Test

- 1. What does CSS stand for?
  - Colorful Style Sheets
  - Cascading Style Sheets
  - Computer Style Sheets
  - Creative Style Sheets

Test

- 2. What is the correct HTML for referring to an external style sheet?
  - <style src="mystyle.css">
  - <link rel="stylesheet" type="text/css" href="mystyle.css">
  - <stylesheet>mystyle.css</stylesheet>

Test

- 3. Where in an HTML document is the correct place to refer to an external style sheet?
  - In the <head> section
  - At the top of the document
  - At the end of the document
  - In the <body> section

Test

- 4. Which HTML tag is used to define an internal style sheet?
  - <css>
  - <style>
  - <script>

Test

- 5. Which is the correct CSS syntax?
  - {body;color:black;}
  - body:color=black;
  - {body:color=black;}
  - body {color: black;}

Test

- 6.How do you insert a comment in a CSS file?
  - // this is a comment //
  - // this is a comment
  - /* this is a comment */
  - ' this is a comment

Test

- 7.Which property is used to change the background color?
  - background-color
  - color
  - bgcolor

Test

- How do you add a background color for all <h1> elements?
  - h1 {background-color:#FFFFFF;}
  - h1.all {background-color:#FFFFFF;}
  - all.h1 {background-color:#FFFFFF;}

Test

- Which CSS property is used to change the text color of an element?
  - color
  - text-color
  - fgcolor

Test

- 10. How do you display hyperlinks without an underline?
  - a {text-decoration:none;}
  - a {underline:none;}
  - a {decoration:no-underline;}
  - a {text-decoration:no-underline;}

Ismail HADJADJ
Log.ismail11@gmail.com

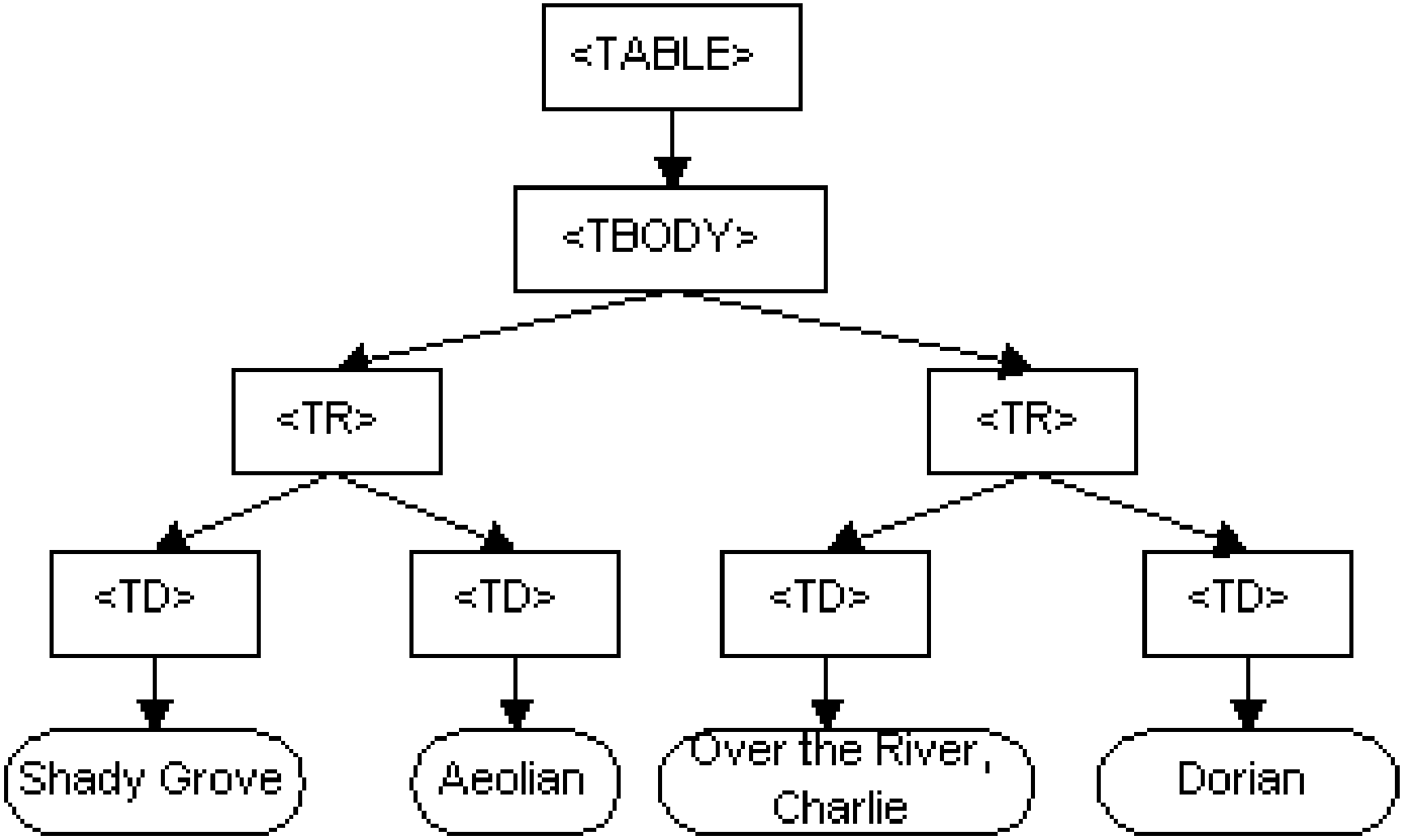
Programmation Web Coté Client : DOM / JavaScript

