

Ressource R209

Initiation au développement Web

HTTP, HTML, DOM, JavaScript



IUT de Béziers, dépt. R&T © 2022

<http://www.borelly.net/>

Christophe.BORELLY@umontpellier.fr

Contenus de la ressource

- Introduction au protocole HTTP.
- Mise en forme de pages Web :
 - balises HTML avancées ;
 - structure d'une page avec son DOM ;
 - CSS avancé ou Framework ;
 - initiation au dynamisme côté client (JavaScript, bibliothèques comme jQuery)
- Scripts côté serveur.
 - Éléments d'interaction client-serveur (requête HTTP, URL, formulaire).
 - Interrogation d'un SGBD ou d'une API.
 - Sensibilisation à la sécurisation de sites : failles XSS, XSS stockée, injections SQL.
- **Cours 1+1 (1h15), TD 5 (1h15), TP 5 (2h45)**

HTTP

- **HTTP** : Hypertext Transfert Protocol
- Port TCP 80 par défaut (TCP 443 pour HTTPS).
- Méthodes: **GET**, **POST**, PUT, ... (RFC 7231)
- Format d'une requête :

METHODE URL VERSION<CR><LF>

Champs d'entête (optionnels)<CR><LF>
<CR><LF>

Corps de message (option)

- Format d'une réponse :

VERSION CODE STATUS<CR><LF>

Champs d'entête (optionnels)<CR><LF>
<CR><LF>

Corps de message (option)

Le champ **Host** est obligatoire en HTTP/1.1 !

Example

GET /cb/cours/index.php?rc=true&search=R109 HTTP/1.1

Host: www.borelly.net

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:49.0) Gecko/20100101 Firefox/49.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: http://www.borelly.net/cb/

Cookie: PHPSESSID=lq1m96uqge63pbk396atk3tns6

Connection: keep-alive

Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK

Date: Tue, 27 Sep 2016 08:26:31 GMT

Server: Apache/2.4.23 (Unix) OpenSSL/1.0.2i PHP/5.6.26

X-Powered-By: PHP/5.6.26

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

Pragma: no-cache

Content-Length: 7684

Keep-Alive: timeout=5, max=100

Connection: Keep-Alive

Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>

<html>

...

Codes HTTP

- Le code d'erreur est un nombre de 3 digits avec 5 catégories (RFC 7231 page 47) :
 - **1xx** : Information, requête reçue en cours de traitement.
 - **2xx** : Succès, requête reçue, comprise et acceptée.
 - **3xx** : Redirection, une action supplémentaire est nécessaire pour traiter la requête.
 - **4xx** : Erreur du client, mauvaise syntaxe ou requête interrompue.
 - **5xx** : Erreur du serveur sur une requête qui semble valide.

HTML avancé

- Interactions avec l'utilisateur :
 - Formulaires : form, input (text, password, radio, checkbox, ...), textarea, select, button
- Style CSS : attribut **class**
- Identification unique des balises : attribut **id**
- Événements : onclick, onmouseover, onload, onchange, onselect, ...

Exemple

```
<form action="auth.php" method="get">
Nom :<input type="text" name="nom" value=""/><br/>
Password :<input type="password" name="pwd"
value="titi"/><br/>

Homme :<input type="radio" name="c" value="homme" checked/>
Femme :<input type="radio" name="c" value="femme"/><br/>

10 :<input type="checkbox" name="choice" value="10" checked/>
20 :<input type="checkbox" name="choice" value="20"/>
30 :<input type="checkbox" name="choice" value="30"/><br/>

Pays :<select name="pays">
<option value="1">angleterre</option>
<option value="2" selected>france</option>
<option value="3">suisse</option>
</select>

<input type="hidden" name="cache" value="secret"/>
<input type="submit" value="ok"/>
</form>
```

Nom :

Password :

