code] } blur = function blur() { [native code] } back = function back() { [native code] } forward = function forward() { [native code] } home = function home() { [native code] } stop =
function stop() { [native code] } print = function print() { [native code] } moveTo = function moveTo() { [native code] } moveBy = function moveBy() { [native code] } resizeTo = function
resizeTo() { [native code] } resizeBy = function resizeBy() { [native code] } scroll = function scroll() { [native code] } close = function close() { [native code] } updateCommands =
function updateCommands() { [native code] } atob = function atob() { [native code] } btoa = function btoa() { [native code] } frameElement = null removeEventListener = function
removeEventListener() { [native code] } dispatchEvent = function dispatchEvent() { [native code] } getComputedStyle = function getComputedStyle() { [native code] }
sessionStorage = null globalStorage = [object Storage]

```

##### Propriété de l'objet document

```
xmlVersion = null createElement = function createElement() { [native code] } createTextNode = function createTextNode() { [native code] } getElementsByTagName = function
getElementsByTagName() { [native code] } write = function write() { [native code] } title = Cours Web 2.0 javaScript referrer = styleSheets = [object StyleSheetList] baseURI =
file:///C:/Documents%20and%20Settings/David/Bureau/Miages-Exemple/Figure16.html compareDocumentPosition = function compareDocumentPosition() { [native code] }
textContent = null isSameNode = function isSameNode() { [native code] } lookupPrefix = function lookupPrefix() { [native code] } isDefaultNamespace = function
isDefaultNamespace() { [native code] } lookupNamespaceURI = function lookupNamespaceURI() { [native code] } isEqualNode = function isEqualNode() { [native code] } getFeature =
function getFeature() { [native code] } setData = function setData() { [native code] } getUserData = function getUserData() { [native code] }
DOCUMENT_POSITION_DISCONNECTED = 1 DOCUMENT_POSITION_PRECEDING = 2 DOCUMENT_POSITION_FOLLOWING = 4 DOCUMENT_POSITION_CONTAINS = 8
DOCUMENT_POSITION_CONTAINED_BY = 16 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC = 32 nodeName = #document nodeValue = null.nodeType = 9 parentNode =
null.childNodes = [object NodeList] firstChild = [object DocumentType] lastChild = [object HTMLHtmlElement] previousSibling = null nextSibling = null attributes = null
ownerDocument = null insertBefore = function insertBefore() { [native code] } replaceChild = function replaceChild() { [native code] } removeChild = function removeChild() {
[native code] } appendChild = function appendChild() { [native code] } hasChildNodes = function hasChildNodes() { [native code] } cloneNode = function cloneNode() { [native code] }
normalize = function normalize() { [native code] } isSupported = function isSupported() { [native code] } namespaceURI = null prefix = null localName = null hasAttributes =
function hasAttributes() { [native code] } doctype = [object DocumentType] implementation = [object DOMImplementation] documentElement = [object HTMLHtmlElement]
createDocumentFragment = function createDocumentFragment() { [native code] } createComment = function createComment() { [native code] } createCDATASection = function
createCDATASection() { [native code] } createProcessingInstruction = function createProcessingInstruction() { [native code] } createAttribute = function createAttribute() { [native
code] } createEntityReference = function createEntityReference() { [native code] } importNode = function importNode() { [native code] } createElementNS = function
createElementNS() { [native code] } createAttributeNS = function createAttributeNS() { [native code] } getElementsByTagNameNS = function getElementsByTagNameNS() { [native
code] } getElementById = function getElementById() { [native code] } ELEMENT_NODE = 1 ATTRIBUTE_NODE = 2 TEXT_NODE = 3 CDATA_SECTION_NODE = 4
ENTITY_REFERENCE_NODE = 5 ENTITY_NODE = 6 PROCESSING_INSTRUCTION_NODE = 7 COMMENT_NODE = 8 DOCUMENT_NODE = 9 DOCUMENT_TYPE_NODE = 10
DOCUMENT_FRAGMENT_NODE = 11 NOTATION_NODE = 12 URL = file:///C:/Documents%20and%20Settings/David/Bureau/Miages-Exemple/Figure16.html body = [object
HTMLBodyElement] images = [object HTMLCollection] applets = [object HTMLCollection] links = [object HTMLCollection] forms = [object HTMLCollection] anchors = [object
HTMLCollection] cookie = close = function close() { [native code] } getElementsByTagName = function getElementsByTagName() { [native code] } width = 597 height = 2473 linkColor =
#0000ff linkColor = #0000ff vlinkColor = #0000ff bgColor = #0000ff fgColor = #000000 domain = embeds = [object HTMLCollection] getSelection = function getSelection() {
[native code] } writeln = function writeln() { [native code] } clear = function clear() { [native code] } captureEvents = function captureEvents() { [native code] } releaseEvents =
function releaseEvents() { [native code] } routeEvent = function routeEvent() { [native code] } compatMode = CSS1Compat plugins = [object HTMLCollection] designMode = off
execCommand = function execCommand() { [native code] } execCommandShowHelp = function execCommandShowHelp() { [native code] } execCommandEnabled = function

```





### 3.4.5.8 - Tests

switch ( ... ) { case : ...; break; default: ... }



```
<script type="text/javascript">
/* Variables globales */
doctitre1="Cours Web 2.0";
doctitre2="JavaScript";
/* Variables locales */
var tab= new Array();
var i=0;
tab[0] = "document.write(...)";
tab[1] = "boucle while {}";
tab[2] = "boucle do {} while {}";
tab[3] = "for(... ; ... ; ... ) {}";
tab[4] = "switch () { case : ...; break; default: ...}";
/* Write methode de la classe document permettant d'ecrire dans le document*/
document.write("<h1 id=\"Titre1\">"+doctitre1+"</h1>");
document.write("<h2 id=\"Titre2\">"+doctitre2+"</h2>");
document.write("<table border=\"1\">");
for (i=1;i <6;i++) {
    document.write("<tr><th>"+i+"</th><th>"+tab[i-1]+"</th>");
    switch (i){
        case 1: document.write("<th> Exemple 1 </th><tr>");break;
        case 2: document.write("<th> Exemple 2 </th><tr>");break;
        case 3: document.write("<th> Exemple 3 </th><tr>");break;
        default: document.write("<th> Exemple par default </th><tr>"); break;}
    }
    document.write("</table>");
</script>
```



## 3.4.5.8a - Tests

switch ( ... ) { case : ...; break; default: ... }



### Cours Web 2.0

#### JavaScript

1	document.write(...)	Exemple 1
2	boucle while {}	Exemple 2
3	boucle do {} while {}	Exemple 3
4	for(...; ...; ...) {}	Exemple par défaut
5	switch {} { case : ...; break; default: ... }	Exemple par défaut



## 3.4.5.9 - Tests if (...) { ... } else { ... }

```
<script type="text/javascript">
  /* Variables globales */
  doctitre1="Cours Web 2.0";
  doctitre2="JavaScript";
  /* Variables locales */
  var tab= new Array(), i=0, continuer = true;
  tab[0] = "document.write(...)";
  tab[1] = "boucle while {}";
  tab[2] = "boucle do {} while {}";
  tab[3] = "for(... ; ... ; ... ) {}";
  tab[4] = "switch () { case : ...; break; default: ...}";
  tab[5] = "if (...) {} else {}";
  /* Write methode de la classe document permettant d'ecrire dans le document*/
  document.write("<h1 id=\"Titre1\">"+doctitre1+"</h1>");
  document.write("<h2 id=\"Titre2\">"+doctitre2+"</h2>");
  document.write("<h3 id=\"Titre3\"> Boucles </h3>");
  document.write("<table border=\"1\">");
  while ( i<7 && continuer==true) {
    i++;
    if ( i != 5) {
      document.write("<tr><th>"+i+"</th><th>"+tab[i-1]+"</th><th> Exemple "+i+"</th></tr>");
    } else {
      document.write("</table>");
      document.write("<h3 id=\"Titre3\"> Tests </h3>");
      document.write("<table border=\"1\">");
      document.write("<tr><th>"+i+"</th><th>"+tab[i-1]+"</th><th> Exemple "+i+"</th></tr>");
    }
  }
  document.write("</table>");
</script>
```



## Cours Web 2.0

### JavaScript

#### Boucles

1	document.write(...)	Exemple 1
2	boucle while {}	Exemple 2
3	boucle do {} while {}	Exemple 3
4	for(... ; ... ; ... ) {}	Exemple 4

#### Tests

5	switch () { case : ...; break; default: ... }	Exemple 5
6	if (...) {} else {}	Exemple 6
7	undefined	Exemple 7

```
/*Si x est inférieur à 0, l'exception erreur est lancée dans try.*/  
try {  
    if (x>0)  
        { resultat=Math.sqrt(x); }  
    else  
        { throw "erreur"; };  
}  
//ex stocke erreur  
catch (ex)  
{  
    //retourne l'expression indiquée  
    resultat="Opération Impossible";  
    alert(resultat + " " + ex);  
}
```



## 3.4.5.11 - Les 4 Types de bases



- entier

- 127 (base 10), 0755 (base 8), 0xFA15 (base 16)

- flottant

- 0.123, -0.4e5, .67E-89

- booléen

- true, false

- chaîne

- "L'étoile" // L'étoile

- 'filante' // filante

- "'filante'" // 'filante'

- \b, \f, \n, \r, \t, \"



## 3.4.5.12 - Déclaration de variables



### ➤ Pas de déclaration des Variables

- `nbr = 10;`
- `fl = 3.141;`
- `str1 = "L'étoile";`
- `str2 = 'brille';`
- `lien = "<a href=\"index.htm\">Home</a>";`
- `aTest = true;`

### ➤ Portée des Variables

- Local (uniquement dans le script ou la fonction)
  - `var vloc = 0 ;`
- Global (en tout point du document)
  - `vglob = 0 ;`



## 3.4.5.13 - Les Expressions



### ➤ Arithmétique :

- $(3+4) * (56.7 / 89)$

### ➤ Chaîne

- "L'étoile" + " " + "filante"

### ➤ Logique

- `temp == 37`
- `h2o = (temp < 100) ? "eau" : "vapeur";`
- `h2o = (temp > 0) ? ((temp < 100) ? "eau" : "vapeur") : "glace";`





## 3.4.5.14 - Les Opérateurs



### ➤ Affectations

- $+=$ ,  $-=$ ,  $*=$ ,  $/=$ ,  $\%=$ ,  $\&=$ ,  $|=$ ,  $<<=$ ,  $>>=$

### ➤ Comparaisons

- $==$ ,  $!=$ ,  $<$ ,  $<=$ ,  $>$ ,  $>=$

### ➤ Arithmétiques

- $\%$ ,  $++$ ,  $--$

### ➤ Logiques

- $\&\&$ ,  $||$ ,  $!$

### ➤ Bit

- $\&$ ,  $|$ ,  $\wedge$  (XOR),  $<<$ ,  $>>$ ,  $>>>$



## 3.4.5.15a - Les Fonctions