DOM = Document Object Model

- API (Application Programming Interface) pour la manipulation de HTML / XML
- Définit la structure logique des documents
- Définit la façon d'y accéder, de la manipuler

→ Créer des documents
→ Parcourir leur structure
→ Ajouter, effacer, modifier des éléments
→ Ajouter, effacer, modifier leur contenu

Qu'est-ce que le DOM ?



Qu'est-ce que le DOM ?

- Représentation arborescente du document
- Modèle objet (structure + méthodes)
- Permet la manipulation du document
- Une implémentation : JavaScript…
- … Des implémentations :
  - JavaScript IE
  - JavaScript Mozilla / Firefox
  - JavaScript Opera
  - …

JavaScript : Principe

- Langage de script objet
- Syntaxe style C / C++ / Java
- Sensible à la casse
- N'est PAS du Java
- Exécuté par le client Web
- Peut être désactivé sur le client
- Nombreux objets pour la manipulation HTML
- Gestion des événements HTML
- Rendre les pages dynamiques (HTML+CSS+JS)
- Haut niveau d'incompatibilité…

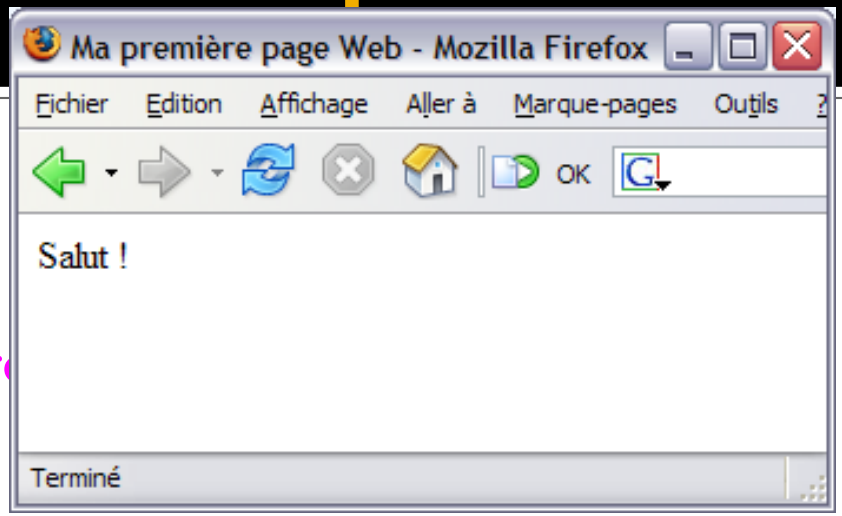
JavaScript : Balise script

```
<script type="text/javascript"
  language="JavaScript">
<!--
script
// -->
</script>

<script type="text/javascript"
  language="JavaScript" src="URI">
</script>
```

JavaScript : Exemple

```
<html>
 <head>
  <title>Ma première pag
 </head>
 <body>
 <script type="text/javascript"
  language="JavaScript">
  <!--
   document.writeln("Salut !") ;
  // -->
 </script>
</body>
</html>
```



Variables

- Déclaration de variables facultative
- Variables non typées à la déclaration

```
var nom_variable ;
```

- Typage dynamique à l'affectation
- Types gérés:
  - Nombres (`10, 3.14`)
  - Booléens (`true`, `false`)
  - Chaînes ("Salut !", 'Salut !')
  - `null`
  - `undefined`

Structures conditionnelles

```
if (condition)
{
  instructions ;
}
[ else
{
  instructions ;
} ]
```