Homme : ☒ Femme : ☐

10 : ☒ 20 : ☐ 30 : ☐

Pays :

france

angleterre

france

suisse

Javascript

- JavaScript est un langage interprété (coté client) supportant quelques fonctionnalités rudimentaires de programmation objet.
- Il a été « embarqué » dans les navigateurs WEB pour améliorer le rendu des pages HTML.
- La syntaxe générale ressemble à celle du C/C++ et du JAVA (bien sur).
- Défini par l'ECMA (European Computer Manufacturers Association) : Standard **ECMA-262**
- La version ES7 date de 2016 (ES12-2021)

Utilisation de javascript (1)

- On utilise la balise **<script>**
- On peut soit directement taper le code javascript :

```
<script>  
alert("Hello world !");  
</script>
```
- Ou bien donner une URL avec l'attribut **src** :

```
<script src="calcul.js"></script>
```

Utilisation de javascript (2)

- On peut soit écrire directement du code ou un appel à une fonction javascript dans certains attributs des balises HTML.

```
<p onmouseover="alert('salut');">...</p>  
<button onclick="f3();" />OK</button>
```

- La fonction **f3()** doit bien entendu avoir été définie auparavant.
- Attributs possibles : **onclick**, **ondblclick**, **onmousedown**, **onmouseup**, **onmouseover**, **onmousemove**, **onmouseout**, **onwheel**, **onkeypress**, **onkeydown**, **onkeyup**, ...

Déclaration des variables

- Le mot clé **var** permet de déclarer une variable globale non typée :

```
var i;  
var sum = 3.14;  
var j, k, l;  
var z = 2;  
z = "toto";  
k = 'titi';
```

- On peut aussi déclarer des variables locales avec le mot clé **let** ou **const**.

Les chaînes

- Une chaîne se définit avec des " ou bien des '
- L'opérateur + permet de concaténer 2 chaînes

```
var x="toto";  
x=x+"titi"; // x vaut « tototiti »
```

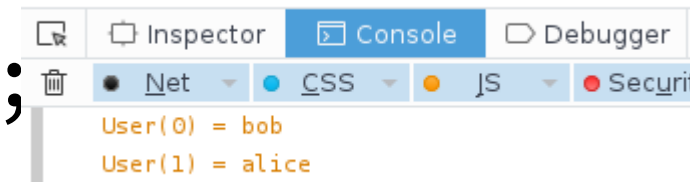
- La propriété length donne la longueur (nombre de caractères)

```
var y=x.length;
```

Éléments de programmation

- Javascript utilise les mêmes éléments de programmation que le C ou le JAVA :
 - `if/else`, `switch/case`, `for`, `while`, `do/while`, avec le support des mots clés `break` et `continue`.
- Une notation de boucle simplifiée existe :

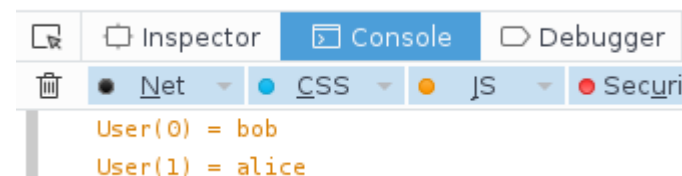
```
var tab=["bob","alice"];  
for (k in tab) {  
    console.log("User("+k+") = "+tab[k]);  
}
```



Méthodes pour les tableaux

- On peut définir une fonction à appliquer aux éléments d'un tableau :

```
var tab=["bob","alice"];  
tab.forEach(function(val,i) {  
    console.log("User("+i+") = "+val);  
});
```



- Il existe également :

push(), pop(), shift(), sort(),
reverse(), ...

Opérateurs

- Utilise tous les opérateurs du C et du JAVA

+ - * / % & | ^ ~ << >> && || ! ++ -- += -= *= /= ...