```
function remplirtableau(tab) {
  tab[0] = "document.write(...)";
  tab[1] = "boucle while {}";
  tab[2] = "boucle do {} while {}";
  tab[3] = "for(... ; ... ; ... ) {}";
  tab[4] = "switch () { case : ...; break; default: ...}";
  tab[5] = "if (...) {} else {}";
  tab[6] = "fonction nomFonction( arg1,...) { ... }";
  return tab.length;
}

function lignetableau() {
  document.write("<tr><th>" + arguments[0] + "</th><th>" + arguments[1] + "</th><th> Exemple " + arguments[0]
    + "</th><tr>");
}

function debutableau() {
  document.write("<table border=\"1\">");
}

function fintableau() {
  document.write("</table>");
}

function titretableau(nom) {
  document.write("<h3 id=\"Titre3\">", nom, "</h3>");
}
```



## 3.4.5.15b - Les Fonctions



```
<script type="text/javascript">
  /* Variables globales */
  doctitre1="Cours Web 2.0";
  doctitre2="JavaScript";
  /* Variables locales */
  var tab= new Array(), i=0 , continuer = true;
  taille = remplirtableau(tab);
  /* Write methode de la classe document permettant d'écrire dans le document*/
  document.write("<h1 id=\"Titre1\">"+doctitre1+"</h1>");
  document.write("<h2 id=\"Titre2\">"+doctitre2+"</h2>");
  titretableau("Boucles");
  debutableau();
  for( i=1;i<=taille;i++) {
    if ( i != 5) {
      lignetableau(i,tab[i-1]);
    } else {
      fintableau();
      titretableau("Tests");
      debutableau();
      lignetableau(i,tab[i-1]);
    }
  }
  fintableau();
</script>
```

## Cours Web 2.0

### JavaScript

#### Boucles

1	<code>document.write(...)</code>	Exemple 1
2	<code>boucle while {}</code>	Exemple 2
3	<code>boucle do {} while {}</code>	Exemple 3
4	<code>for(... ; ... ; ... ) {}</code>	Exemple 4

#### Tests

5	<code>switch () { case : ...; break; default: ...}</code>	Exemple 5
6	<code>if (...) {} else {}</code>	Exemple 6
7	<code>fonction nomFonction( arg1,...) { ... }</code>	Exemple 7



## 3.4.5.16a - Les Objets : Exemple



```
function Tableau(nom) {
  this.nomtab = nom;
  this.tab = new Array();
  this.border_width="5px";
  this.color="red";
  this.ligne = function(numero,valeur) {
    document.write("<tr><th>" + numero + "<th>" + valeur + "<th>" + "Exemple " + numero + "<th>" + "<tr>");
  }
  this.debut = function(){
    document.write("<table style=\"border-width:\",this.border_width,\";color:\",this.color,\" \">");
  }
  this.fin = function() {
    document.write("<table>");
  }
  this.titre = function(nom){
    document.write("<h3 id=\"Titre3\">", nom , "<h3>");
  }
}

this.remplir = function(){
  this.tab[0] = "document.write(...)";
  this.tab[1] = "boucle while {}";
  this.tab[2] = "boucle do {} while {}";
  this.tab[3] = "for(... ; ... ; ... ) {}";
  this.tab[4] = "switch () { case : ...; break; default: ...}";
  this.tab[5] = "if (...) {} else {}";
  this.tab[6] = "fonction nomFonction( arg1,...) { ... }";
  return this.tab.length;
}
}
```





## 3.4.5.16b - Les Objets : Exemple

```
<script type="text/javascript">
  /* Variables locales */
  var tab= new Tableau("Tableau");
  var tab2= new Tableau("Tableau2");
  tab2.color="blue"
  tab.afficher = function(titre1,titre2) {
    taille = tab.remplir();
    document.write("<h1 id=\"Titre1\">"+titre1+"</h1>");
    document.write("<h2 id=\"Titre2\">"+titre2+"</h2>");
    tab.titre("Boucles");
    tab.debut();
    for( i=1;i<=taille;i++) {
      if ( i != 5 && i != 7) {
        tab.ligne(i,tab.tab[i-1]);
      } else {
        tab.fin();
        switch (i) {
          case 5 : tab.titre("Tests");break;
          case 7 : tab.titre("Fonctions");break;
        }
        tab.debut();
        tab.ligne(i,tab.tab[i-1]);
      }
    }
    tab.fin();
  }
</script>
```



## 3.4.5.16c - Les Objets : Exemple



```
<script type="text/javascript">  
  doctitre1="Cours Web 2.0";  
  doctitre2="JavaScript";  
  tab.afficher(doctitre1,doctitre2);  
  tab2.titre("Tableau2");  
  tab2.debut();  
  tab2.ligne(i,"propriete");  
  tab2.fin();  
</script>
```

## Cours Web 2.0

### JavaScript

#### Boucles

1	<code>document.write(...)</code>	Exemple 1
2	<code>boucle while ()</code>	Exemple 2
3	<code>boucle do {} while ()</code>	Exemple 3
4	<code>for( ... ; ... ; ... ) {}</code>	Exemple 4

#### Tests

5	<code>switch {} { case : ...; break; default: ... }</code>	Exemple 5
6	<code>if (...) {} else {}</code>	Exemple 6

#### Fonctions

7	<code>function nomFonction( arg1... ) { ... }</code>	Exemple 7
---	--	-----------

#### Tableau2

8	<code>propriete</code>	Exemple 8
---	------------------------	-----------



cd Tableau

**Tableau**

+ Tableau(char) : void

Déclaration : `function Tableau(nom) { ... ... }`

Utilisation : `var tab = new Tableau("tab1");`

cd Rectangle

**Rectangle**

+ Rectangle(double, double) : void

Déclaration : `function Rectangle(a,b) { ... ... }`

Utilisation : `var rec = new Rectangle(3,4);`

cd Tableau

**Tableau**

- nomtab: char
- tab: Array
- border\_width: char
- color: char

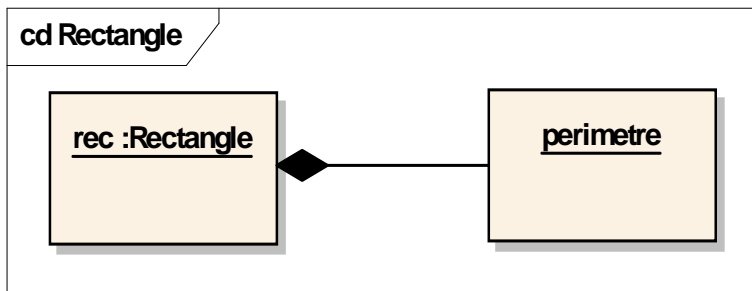
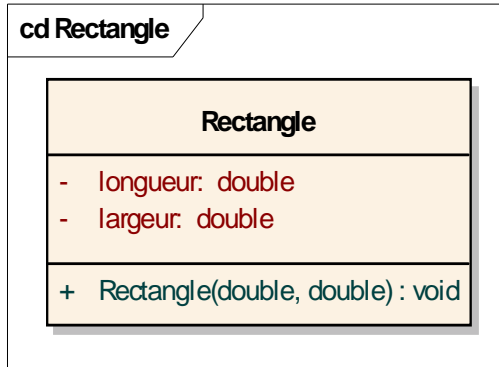
+ Tableau(char) : void

Déclaration :

```
function Tableau(nom) {  
    // propriétés liée à la classe  
    this.nomtab = nom;  
    this.tab = new Array();  
    this.border_width="5px";  
    this.color="red";  
}
```

Utilisation :

```
var tab = new Tableau("tab1");  
tab.border_width="3px";  
tab.nomtab="tab2";
```



Déclaration :

```

function Rectangle(a,b) {
  // propriétés liée à la classe
  this.longueur=a;
  this.largeur= b;
}
  
```

Utilisation:

```

var rec = new Rectangle(3,4);
rec.longueur=30;
rec.largeur=40;
  
```

```

// propriétés liée à l'instance créée
rec.perimetre= 2 * ( rec.longueur + rec.largeur);
  
```



## 3.4.5.16.3a - Les Objets : Méthodes

### cd Tableau

#### Tableau

- nomtab: char
- tab: Array
- border\_width: char
- color: char

- + Tableau(char) : void
- debut() : void
- fin() : void
- + titre(char) : void
- + remplir() : int

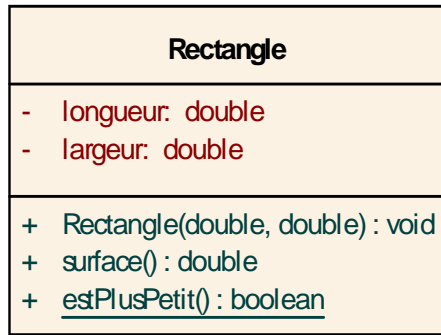
```
function Tableau(nom) {
    this.nomtab = nom;
    this.tab = new Array();
    this.border_width="5px";
    this.color="red";
    this.ligne = function(numero,valeur) {
        document.write("<tr><th>" + numero + "<th>" + valeur + "<th><th>"
        Exemple " + numero + "<th><th>"); }

    this.debut = function(){
        document.write("<table style=\"border-width:\",this.border_width,\"color:\",this.color,\""
        ">");
    }
    this.fin = function() { document.write("<table>"); }

    this.titre = function(nom) { document.write("<h3 id=\"Titre3\">", nom , "<h3>"); }

    this.remplir = function(){
        this.tab[0] = "document.write(...)";
        this.tab[1] = "boucle while {}";
        this.tab[6] = "fonction nomFonction( arg1,...) { ... }";
        return this.tab.length;
    }
}
```

cd Rectangle



Déclaration:

```
function Rectangle(a,b) {
  // propriétés liée à la classe
  this.longueur=a;
  this.largeur= b;
  this.surface = function() { return a*b; }
}
```

// méthode liée à la classe

```
Rectangle.estPlusPetit = function( a, b ) {
  return (a.surface < b .surface); }
```

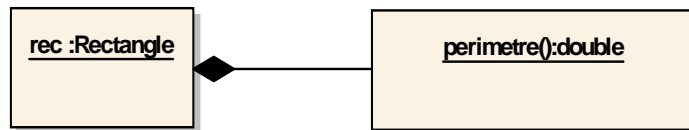
Utilisation:

```
var rec = new Rectangle(3,4);
```

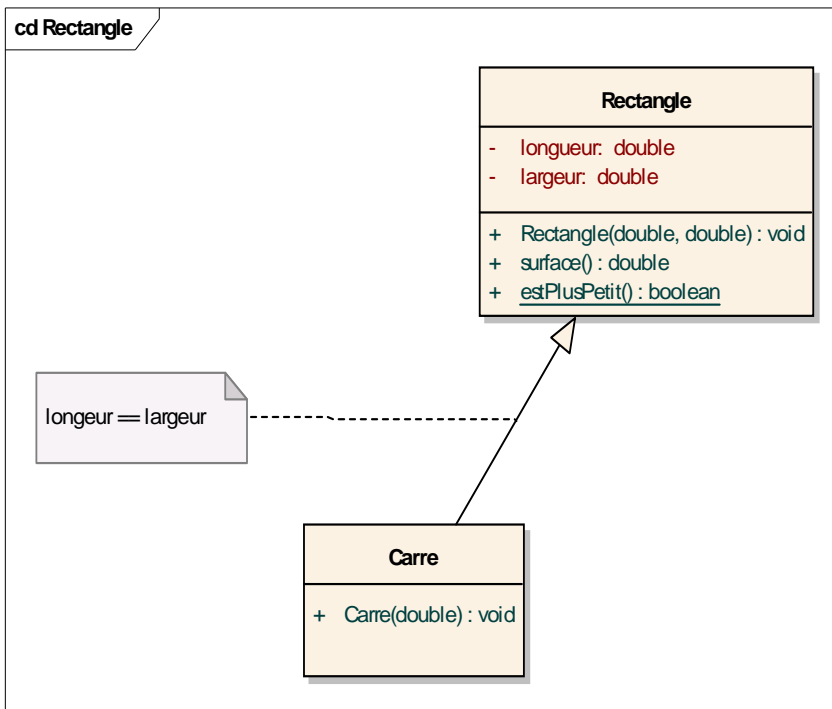
// méthode liée à l'instance créée

```
rec.perimetre = function()
{ return 2 * ( this.longueur + this.largeur); }
```

cd Rectangle







### Déclaration:

```

function Rectangle(a,b) {
  // propriétés liée à la classe
  this.longueur=a;
  this.largeur= b;
  this.surface = function() { return a*b; }
}
  
```

```

function Carre(a) {
  // propriétés liée à la classe
  this.longueur=a;
  this.largeur= a;
  this.coté = function() { return this.longueur; }
}
  
```

### Utilisation:

```
Carre.prototype = new Rectangle('x','y');
```

```

var car = new Carre(5);
var surface = car.surface();
  
```



## 3.4.5.17 - Les Objets prédéfinis



- **Array** : L'objet Array permet de créer des tableaux. Il possède de nombreuses méthodes permettant d'ajouter, de supprimer ou d'extraire des éléments d'un tableau ainsi que de les trier.
- **Boolean** : L'objet Boolean permet de créer des valeurs booléennes, c'est-à-dire des éléments possédant deux états : *vrai* et *faux*.
- **Date** : L'objet Date permet de créer des dates et des durées. Il propose des méthodes permettant de les manipuler.
- **Math** : L'objet Math permet de manipuler des fonctions mathématiques, comme par exemple les fonctions trigonométriques.
- **String** : L'objet String propose une grande variété de méthodes permettant de manipuler des chaînes de caractères.
- **RegExp** : L'objet RegExp permet de créer des expressions régulières, c'est-à-dire des éléments permettant de faire des opérations avancées sur les chaînes de caractères.
- **Function** : L'objet Function permet de définir des fonctions personnalisées.
- **Number** : L'objet Number est un objet permettant de faire des opérations de base sur les nombres.



## 3.4.5.17.1a - Les Objets prédéfinis : Array



Constructeur	Description
<b>Array()</b>	Sans paramètre
<b>Array(n)</b>	Prends la taille en paramètre
<b>Array(ele0, ... ,eleN-1)</b>	Prends une liste d'éléments

Propriété	Description
<b>constructor</b>	Contient le constructeur de l'objet <i>Array</i> .
<b>input</b>	Permet de faire une recherche dans le tableau à l'aide d'une expression régulière
<b>length</b>	Contient le nombre d'éléments du tableau.
<b>prototype</b>	Permet d'ajouter des propriétés personnalisées à l'objet.



## 3.4.5.17.1b - Les Objets prédéfinis : Array



Méthode	Description
<b>concat</b> (tab1, tab2[, tab3, ...])	Permet de concaténer plusieurs tableaux
<b>join</b> (tableau) ou <b>Tableau.join()</b>	Renvoie une chaîne de caractères contenant tous les éléments du tableau.
<b>pop</b> (tableau) ou <b>Tableau.pop()</b>	Supprime le dernier élément du tableau et retourne sa valeur.
<b>Tableau.push</b> (valeur1[, valeur2, ...])	Ajoute un ou plusieurs éléments au tableau.
<b>Tableau.reverse()</b>	Inverse l'ordre des éléments du tableau.
<b>Tableau.shift()</b>	Supprime le premier élément du tableau.
<b>Tableau.slice()</b>	Renvoie un tableau contenant une partie des éléments d'un tableau.
<b>Tableau.splice()</b>	Ajoute/retire des éléments d'un tableau.
<b>Tableau.sort()</b>	Permet de trier les éléments d'un tableau.
<b>Tableau.unshift</b> (valeur1[, valeur2, ...])	Renvoie le code source qui a permis de créer l'objet <i>Array</i> .
<b>Tableau.toString()</b>	Renvoie la chaîne de caractères correspond à l'instruction qui a permis de créer l'objet <i>Array</i> .
<b>Tableau.unshift()</b>	Permet d'ajouter un ou plusieurs élément au début du tableau.
<b>Tableau.valueOf</b>	Retourne tout simplement la valeur de l'objet <i>Array</i> auquel elle fait référence.



## 3.4.5.17.2 - Les Objets prédéfinis : Boolean



Constructeur	Description
<b>Boolean(valeur)</b>	Prends en paramètre une valeur

Propriété	Description
<b>constructor</b>	Contient le constructor l'objet <i>Boolean</i> .
<b>prototype</b>	Permet d'ajouter des propriétés personnalisées à l'objet.

Méthode	description
<b>toSource()</b>	Renvoie le code source qui a permis de créer l'objet <i>Boolean</i> .
<b>toString()</b>	Renvoie la chaîne de caractères correspond à l'instruction qui a permis de créer l'objet <i>Boolean</i> .
<b>valueOf</b>	Retourne la valeur de l'objet <i>Boolean</i> auquel elle fait référence.





## 3.4.5.17.3a - Les Objets prédéfinis : Date



Constructeur	Description
Date()	Sans argument, permet de stocker la date et l'heure actuelle
Date("jour mois date année heures:minutes:secondes")	les arguments sont une chaîne de caractère suivant la notation ci-contre
Date(année, mois, jour)	Les arguments sont trois entiers séparés par des virgules. Les paramètres omis sont mis à zéro par défaut
Date(année, mois, jour, heures, minutes, secondes[, millisecondes])	les arguments sont six entiers séparés par des virgules. Les paramètres omis sont mis à zéro par défaut



## 3.4.5.17.3b - Les Objets prédéfinis : Date



Méthode	Description	Type de valeurs retournée
getDate()	Permet de récupérer la valeur du jour du mois	L'objet retourné est un entier (entre 1 et 31) qui correspond au jour du mois :
getDay()	Permet de récupérer la valeur du jour de la semaine pour la date spécifiée	L'objet retourné est un entier qui correspond au jour de la semaine : <ul style="list-style-type: none"><li>•0: dimanche</li><li>•1: lundi ...</li></ul>
getFullYear()	Permet de récupérer la valeur de l'année sur 4 chiffres pour la date passée en paramètre	L'objet retourné est un entier qui correspond à l'année (XXXX) : 2007
getHours()	Permet de récupérer la valeur de l'heure	L'objet retourné est un entier (entre 0 et 23) qui correspond à l'objet Date.
getMilliseconds()	Permet de récupérer le nombre de millisecondes	L'objet retourné est un entier (entre 0 et 999) qui correspond aux millisecondes de l'objet passé en paramètre.
getMinutes()	Permet de récupérer la valeur des minutes	L'objet retourné est un entier (entre 0 et 59) qui correspond aux minutes de l'objet Date.
getMonth()	Permet de récupérer le numéro du mois	L'objet retourné est un entier (entre 0 et 11) qui correspond au mois : <ul style="list-style-type: none"><li>•0: janvier</li><li>•1: février ...</li></ul>
getSeconds()	Permet de récupérer le nombre de secondes	L'objet retourné est un entier (entre 0 et 59) qui correspond aux secondes de l'objet passé en paramètre.
getTime()	Permet de récupérer le nombre de millisecondes depuis le 1 <sup>er</sup> janvier 1970	L'objet retourné est un entier. Cette méthode est très utile pour convertir des dates, soustraire ou ajouter deux dates, etc.
getTimezoneOffset() )	Retourne la différence entre l'heure locale et l'heure GMT (Greenwich Mean Time)	L'objet retourné est un entier, il représente le nombre de <b>minutes</b> de décalage
getYear()	Permet de récupérer la valeur de l'année sur 2 chiffres pour l'objet Date.	L'objet retourné est un entier qui correspond à l'année (XX) :



## 3.4.5.17.3c - Les Objets prédéfinis : Date

Méthode de mise en forme de la date	Description	Type de valeurs retournée
toGMTString()	Permet de convertir une date en une chaîne de caractères au format GMT	L'objet retourné est une chaîne de caractère du type : Wed, 28 Jul 1999 15:15:20 GMT
toLocaleString()	Permet de convertir une date en une chaîne de caractères au format local	L'objet retourné est une chaîne de caractère dont la syntaxe dépend du système, par exemple : 28/07/99 15:15:20

Méthode de modification	Description	Type de valeur en paramètre
setDate(X)	Permet de fixer la valeur du jour du mois	Le paramètre est un entier (entre 1 et 31) qui correspond au jour du mois
setDay(X)	Permet de fixer la valeur du jour de la semaine	Le paramètre est un entier qui correspond au jour de la semaine : <ul style="list-style-type: none"><li>0: dimanche</li><li>1: lundi ...</li></ul>
setHours(X)	Permet de fixer la valeur de l'heure	Le paramètre est un entier (entre 0 et 23) qui correspond à l'heure
setMinutes(X)	Permet de fixer la valeur des minutes	Le paramètre est un entier (entre 0 et 59) qui correspond aux minutes
setMonth(X)	Permet de fixer le numéro du mois	Le paramètre est un entier (entre 0 et 11) qui correspond au mois : <ul style="list-style-type: none"><li>0: janvier</li><li>1: février ...</li></ul>
setTime(X)	Permet d'assigner la date	Le paramètre est un entier représentant le nombre de millisecondes depuis le 1 <sup>er</sup> janvier 1970



## 3.4.5.17.4a - Les Objets prédéfinis : Math

Méthode	Description	Exemple
<b>abs()</b>	Cette méthode renvoie la valeur absolue d'un nombre, il renvoie donc le nombre s'il est positif, son opposé (positif) s'il est négatif	<pre>x = Math.abs(3.26); //donne x = 3.26 x = Math.abs(-3.26); //donne x = 3.26</pre>
<b>ceil()</b>	Renvoie le plus petit entier supérieur ou égal à la valeur donnée en paramètre	<pre>x = Math.ceil(6.01); //donne x = 7 x = Math.ceil(3.99); //donne x = 4</pre>
<b>floor()</b>	La méthode <i>floor()</i> retourne le plus grand entier inférieur ou égal à la valeur donnée en paramètre.	<pre>x = Math.floor(6.01); //donne x = 6 x = Math.floor(3.99); //donne x = 3</pre>
<b>round()</b>	Arrondit à l'entier le plus proche la valeur donnée en paramètre. Si la partie décimale de la valeur entrée en paramètre vaut 0.5, la méthode <i>Math()</i> arrondi à l'entier supérieur.	<pre>x = Math.round(6.01); //donne x = 6 x = Math.round(3.80); //donne x = 4 x = Math.round(3.50); //donne x = 4</pre>
<b>max</b> (Nombre1, Nombre2)	<i>max()</i> renvoie le plus grand des deux nombres donnés en paramètre	<pre>var x = Math.max(6,7.25); //donne x = 7.25 var x = Math.max(-8.21,-3.65); //donne x = -3.65 var x = Math.max(5,5); //donne x = 5</pre>
<b>min</b> (Nombre1, Nombre2)	Retourne le plus petit des deux nombres donnés en paramètre	<pre>x = Math.min(6,7.25); //donne x = 6 x = Math.min(-8.21,-3.65); //donne x = -8.21 x = Math.min(5,5); //donne x = 5</pre>
<b>pow</b> (Valeur1, Valeur2)	Retourne le nombre <i>Valeur1</i> à la puissance <i>Valeur2</i>	<pre>x = Math.pow(3,3); //donne x = 27 x = Math.pow(9,0.5); //(racine carrée) //donne x = 3</pre>
<b>random()</b>	La méthode <i>random()</i> renvoie un nombre pseudo-aléatoire compris entre 0 et 1. La valeur est générée à partir des données de l'horloge de l'ordinateur.	<pre>x = Math.random(); //donne x = 0.6489534931546957</pre>
<b>sqrt</b> (Valeur)	Renvoie la racine carrée du nombre passé en paramètre	<pre>x = Math.sqrt(9); //donne x = 3</pre>





## 3.4.5.17.4b - Les Objets prédéfinis : Math



Méthode Logarithmes et exponentielle	Description
<b>Math.E</b>	Propriété qui retourne le nombre d'Euler (environ 2.718).
<b>Math.exp(valeur)</b>	Cette méthode renvoie l'exponentielle de la valeur entrée en paramètre.
<b>Math.LN2</b>	La propriété <i>LN2</i> fournit le logarithme népérien de 2.
<b>Math.LN10</b>	Propriété donne le logarithme népérien de 10.
<b>Math.log(valeur)</b>	La méthode <i>log()</i> renvoie le logarithme de la valeur entrée en paramètre.
<b>Math.LOG2E</b>	Propriété qui renvoie la valeur du logarithme du nombre d'Euler en base 2.
<b>Math.SQRT1_2</b>	Propriété qui retourne la valeur de "1 divisé par racine de 2" (0.707).
<b>Math.SQRT2</b>	La propriété <i>SQRT2</i> ( <i>Square Root 2</i> ) donne la racine de 2 (1.414).



## 3.4.5.17.4c - Les Objets prédéfinis : Math



Méthode	Description
<b>Math.PI</b>	Retourne la valeur du nombre PI, soit environ 3.1415927
<b>Math.sin(valeur)</b>	Retourne le sinus de la valeur entrée en paramètre (doit être donnée en radians). La valeur retourné est comprise dans l'intervalle [-1;1].
<b>Math.asin(valeur)</b>	Retourne l'arcsinus de la valeur entrée en paramètre. La valeur doit être comprise dans l'intervalle [-1;1]. Dans le cas contraire, la méthode <i>asin()</i> renvoie la valeur <i>NaN</i> ( <i>Not a Number</i> ).
<b>Math.cos(valeur)</b>	Retourne le cosinus de la valeur entrée en paramètre (doit être donnée en radians). La valeur retourné est comprise dans l'intervalle [-1;1].
<b>Math.acos(valeur)</b>	Retourne l'arccosinus de la valeur entrée en paramètre. La valeur doit être comprise dans l'intervalle [-1;1]. Dans le cas contraire, la méthode <i>acos()</i> renvoie la valeur <i>NaN</i> ( <i>Not a Number</i> ).
<b>Math.tan(valeur)</b>	Retourne la tangente de la valeur entrée en paramètre (doit être donnée en radians)
<b>Math.atan(valeur)</b>	Retourne l'arctangente de la valeur entrée en paramètre. La valeur doit être comprise dans l'intervalle [-1;1]. Dans le cas contraire, la méthode <i>atan()</i> renvoie la valeur <i>NaN</i> ( <i>Not a Number</i> ).



## 3.4.5.17.5a - Les Objets prédéfinis : String



Méthode	Description
<b>String.anchor</b> ("nom_a_donner");	Transforme le texte <i>String</i> en <a href="#">ancrage HTML</a> .
<b>String.big</b> ()	Augmente la taille de la police.
<b>String.blink</b> ()	Transforme la chaîne en texte clignotant.
<b>String.bold</b> ()	Met le texte en <b>gras</b> (balise <B>).
<b>String.charAt</b> (position)	Retourne le caractère situé à la position donnée en paramètre
<b>String.charCodeAt</b> (position)	Renvoie le code <i>Unicode</i> du caractère situé à la position donnée en paramètre
<b>String.concat</b> (chaîne1, chaîne2[, ...])	Permet de concaténer les chaînes passées en paramètre, c'est-à-dire de les joindre bout à bout.
<b>String.fixed</b> ()	Transforme la <i>String</i> en caractères de police fixe (balise <TT>)
<b>String.fontcolor</b> (couleur)	Modifie la couleur du texte (admet comme argument la couleur en hexadécimal ou en valeur littérale)
<b>String.fontSize</b> (Size)	Modifie la taille de la police, en affectant la valeur passée en paramètre
<b>String.fromCharCode</b> (code1[, code2, ...])	Renvoie une chaîne de caractères composée de caractères correspondant au(x) code(s) <i>Unicode</i> donné(s) en paramètre.
<b>String.indexOf</b> (sous-chaîne, position)	Retourne la position d'une sous-chaîne (lettre ou groupe de lettres) dans une chaîne de caractère, en effectuant la recherche de gauche à droite, à partir de la position spécifiée en paramètre.
<b>String italics</b> ()	Transforme le texte en <i>italique</i> (balise <I>)



## 3.4.5.17.5b - Les Objets prédéfinis : String



Méthode	Description
<b>String.lastIndexOf(sous-chaîne, position)</b>	La méthode est similaire à <i>indexOf()</i> , à la différence que la recherche se fait de droite à gauche : Retourne la position d'une sous-chaîne (lettre ou groupe de lettres) dans une chaîne de caractère, en effectuant la recherche de droite à gauche, à partir de la position spécifiée en paramètre.
<b>String.link(URL)</b>	Transforme le texte en hypertexte (balise <A href>)
<b>String.small()</b>	Diminue la taille de la police
<b>String.strike()</b>	Transforme le texte en texte barré (balise <strike>)
<b>String.sub()</b>	Transforme le texte en <sub>indice</sub> (balise <sub>)
<b>String.substr(position1, longueur)</b>	La méthode retourne une sous-chaîne commençant à l'index dont la position est donnée en argument et de la longueur donnée en paramètre.
<b>String.substring(position1, position2)</b>	La méthode retourne la sous-chaîne (lettre ou groupe de lettres) comprise entre les positions 1 et 2 données en paramètre.
<b>String.sup()</b>	Transforme le texte en <sup>exposant</sup> (balise <sup>).
<b>String.toLowerCase()</b>	Convertit tous les caractères d'une chaîne en minuscule.
<b>String.toSource()</b>	Renvoie le code source de création de l'objet.
<b>String.toUpperCase()</b>	Convertit tous les caractères d'une chaîne en majuscule.
<b>String.valueOf()</b>	Renvoie la valeur de l'objet String.





## 3.4.5.17.6a - Les Objets prédéfinis : RegExp



L'objet *RegExp* permet de manipuler des expressions régulières, c'est-à-dire des modèles créés à l'aide de caractères ASCII permettant de manipuler des chaînes de caractères, afin de trouver des portions de la chaîne correspondant au modèle.

```
Expression = new RegExp("motif","option") ;
```

Option :

- **i** : ignorer la casse (diff maj/min) ;
- **g** : recherche globale (toutes les occurrences) ;
- **gi** : recherche globale ignorant la casse ;

```
var re = /(w+)\s(w+)/; //définit un masque de
    recherche
var str = "Jean Dupont";
var newstr = str.replace(re, "$2, $1");
document.write(newstr);
```

Ce code affichera « Dupont, Jean ».

```
var s = "Voila une chaine de caractères."
var re = new RegExp("ne\\b","gi");
while (true) {
    res = re.exec(s);
    if ( res == null ) break;
    document.writeln("<P>J'ai trouvé \""+res[0]+"\" en position "+
        res.index+" dans \""+res.input+"\". ");
    document.writeln("Je l'ai trouvée entre ["+RegExp.leftContext+"]
        et ["+RegExp.rightContext+"]");
}
```



## 3.4.5.17.6b - Les Objets prédéfinis : RegExp



Caractère	Exemples	Commentaire
^ \$		Début et fin de la saisie
[ ]	[asp] [0-9] [a-zA-Z]	Liste de caractères
[^ ]	[^asp] [^0-9] [^a-zA-Z]	Liste de caractères exclus
(modele)		Modèle
x y	(f z)oo : marche pour foot ou zoo	x ou y
*		Le caractère précédent apparaît 0 ou plusieurs fois
+		Le caractère précédent apparaît 1 ou plusieurs fois
?		Le caractère précédent apparaît 0 ou 1 fois
{n}		Le caractère précédent apparaît exactement n fois
{n,}		Le caractère précédent apparaît au moins n fois
{n,m}		Le caractère précédent apparaît de n à m fois
\	\\ --> \    \ ( --> (    \ ) --> ) \n --> nouvelle ligne \f --> saut de page \t --> tabulation	Le caractère suivant est spécial
\b		Fin d'un mot
\B		Pas à la fin d'un mot
\xHH	\x41 <--> A	Code hexadécimal d'un caractère
\s	\s <--> [ \f\n\r\t\v]	Tout caractère séparateur
\S	\S <--> [^ \f\n\r\t\v]	Tout caractère non séparateur
\ autres	\d <--> [0-9] \D <--> [^0-9] \w <--> [A-Za-z0-9_] \W <--> [^A-Za-z0-9_]	Raccourcis



## 3.4.5.17.6c - Les Objets prédéfinis : RegExp



Propriété	Description
\$_	Propriété correspondant à la propriété <i>input</i> .
\$*	Propriété correspondant à la propriété <i>multiline</i> .
\$&	Propriété correspondant à la propriété <i>LastMatch</i> .
\$+	Propriété correspondant à la propriété <i>LastParen</i> .
\$`	Propriété correspondant à la propriété <i>LeftContext</i> .
\$'	Propriété correspondant à la propriété <i>RightContext</i> .
global	Propriété booléenne indiquant si la recherche est globale ( <i>true</i> ) ou non ( <i>false</i> ).
ignoreCase	Propriété booléenne indiquant si la recherche est sensible à la casse ( <i>true</i> ) ou non ( <i>false</i> ).
input	Indique la chaîne d'entrée sur laquelle la recherche est réalisée.
lastIndex	Indique la position à laquelle la recherche suivante va se faire.
lastMatch	Contient la dernière occurrence trouvée.
lastParen	Contient la dernière occurrence correspondant à un motif entre parenthèses.
leftContext	Contient la chaîne situé à gauche de l'occurrence trouvée.
multiline	Propriété booléenne indiquant si la recherche porte sur plusieurs lignes ( <i>true</i> ) ou non ( <i>false</i> ).
rightContext	Contient la chaîne situé à droite de l'occurrence trouvée.
source	Contient le motif de l'expression régulière.



## 3.4.5.17.6d - Les Objets prédéfinis : RegExp



Méthode	Description
<b>Expression.compile("chaîne");</b>	Permet de redéfinir une nouvelle expression régulière.
<b>Expression.exec("chaîne");</b>	Effectue une recherche sur la chaîne de caractère avec l'expression régulière définie. Cette méthode retourne un tableau contenant les occurrences trouvées.
<b>Expression.test("chaîne");</b>	Teste une chaîne de caractère avec l'expression régulière. Cette méthode retourne <i>True</i> si la recherche est fructueuse, <i>false</i> dans le cas contraire.



## 3.4.5.17.7 - Les Objets prédéfinis : Function

Constructeur	Description
<b>Function("nom", "corps fonction");</b>	<p>Exemple :</p> <pre>couleur = new     Function("nouvelle_couleur", "document.bgColor=nouvelle_co         uleur;");</pre> <p>Correspond à :</p> <pre>function couleur(nouvelle_couleur) {     document.bgColor=nouvelle_couleur; }</pre> <p>Exemple d'appelle de la fonction :</p> <pre>&lt;a href="javascript:couleur('#FFCC99')"&gt;Nouvel arrière plan&lt;/a&gt;</pre>
Propriété	Description
<b>arguments</b>	tableau de noms d'arguments
<b>arity</b>	nombre d'arguments
<b>caller</b>	nom de la fonction qui appelle





## 3.4.5.17.8 - Les Objets prédéfinis : Number

Propriété	Description
<b>MAX_VALUE</b>	plus grand nombre pouvant être sauvegardé
<b>MIN_VALUE</b>	plus petit nombre pouvant être sauvegardé
<b>Nan</b>	nombre non valable
<b>NEGATIVE_INFINITY</b>	nombre trop petit
<b>POSITIVE_INFINITY</b>	nombre trop grand
Méthode	Description
<b>toExponential()</b>	impose la formulation exponentielle d'un nombre.
<b>toFixed(nb)</b>	impose un certain nombre de chiffres après la virgule.
<b>toPrecision(nb)</b>	Impose une certaine exactitude pour la représentation d'un nombre.
<b>toString()</b>	Transforme un nombre en une chaîne de caractères



## 3.4.5.18.1 - Les Objets prédéfinis du navigateur



- Navigator : qui contient des informations sur le navigateur de celui qui visite la page
- Windows : c'est l'objet où s'affiche la page, il contient donc des propriétés concernant la fenêtre elle-même mais aussi tous les objets-enfants contenus dans celle-ci
  - history: c'est l'historique, c'est-à-dire la liste de liens qui ont été visités précédemment
  - location : contient des informations relatives à l'adresse de la page à l'écran
  - document : il contient les propriétés sur le contenu du document (couleur d'arrière plan, titre, ...)



## 3.4.5.18.1a - Les Objets prédéfinis du navigateur : Windows



Propriété	Description	Lecture seule
defaultStatus	Il s'agit du message qui s'affiche par défaut dans la barre d'état du navigateur	non, modifiable à tout moment
frames	Il s'agit d'un tableau qui contient les cadres présents dans la fenêtre	Tous les éléments de <i>frames</i> sont en lecture seule
length	nombre de cadres (nombre d'éléments du tableau <i>frames</i> )	Lecture seule
name	nom de la fenêtre dans laquelle on se trouve	Lecture seule
parent	Il s'agit de la fenêtre qui englobe celle dans laquelle on se trouve (si il y en a une..)	Lecture seule, contient des propriétés
self	Synonyme de la fenêtre actuelle (redondance ou précision?)	Lecture seule, contient des propriétés
status	Il s'agit d'un message temporaire qui s'affiche dans la barre d'état du navigateur suite à un événement	non, modifiable à tout moment, vous devez retourner la valeur <i>true</i> pour l'utiliser avec <i>onMouseOver</i>
top	Il s'agit de la fenêtre de plus haut niveau, celle qui contient tous les cadres ( <i>frames</i> )	Lecture seule, contient des propriétés
window	Il s'agit de la fenêtre actuelle...	Lecture seule, contient des propriétés



## 3.4.5.18.1b - Les Objets prédéfinis du navigateur : Windows



Méthode	Description
alert("Message")	permet d'afficher dans une boîte toute simple
confirm("Message")	similaire à la méthode alert(), si ce n'est qu'elle permet un choix entre "OK" et "Annuler"
prompt("Message", "chaine par défaut")	fournit un moyen simple de récupérer une information provenant de l'utilisateur, on parle alors de boîte d'invite
open("URL","nom_de_la_fenetre","options_de_la_fenetre" )	ouverture de fenêtres Option : <ul style="list-style-type: none"><li>• directories = yes/no, Affiche ou non les boutons de navigation</li><li>• location = yes/no, Affiche ou non la barre d'adresse</li><li>• menubar = yes/no ,Affiche ou non la barre de menu (fichier, édition, ...)</li><li>• resizable = yes/no ,Définit si la taille de la fenêtre est modifiable ou non</li><li>• scrollbars = yes/no , Affiche ou non les ascenseurs (barres de défilement)</li><li>• status = yes/no, Affiche ou non la barre d'état</li><li>• toolbar = yes/no, Affiche ou non la barre d'outils</li><li>• width = largeur (en pixels), Définit la largeur</li><li>• height = hauteur (en pixels), Définit la hauteur</li></ul>
close()	fermeture de fenêtres



## 3.4.5.18.2a - Les Objets prédéfinis du navigateur : Navigator



Propriété	Description
<i>appCodeName</i>	retourne le code du navigateur. Un navigateur a généralement pour nom de code <i>Mozilla</i> , le moteur utilisé par la plupart des navigateurs (internet explorer, netscape, mais aussi beaucoup de navigateurs sous Unix...). Cette valeur sera différente si le navigateur du client est pas basé sur un autre moteur (e.g. Opera, ...).
<i>appName</i>	retourne le nom du navigateur (la plupart du temps la marque). Cette propriété est utile pour différencier les navigateurs de Netscape et de Microsoft).
<i>appVersion</i>	retourne la version du navigateur. Cette propriété prend la forme suivante : Numéro de version( plateforme (système d'exploitation), nationalité) Elle est utile pour connaître le système d'exploitation de l'utilisateur, mais surtout, associée avec la propriété <i>navigator.appName</i> elle permet de connaître les fonctionnalités que supporte le navigateur de votre visiteur.
<i>language</i>	renvoie une chaîne de caractère donnant la langue utilisée par la navigateur du client. Cette propriété n'est comprise que par les navigateurs supportant les versions 1.2 et supérieures de Javascript.
<i>mimeTypes</i>	Cette propriété renvoie un tableau répertoriant les types MIME supportés par le navigateur, c'est-à-dire les types de fichiers enregistrés.
<i>platform</i>	Cette propriété renvoie une chaîne de caractère indiquant la plateforme sur laquelle le navigateur fonctionne, c'est-à-dire le système d'exploitation du client. Cette propriété n'est comprise que par les navigateurs supportant les versions 1.2 et supérieures de Javascript.
<i>plugins</i>	Cette propriété renvoie un tableau contenant la liste des plugins installés sur la machine cliente.
<i>userAgent</i>	retourne la chaîne de caractère qui comprend toutes les informations sur le navigateur de client. Les propriétés ci-dessus offrent un moyen pratique de récupérer une partie de cette information.





## 3.4.5.18.2b - Les Objets prédéfinis du navigateur : Navigator



Méthode	Description
<i>javaEnabled()</i>	Cette méthode permet de vérifier si le navigateur du client est configuré pour exécuter des applets <a href="#">Java</a> .
<i>plugins.refresh()</i>	La méthode <i>refresh()</i> de la propriété <i>plugin</i> permet de rafraîchir la liste des plugins installés sur le poste client.
<i>preference("preference", valeur)</i>	Cette méthode supportée à partir de la version 1.2 de Javascript permet à un script signé de redéfinir les préférences du navigateur. Le script doit ainsi obtenir les privilèges suffisants pour pouvoir effectuer ces actions.
<i>SavePreferences()</i>	Cette méthode supportée à partir de la version 1.2 de Javascript permet de sauvegarder les modifications apportées aux préférences du navigateur du client.
<i>taintEnabled()</i>	Cette méthode permet de vérifier que la protection des données a été activée grâce à la méthode <i>taint()</i> de Javascript. Cette méthode est obsolète depuis la version 1.2 de Javascript, et ne fonctionne ainsi que sur <i>Netscape Navigator 3</i> .



### 3.4.5.18.3 - Les Objets prédéfinis du navigateur : History



Propriété	Description
length	permet de connaître le nombre d'objets dans l'historique
Méthode	Description
back	permet d'aller à l'URL précédent dans l'historique
forward	permet d'aller à l'URL suivant dans l'historique
go (variable)	permet d'aller à un des URL de l'historique. Le paramètre variable est un nombre entier (positif ou négatif) qui détermine le nombre de pages relatif auquel se trouve l'URL désirée. Il est possible d'entrer une chaîne de caractères en paramètre, auquel cas le navigateur cherchera la page de l'historique la plus proche contenant cette chaîne.



## 3.4.5.19 - Bibliographie



### ➤ Cours en ligne

- <http://perso.orange.fr/philippe.medan/>
- <http://www.toutjavascript.com/>
- <http://quentinc.net/>

### ➤ Liens Généraliste

- <http://web.developpez.com/>
- <http://www.w3schools.com/>
- <http://www.xul.fr/>

### ➤ Livres

- Ajax par la pratique, O'REILLY
- Pratique de CSS et JavaScript, O'REILLY
- Introduction à HTML et CSS, O'REILLY
- Le guide complet CSS, Micro application
- Le guide complet AJAX , Micro application



### **3.4.6.1 - Définition**

### **3.4.6.2 - Avantages & Inconvénients**

### **3.4.6.3 - Interface IDL de XMLHttpRequest**

### **3.4.6.4 - Propriétés**

### **3.4.6.5 - Méthodes**

### **3.4.6.6 - Exemples**



### 3.4.6.1 - Définition



**XMLHttpRequest est un objet Javascript qui permet d'obtenir, en interagissant avec le serveur, des données au format XML, mais aussi HTML, ou encore texte simple à l'aide de requêtes HTTP.**

#### Historique

XMLHttpRequest a d'abord été développé par Microsoft, en tant qu'objet ActiveX, pour Internet Explorer 5.0. (09/1998).

Il a ensuite été repris et implémenté successivement sous Mozilla 1.0 (05/2002), Safari 1.2 (02/2004), Konqueror 3.4 (03/2005) et dernièrement Opera 8.0 (04/2005).

En raison de ces implémentations plutôt récentes, l'objet n'est pas supporté par les navigateurs dits de « vieille génération ».

En avril 2006, il a été proposé pour devenir une recommandation du W3C.





## 3.4.6.2 - Avantages & Inconvénients



### Avantages

- Gain de bande passante, car la page n'est pas rechargée
- Rapidité d'affichage
- Meilleure interactivité

### Inconvénients

- Ne fonctionne pas si JavaScript est désactivé
- N'est pas compatible avec les anciennes versions de navigateur



### 3.4.6.3 - Interface IDL de XMLHttpRequest