- Test d'égalité == (compare les valeurs)
- Test complet === (compare aussi le type)
- Inégalité != ou !==
- Mot clés **null** (pas de valeur) et **undefined** (variable non définie)

Les fonctions

- On peut créer une fonction javascript avec le mot clé **function**.

```
function calcul(a,b) {return a+b;}
```

- On peut préciser les arguments ou utiliser le tableau **arguments**.

```
function test() {  
    var i;  
    for (i=0;i<arguments.length;i++){  
        console.log("arg("+i+")="+arguments[i]);  
    }  
}
```


Boîtes de dialogue

- Note d'information (OK)

```
alert("texte");
```

- Boîte de confirmation (OK, Cancel)

```
confirm("texte"); // renvoie true si OK
```

- Boîte de lecture (OK, Cancel)

```
prompt("texte", "defaultValue");  
// renvoie la valeur entrée si OK
```

Objets de base

- JavaScript supporte un grand nombre d'objets JAVA (Array, Boolean, Date, Math, String, Object, etc...). Voir la documentation !
- On peut créer un objet avec `{}` :

```
var obj={"prenom":"john","nom":"DOE","age":42};
obj.prenom="John";
obj.age+=1;
obj.calcul=function() { // Création d'une méthode
    return this.age-10;
};
console.log("Age : "+obj.calcul()); // Age : 33
```

Objets du navigateur

- En javaScript, on peut accéder directement à un certain nombre d'objets représentant les éléments du navigateur :

Navigator : language, userAgent, ...

Window : innerHeight, innerWidth, screenX, screenY, pageXoffset, pageYoffset, ...

Document: voir suite...

Screen : height, width, availHeight, availWidth, ...

History : length, back(), forward(), go()

Location : protocol, host, port, pathname, href, ...

Examples

navigator.language : en-US

navigator.userAgent : Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0

window.innerWidth x **window.innerHeight** : 1600x516

screen.width x **screen.height** : 1600x900

history.len : undefined

location.href : file:///home/cb/R209-WEB/nav.html

DOM

- Le W3C à défini en 1997 la norme **DOM** (Document Object Model) qui permet de manipuler et modifier les éléments d'un document XML.
- Il s'agit d'une hiérarchie dont la base est l'objet de type **Node** : NodeList, Document, Element, Attr, Text, ...

Propriétés et Méthodes

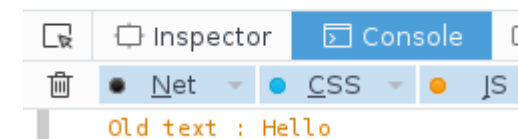
- **Node** : nodeName, nodeValue, parentNode, childNodes, firstChild, lastChild, previousSibling, nextSibling, appendChild(), insertBefore(), replaceChild(), removeChild(), hasAttributes(), hasChildNodes(), ...
- **NodeList** : length, item()
- **Document** : createAttribute(), createElement(), createTextNode(), createComment(), **getElementById()**, getElementsByName(), getElementsByTagName(), querySelector(), querySelectorAll(), ...
- **Element :: Node** : tagName, **innerHTML**, **getAttribute()**, hasAttribute(), **setAttribute()**, removeAttribute(), ...
- **Attr :: Node** : name, **value**, ...

Example

```
<style>.warn {color:red;}</style>
<p class="" id="p1">Hello</p>
<button onclick="change();">Change Text</button>
<script>
var text;
function change() {
    var p=document.getElementById('p1');
    if (p.hasAttribute("class")) {
        p.setAttribute("class","warn");
    }
    text=p.innerHTML;
    console.log("Old text : "+text);
    p.innerHTML="Salut";
}
</script>
```

Salut

Change Text



Gestion des événements

```
<button id="ok">OK</button>
<script>
var i=0;
var btn=document.getElementById("ok");
btn.addEventListener('click',click1);
function click1() {
    btn.innerHTML="click "+i;
    i++;
}
</script>
```


Références

- <http://www.w3schools.com/js/>
- <https://www.w3.org/DOM/Activity>
- <https://dom.spec.whatwg.org/>