```
interface XMLHttpRequest {  
  
    attribut EventListener onreadystatechange;  
    readonl attribut unsigned short readyState;  
    attribut DOMString responseText;  
    attribut Document responseXML;  
    attribut unsigned short status;  
    attribut DOMString statutText;  
  
    void open (en méthode DOMString, en uri DOMString);  
    void open (en méthode DOMString, en uri DOMString, en boolean async);  
    void open (en méthode DOMString, en uri DOMString, en boolean async, en utilisateur DOMString);  
    void open (en méthode DOMString, en uri DOMString, en boolean async, en utilisateur DOMString,  
        mot-de-passe DOMString);  
    void setRequestHeader (en en-tête DOMString, en valeur DOMString) raises(DOMException);  
    void send (en donnée DOMString);  
    void send (en donnée Document);  
    void abort ();  
    DOMString getAllResponseHeaders ();  
    DOMString getResponseHeader (en en-tête DOMString);  
};
```



## 3.4.6.4 - Propriétés



Propriété	Description
readyState	Contient une indication de l'avancement de la requête. L'attribut a successivement l'une des valeurs suivantes:  0 Non initialisé : La valeur de départ. 1 Ouvert : La méthode open() a été appelée avec succès. 2 Envoyé : L'agent utilisateur a effectué la requête avec succès, mais aucune donnée n'a encore été reçue. 3 Réception : Juste avant de recevoir le corps du message (s'il y en a un). Tous les en-têtes HTTP ont été reçus. 4 Chargé : Le transfert de données a été achevé.
statusText	Contient une description du code HTTP retourné par la requête.
status	Contient le code HTTP indiquant le résultat de la dernière requête effectuée. 200 est ok 404 si la page n'est pas trouvée.
responseText	contient les données chargées dans une chaîne de caractères.
responseXml	contient les données chargées sous forme xml, les méthodes de DOM servent à les extraire.
onreadystatechange	Évènement déclenché lorsque le chargement des données est terminé. On lui assigne une fonction.



## 3.4.6.5 - Méthodes



Méthode	Description
open(mode, url, boolean sync, [String userName, String password ])	mode: type de requête, GET,POST ,HEAD ,PUT ,DELETE ,PROPFIND ,PROPPATCH ,MKCOL ,COPY ,MOVE ,LOCK ,UNLOCK ,VERSION-CONTROL ,REPORT ,CHECKOUT ,CHECKIN ,UNCHECKOUT ,MKWORKSPACE ,UPDATE ,LABEL ,MERGE ,BASELINE-CONTROL ,MKACTIVITY ,ORDERPATCH , ACL <i>url</i> : l'endroit où trouver les données, un fichier avec son chemin sur le disque.  <i>sync</i> : true (asynchrone) / false (synchrone). en option on peut ajouter un login et un mot de passe.
setRequestHeader	Définit un paramètre d'en-tête pour la requête.
getAllResponseHeaders	Renvoie la liste de tous les en-têtes HTTP retournés par la requête.
getResponseHeader	Retourne la valeur d'un en-tête HTTP.
abandon	Annule la requête en cours.
send("chaine")	null pour une commande GET.



## 3.4.6.6 - Exemples



**3.4.6.6.1 - Exemple : Lecture Fichier**

**3.4.6.6.2 - Exemple : Récupérer la date d'un fichier**

**3.4.6.6.3 - Exemple : Test si la page existe**

**3.4.6.6.4 - Exemple : Remplir une forme avec d'un document XML**

**3.4.6.6.5 - Exemple : Appel de code PHP**

**3.4.6.6.6 - Bibliographie**





## 3.4.6.6.1a - Exemple : Lecture Fichier



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" >
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <link rel="stylesheet" media="screen" type="text/css" title="ExempleCSS" href="css/design.css" />
    <script type="text/javascript" src="js/jstest.js"> </script>
    <title>Web 2.0 : Ajax - HTTPRequestAjax Demo </title>
  </head>
  <body>
    <h1> HTTPRequestAjax : Lecture fichier</h1>
    <hr/>
    Lecture du fichier <b>test-01.txt</b> et ecriture du contenu dans la page
    <form name="ajax" method="post" action="">
      <p>
        <input type="BUTTON" value="Cliquer" onclick="actionClick()" />
      </p>
    </form>
    <div id="espace">
    </div>
    <p>&nbsp;</p>
  </body>
</html>
```



## 3.4.6.6.1b - Exemple : Lecture Fichier



```
function createXHR() {  
    var request = false;  
    try {  
        request = new ActiveXObject('Msxml2.XMLHTTP');  
    } catch (err2) {  
        try {  
            request = new ActiveXObject('Microsoft.XMLHTTP');  
        } catch (err3) {  
            try {  
                request = new XMLHttpRequest();  
            } catch (err1) {  
                alert("Votre navigateur ne supporte pas l'objet XMLHttpRequest");  
                request = false;  
            }  
        }  
    }  
    return request;  
}
```

```
function actionClick() {  
    var xhr=createXHR();  
    xhr.open("GET", "test-01.txt",true);  
    document.getElementById("espace").innerHTML="Demarrage ...";  
  
    xhr.onreadystatechange=function() {  
        document.getElementById("espace").innerHTML=document.getElementById(  
            "espace").innerHTML+ "<br/>Serveur en attente ..." + xhr.readyState;  
        if(xhr.readyState == 4) {  
            document.getElementById("espace").innerHTML=  
                document.getElementById("espace").innerHTML+  
                    "<br/>" + xhr.responseText;  
        }  
    };  
    xhr.send(null);  
}
```



## 3.4.6.6.1d - Exemple : Lecture Fichier



### HttpRequestAjax : Lecture fichier

Lecture du fichier **test-01.txt** et ecriture du contenu dans la page

Cliquer

### HttpRequestAjax : Lecture fichier

Lecture du fichier **test-01.txt** et ecriture du contenu dans la page

Cliquer

Demarrage ...  
Serveur en attente ...1  
Serveur en attente ...2  
Serveur en attente ...3  
Serveur en attente ...4  
je suis le contenu lue dans le fichier



## 3.4.6.6.2a - Exemple : Récupérer la date d'un fichier



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" dir="ltr" lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <link rel="stylesheet" media="screen" type="text/css" title="ExempleCSS" href="css/design.css" />
    <script type="text/javascript" src="js/jstest.js"> </script>
    <title>Web 2.0 : Ajax - XMLHttpRequestAjax Demo </title>
  </head>
  <body>
    <h1>XMLHttpRequestAjax : Recuperer la date d'un fichier</h1>
    <hr/>
    Retourne la date de modification du fichier ...
    <form name="ajax" method="post" action="">
      <p>
        <input type="BUTTON" value="Cliquer" onclick="actionClick_02(document.ajax.nomfichier.value)"/>
        <input type="text" name="nomfichier" value="Figure26.html" maxlength="40"/>
      </p>
    </form>
    <div id="espace">
    </div>
  </body>
</html>
```



### 3.4.6.6.2b - Exemple : Récupérer la date d'un fichier



```
function actionClick_02(url) {  
    xhr=createXHR();  
    document.getElementById("espace").innerHTML="Demarrage ...";  
    xhr.open("HEAD", url, true);  
    xhr.onreadystatechange=function() {  
        var value="<br/>Serveur en attente ..." + xhr.readyState;  
        if (xhr.readyState == 3) {  
            value = value + "<dd>=> Toutes les informations : " + xhr.getAllResponseHeaders() + "</dd>";  
            value = value + "<dd>=> Taille: " + xhr.getResponseHeader('Content-Length') + "</dd>";  
            value = value + "<dd>=> Type: " + xhr.getResponseHeader('Content-Type') + "</dd>";  
            value = value + "<dd>=> Server: " + xhr.getResponseHeader('Server') + "</dd>"; }  
        if(xhr.readyState == 4) {  
            if (xhr.status==200) {  
                value = value + "<br/>Date: " + xhr.getResponseHeader("Date");  
                value = value + "<br/>Taille : " + xhr.getResponseHeader("Content-Length");  
            } else {  
                if(xhr.status==404) value = value + "<br/>" + url + " n'existe pas !";  
                else value = value + "<br/>" + "Erreur, status est " + xhr.status;  
            }  
        }  
        document.getElementById("espace").innerHTML=document.getElementById("espace").innerHTML+value;  
    }  
    xhr.send(null);  
}
```



## 3.4.6.6.2c - Exemple : Récupérer la date d'un fichier

### HTTPRequestAjax : Recuperer la date d'un fichier

Retourne la date de modification du fichier ...

### HTTPRequestAjax : Recuperer la date d'un fichier

Retourne la date de modification du fichier ...

Demarrage ...

Serveur en attente ...1

Serveur en attente ...2

Serveur en attente ...3

=> Toutes les informations : Date: Fri, 26 Jan 2007 11:53:14 GMT Server: Apache/2.0.59

(Win32) PHP/5.2.0 Last-Modified: Fri, 26 Jan 2007 10:44:51 GMT Etag:

"3c4da-3a2-37b0aaab" Accept-Ranges: bytes Content-Length: 930 Content-Type: text/html

=> Taille: 930

=> Type: text/html

=> Server: Apache/2.0.59 (Win32) PHP/5.2.0

Serveur en attente ...4

Date: Fri, 26 Jan 2007 11:53:14 GMT

Taille : 930



## 3.4.6.6.3a - Exemple : Test si la page existe



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" dir="ltr" lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <link rel="stylesheet" media="screen" type="text/css" title="ExempleCSS" href="css/design.css" />
    <script type="text/javascript" src="js/jstest.js"> </script>
    <title>Web 2.0 : Ajax - HTTPRequestAjax Demo </title>
  </head>
  <body>
    <h1>HTTPRequestAjax : Test si la page existe</h1>
    <hr/>Lit l'entete d'un fichier et rend une erreur si le fichier n'existe pas...
    <form name="ajax" method="post" action="">
      <p>
        <input type="BUTTON" value="Check" onclick="actionClick_03(document.ajax.filename.value)"/>
        <input type="text" name="filename" value="not-exists.html" maxlength="40"/>
      </p>
    </form>
    <div id="espace">
    </div>
  </body>
</html>
```



## 3.4.6.6.3b - Exemple : Test si la page existe



```
function actionClick_03(url) {  
    xhr=createXHR();  
    document.getElementById("espace").innerHTML="Demarrage ...";  
    xhr.open("HEAD", url, true);  
    xhr.onreadystatechange=function() {  
        var value ="<br/>Serveur en attente ..." + xhr.readyState;  
        if( xhr.readyState == 4) {  
            if (xhr.status == 200) {  
                value = value+"<br/>" + url + " existe ...";  
            } else {  
                if( xhr.status==404 )  
                    value = value+"<br/>" + url + " doesn't exist!";  
                else  
                    value = value+"<br/>Error, status is " + xhr.status;  
            }  
        }  
        document.getElementById("espace").innerHTML=document.getElementById("espace").innerHTML +  
        value;  
    }  
    xhr.send(null);  
}
```

## 3.4.6.6.3c - Exemple : Test si la page existe

### HttpRequestAjax : Test si la page existe

Lit l'entete d'un fichier et rend une erreur si le fichier n'existe pas...

### HttpRequestAjax : Test si la page existe

Lit l'entete d'un fichier et rend une erreur si le fichier n'existe pas...

Demarrage ...  
 Serveur en attente ...1  
 Serveur en attente ...2  
 Serveur en attente ...3  
 Serveur en attente ...4  
 Figure27.html existe ...

### HttpRequestAjax : Test si la page existe

Lit l'entete d'un fichier et rend une erreur si le fichier n'existe pas...

Demarrage ...  
 Serveur en attente ...1  
 Serveur en attente ...2  
 Serveur en attente ...4  
 not-exists.html n'existe pas!



### 3.4.6.6.4a - Exemple : Remplir une forme avec d'un document XML



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"
>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <link rel="stylesheet" media="screen" type="text/css" title="ExempleCSS" href="css/design.css" />
    <script type="text/javascript" src="js/jstest.js"> </script>
    <title>Web 2.0 : Ajax - XMLHttpRequestAjax Demo </title>
  </head>
  <body>
    <h1>XMLHttpRequest Ajax : Remplir une forme avec d'un document XML </h1>
    <hr/>Lit le fichier <b><a href="donnees.xml" target="_parent">donnees.xml</a></b> et affiche les champs
    correspondants.
    <form name="ajax" method="post" action="">
      <p> <input type="BUTTON" value="Submit" onclick=" actionClick_04()"/> </p>
      <p> <input type="text" name="one" size="32" value=""/></p>
      <p> <input type="text" name="two"/></p>
      <p><input type="text" name="three"/></p>
    </form>
    <div id="espace">
    </div>
  </body>
</html>
```



### 3.4.6.6.4b - Exemple : Remplir une forme avec d'un document XML



```
function actionClick_04() {  
    var xhr=createXHR();  
    xhr.open("GET", "donnees.xml",true);  
    document.getElementById("espace").innerHTML="Demarrage ...";  
    xhr.onreadystatechange = function() {  
        var value ="<br/>Serveur en attente ..." + xhr.readyState;  
        if(xhr.readyState == 4) {  
            if(xhr.status == 200) { LectureDonnees(xhr.responseXML);  
            } else {  
                alert("Erreur: Le status code est " + xhr.status + " " + xhr.statusText); }  
            value=value+"<br/>Terminer";  
        }  
        document.getElementById("espace").innerHTML=document.getElementById("espace").innerHTML+value;  
    };  
    xhr.send(null);  
}
```





### 3.4.6.6.4c - Exemple : Remplir une forme avec d'un document XML



```
function LectureDonnees(doc)
{
  try {
    document.ajax.one.value = doc.getElementsByTagName('e1').item(0).firstChild.data + "&" +
      doc.getElementsByTagName('e1').item(1).firstChild.data;
    document.ajax.two.value = doc.getElementsByTagName('e2').item(0).firstChild.data;
    document.ajax.three.value= doc.getElementsByTagName('e3').item(0).firstChild.data;
  }
  catch (err) {
    alert("Erreur : Fichier non conforme");
  }
}
```



### 3.4.6.6.4d - Exemple : Remplir une forme avec d'un document XML



#### HTTPRequest Ajax : Remplir une forme avec d'un document XML

lit le fichier [donnees.xml](#) et affiche les champs correspondants

Ce fichier XML ne semble pas avoir d'information de style lui  
L'arbre du document est montré ci-dessous.

```
- <liste>
  - <liste_el>
    <el> donnee 1.1 </el>
    <el> donnee 1.2 </el>
  </liste_el>
  <e2> donnee 2 </e2>
  <e3> donnee 3 </e3>
</liste>
```

#### HTTPRequest Ajax : Remplir une forme avec d'un document XML

lit le fichier [donnees.xml](#) et affiche les champs correspondants

donnee 1.1 & donnee 1.2

donnee 2

donnee 3

Demarrage ...  
Serveur en attente ...1  
Serveur en attente ...2  
Serveur en attente ...3  
Serveur en attente ...4  
Terminer



## 3.4.6.6.5a - Exemple : Appelle de code PHP



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" >
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <link rel="stylesheet" media="screen" type="text/css" title="ExempleCSS" href="css/design.css" />
    <script type="text/javascript" src="js/jstest.js"> </script>
    <title>Web 2.0 : Ajax - HTTPRequestAjax Demo </title>
  </head>
  <body>
    <h1>HTTPRequest Ajax : appelle de code PHP </h1>
    <div id="PHP">
      <form name="xmlfile" action="" method="post">
        Nom fichier: <input type="text" name="nomFichier" value="dataphp.xml" />
        <p> Champ 1 <input type="text" name="e1" size="32" value="valeur 1"/> </p>
        <p> Champ 2 <input type="text" name="e2" size="32" value="valeur 2"/> </p>
        <p> Champ 3 <input type="text" name="e3" size="32" value="valeur 3"/> </p>
        Creation Fichier <input type="BUTTON" value="Rentrer Nom" onclick=" actionClick_05()"/>
      </form>
    </div>
    <hr/> <div id="lirefichier"> En attente du fichier rentrer si-dessus </div>
    <form name="ajax" method="post" action="">
      <p> <input type="BUTTON" value="Submit" onclick=" actionClick_06()"/> </p>
      <p> <input type="text" name="one" size="32" value=""/></p>
      <p> <input type="text" name="two" size="32" value=""/> </p>
      <p> <input type="text" name="three" size="32" value=""/></p>
    </form>
    <hr/>Espace de travail...
    <div id="espace"> </div>
  </body>
</html>
```



## 3.4.6.6.5b - Exemple : Appelle de code PHP



```
function actionClick_05()
{
    var xhr=createXHR();
    xhr.open("POST", "essai.php",true);
    document.getElementById("espace").innerHTML="Demarrage ...";
    xhr.onreadystatechange = function() {
        var value ="<br/>Serveur en attente ..." + xhr.readyState;
        if(xhr.readyState == 4)
        {
            document.getElementById("espace").innerHTML=document.getElementById("espace").innerHTML+document.getEle
            mentById("espace").reponsText;
            document.getElementById("espace").innerHTML=document.getElementById("espace").innerHTML+
            "<br/><b><a href=\""+ document.xmlfile.nomFichier.value+"\"
            target=\"_parent\">"+document.xmlfile.nomFichier.value+"</a></b>";
            document.getElementById("lirefichier").innerHTML="lit le fichier <b><a href=\""+document.xmlfile.nomFichier.value+
            "\" target=\"_parent\">"+document.xmlfile.nomFichier.value+"</a></b> et affiche les champs correspondants.";
        }
        document.getElementById("espace").innerHTML=document.getElementById("espace").innerHTML+value;
    };
    xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

    xhr.send("nomFichier="+document.xmlfile.nomFichier.value+"&Champ1="+document.xmlfile.e1.value+"&Champ2="+d
    ocument.xmlfile.e2.value+"&Champ3="+document.xmlfile.e3.value);
}
```



## 3.4.6.6.5c - Exemple : Appelle de code PHP



```
<?php
function writeMyName($fileName,$champ1,$champ2,$champ3)
{
    @unlink($fileName);
    $str = @file_get_contents($fileName);
    $fd = @fopen($fileName, "w+");
    $str."<Liste>\n<Champ1>";
    $str.=$champ1;
    $str."</Champ1>\n";
    $str."<Champ2>";
    $str.=$champ2;
    $str."</Champ2>\n";
    $str."<Champ3>";
    $str.=$champ3;
    $str."</Champ3>\n</Liste>";
    @fwrite($fd, $str);
    @fclose($fd);
}
writeMyName($_POST["nomFichier"],$_POST["Champ1"],$_POST["Champ2"],$_POST["Champ3"]);
?>
```



## 3.4.6.6.5d - Exemple : Appelle de code PHP



```
function actionClick_06() {
    var xhr=createXHR();
    xhr.open("GET",document.xmlfile.nomFichier.value,true);
    document.getElementById("espace").innerHTML="Demarrage ...";
    xhr.onreadystatechange = function() {
        var value ="<br/>Serveur en attente ..." + xhr.readyState;
        if(xhr.readyState == 4) {
            if(xhr.status == 200) { LectureXML(xhr.responseXML); } else {
                alert("Erreur: Le status code est " + xhr.status + " " + xhr.statusText); }
            value=value+"<br/>Terminer";
        }
        document.getElementById("espace").innerHTML=document.getElementById("espace").innerHTML+value;
    };
    xhr.send(null);
}

function LectureXML(doc) {
    try {
        document.ajax.one.value = doc.getElementsByTagName('Champ1').item(0).firstChild.data;
        document.ajax.two.value = doc.getElementsByTagName('Champ2').item(0).firstChild.data;
        document.ajax.three.value= doc.getElementsByTagName('Champ3').item(0).firstChild.data;
    } catch (err) { alert("Erreur : Fichier non conforme"); }
}
```





## 3.4.6.6.5d - Exemple : Appelle de code PHP



### HTTPRequest Ajax : appelle de code PHP

Nom fichier:

Champ 1

Champ 2

Champ 3

Creation Fichier

---

En attente du fichier rentrer si-dessus

---

Espace de travail ...



## 3.4.6.6.5f - Exemple : Appelle de code PHP

### HTTPRequest Ajax : appelle de code PHP

Nom fichier:

Champ 1

Champ 2

Champ 3

Creation Fichier

lit le fichier [dataphp.xml](#) et affiche les champs correspondant.

Espace de travail ...

Demarrage ...

Serveur en attente ...1

Serveur en attente ...2undefined

[dataphp.xml](#)

Serveur en attente ...4

Ce fichier XML ne semble pas avoir d'information de style lui étant associé. L'arbre du document est montré ci-dessous.

```
- <Liste>
  <Champ1>valeur 1</Champ1>
  <Champ2>valeur 2</Champ2>
  <Champ3>valeur 3</Champ3>
</Liste>
```

### HTTPRequest Ajax : appelle de code PHP

Nom fichier:

Champ 1

Champ 2

Champ 3

Creation Fichier

---

lit le fichier [dataphp.xml](#) et affiche les champs correspondant.

---

Espace de travail ...

Demarrage ...

Serveur en attente ...1

Serveur en attente ...2

Serveur en attente ...3

Serveur en attente ...4

Terminer



### ➤ Voir Bibliographie Javascript

#### ● 3.4.5.19 - Bibliographie

### ➤ Liens divers

● <http://www.toutjavascript.com/>

● <http://www.phpfrance.com/>



### **3.4.7.1 - Définition**

### **3.4.7.2 - Métadonnées**

### **3.4.7.3 - Installation Mozilla/Firefox de Sage**

### **3.4.7.4 - Exemple Fichier RSS**

### **3.4.7.5 - Bibliographie**



Un **flux RSS** ou **fil RSS** ("RSS feed" en anglais), sigle de **Really Simple Syndication** (*souscription vraiment simple*), ou de **Rich Site Summary** (Sommaire d'un site enrichi) est un format de syndication de contenu Web, codé sous forme XML.

Ce système permet de diffuser en temps réel les nouvelles des sites d'information ou des blogs, ce qui permet de rapidement consulter ces dernières sans visiter le site.





## 3.4.7.2a - Métadonnées

Liste de tous les éléments pouvant se trouver dans la balise <channel>	
Balise	Description
title	Titre du channel
link	URL du site contenant le channel
description	Description du channel
language	Langue du channel
copyright	Info sur le copyright du channel
managingEditor	Mail de la personne responsable du contenu
webMaster	Mail du webmaster
pubDate	Date de publication
lastBuildDate	Date de la dernière publication
category	Catégorie à laquelle le channel appartient
generator	Programme utilisé pour générer le channel
docs	Lien vers la documentation du format utilisé dans le fichier RSS
cloud	Permet à un programme de s'enregistrer pour être notifié des modifications de ce channel
ttl	Time to live, avant le prochain rafraîchissement
image	Image affichée avec le channel
rating	note PICS
textInput	Ajouter une zone de saisie de texte
skipHours	Heures que les agrégateurs peuvent ignorer
skipDays	Jours que les agrégateurs peuvent ignorer



## 3.4.7.2b - Métadonnées



### Liste de tous les éléments pouvant se trouver dans la balise <item>

Balise	Description
title	Titre de l'item
link	URL de l'item
description	Description de l'item
author	Mail de l'auteur de l'item
category	Catégorie à laquelle l'item appartient
comments	Lien vers une page de commentaires sur l'item
enclosure	Objet media attaché à l'item
guid	Texte qui identifie de manière unique cet item
pubDate	Date de publication
source	Channel auquel l'item appartient



### 3.4.7.3 - Installation MozillaFirefox de Sage



Sage, le lecteur de flux par Mozilla/Firefox

➤ <http://sage.mozdev.org/install>



## 3.4.7.4a - Exemple Fichier RSS



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="xsl/rss.xsl"?>
<rss version="2.0">
  <channel>
    <title>Cours Web 2.0</title>
    <description>Les Flux RSS</description>
    <pubDate>Wed, 31 Jul 2007 00:30:30 -0700</pubDate>
    <link>www.yantra-technologies.com</link>
    <image>./images/P3082320.JPG</image>
    <language>fr-fr</language>
    <copyright>yantra-technologies</copyright>
    <ttl>60</ttl>
    <item>
      <title>AJAX Actualité</title>
      <description>Actualite Ajax</description>
      <pubDate>Wed, 25 Jul 2007 </pubDate>
      <image>./images/P3102375.gif</image>
      <link href="www.yantra-technologies.com/Actualites.html">Actualites.html</link>
    </item>
    <item>
      <title>RSS Actualité</title>
      <description>Actualite RSS</description>
      <pubDate>Tue, 19 Jul 2007 04:32:51 -0700</pubDate>
      <link>Actualites.html</link>
    </item>
  </channel>
</rss>
```



## 3.4.7.4b - Exemple Fichier RSS



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html> <head>
      <title> Pages Actualite Cours Web 2.0 </title>
      <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
      <link rel="stylesheet" media="screen" type="text/css" title="ExempleCSS" href="http://localhost:81/www/css/design.css" />
    </head>
    <body id="CorpsPages">
      <h1 id="Titre1" >Pages Actualite du Cours Web 2.0 </h1>
      Titre : <xsl:value-of select="//rss/channel/title"/>
      <blockquote>
        <table class="Tableau" border="2">
          <caption>Tableau des Informations</caption>
          <thead>
            <tr><th>Titre</th><th>Description</th><th>Image</th><th>liens</th><th>date</th></tr>
            <xsl:for-each select="//rss/channel/item">
              <tr> <th><code><xsl:value-of select="title"/></code></th>
                <th><code><xsl:value-of select="description"/></code></th>
                <th><code><xsl:value-of select="image"/></code></th>
                <th><a href="http://{link/@href}"> <code><xsl:value-of select="link"/></code> </a></th>
                <th> <code><xsl:value-of select="pubDate"/></code></th>
              </tr>
            </xsl:for-each>
          </thead>
          <tfoot><tr> <th colspan="5"><small><a href="http://www.w3schools.com">W3Schools </a></small></th></tr> </tfoot>
          <tbody>
          </tbody>
        </table>
      </blockquote>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

### Pages Actualite du Cours Web 2.0

Titre : Cours Web 2.0

Tableau des Informations				
Titre	Description	Image	liens	date
<b>AJAX</b> Actualité	Actualite Ajax	<a href="#">./images/P3102375.gif</a>	<a href="#">Actualites.html</a>	Wed, 25 Jul 2007
<b>RSS</b> Actualité	Actualite RSS		<a href="#">Actualites.html</a>	Tue, 19 Jul 2007 04:32:51 -0700
<a href="#">W3Schools</a>				

#### Cours Web 2.0

Les Flux RSS

##### 1. AJAX Actualité

Actualite Ajax

Mer, Jul 25, 2007 0:00



##### 2. RSS Actualité

Actualite RSS

Jeu, Jul 19, 2007 13:32





## 3.4.7.5 - Bibliographie



### ➤ Liens divers sur RSS

- <http://sebsauvage.net/comprendre/rss/creer.html>
- <http://www.elanceur.org/Articles/CommentdemarreravecRSS.html>
- <http://ghostdogpr.developpez.com/articles/rss/>
- <http://delfiweb.developpez.com/flash/sources/lecteur-flux-rss/>

### ➤ Moteur de recherche

- <http://www.lamooche.com/>





## 3.4.8 - Perspectives



### Enjeux du WEB => L 'interface utilisateur

- AJAX est une expression de ce phénomène présage l'évolution des interfaces Web
- AJAX rend possible un GUI (Graphic User Interface, ) riche et dynamique basé sur JavaScript (XMLHttpRequest) et permet l'intégration de données sans recharger entièrement la page

### Problèmes AJAX

- Ne résoud pas la compatibilité avec différents navigateurs
- Pauvreté des WIDGET ne satisfait pas l'utilisateur final.
- Apparition d'autre interfaces riches, basées sur le futur XAML (fondement d'Avalon, l'interface graphique de Microsoft Longhorn) ou sur XUL (Mozilla fondation) commencent à voir le jour.

### L'avenir d'Ajax

- IBM a créé Open AJAX Initiative, un groupe de promotion de cette technologie avec des partenaires tels que BEA, Borland, the Dojo Foundation, Eclipse Foundation, Google, Ilog, Yahoo!, Laszlo Systems, Mozilla Corporation, Novell, Openwave Systems, SAP, Oracle, Red Hat, Zend et Zimbra.
- Création l'AJAX Toolkit Framework, un projet qui vise à proposer des outils pour le développement d'applications AJAX dans l'outil de développement Eclipse.
- La fondation Eclipse propose également un outil pour le développement d'applications web basées sur le framework Eclipse, sous le nom de RAP (Rich AJAX Platform).