Structures conditionnelles

```
switch (expression)
{
   case étiquette :
      instructions ;
      break ;
   case étiquette :
      instructions ;
      break ;
   default :
      instructions ;
}
```

Structures itératives

```
while (condition)
{
  instructions ;
}

do
{
  instructions ;
}
while (condition) ;
```

Structures itératives

```
for (instr ; condition ; instr)
{
  instructions ;
}

for (variable in objet)
{
  instructions ;
}
```

Commentaires

```
// Commentaire ligne

/* Commentaire multi-lignes */
```

Fonctions

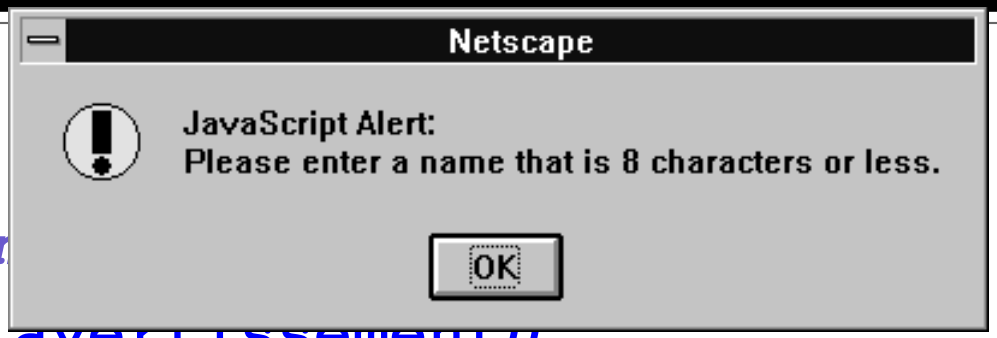
- Valeur de retour non typée
- Arguments non typés

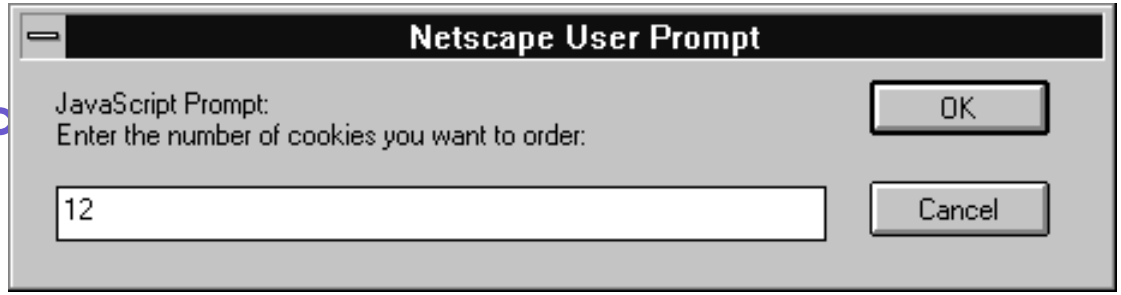
```
// Déclaration
function ma_fonction(arguments)
{
   instructions ;
   return quelque_chose; // ou pas…
}

ma_fonction(12) ; // Appel
```

Objets prédéfinis

- window
  - alert(m
Message d'avertissement//
  - confirm
Message de true//
ou false
  - promp
Boîte de
saisie





Objets prédéfinis

- **document**
  - **write(*message*)**

Ecrire dans le document//

  - **writeln(*message*)**

Ecrire dans le document (retour à// la ligne)

Chaînes : Propriétés & Méthodes

- Propriétés
  - **`length`**
  - …
- Méthodes
  - **`charAt(`*`index`*`)`**
  - **`charCodeAt(`*`index`*`)`**
  - **`concat(`*`chaine2`*`, `*`chaine3`*`, …)`**
  - **`fromCharCode(`*`code1`*`, `*`code2`*`, …)`**
  - **`indexOf(`*`aiguille`*`[, `*`index`*`])`**
  - **`lastIndexOf(`*`aiguille`*`[, `*`index`*`])`**
  - **`match(`*`expr_reg`*`)`**

Chaînes : Exemples

```
var s = "Bon anniversaire Benjamin" ;
document.write(s.charAt(2)) ;
```
➔ **n**
```
document.write(s.charCodeAt(2)) ;
```
➔ **110**
```
document.write(s.concat(" du groupe C12")) ;
```
➔ **Bon anniversaire Benjamin du groupe C12**
```
document.write(String.fromCharCode(49, 50)) ;
```
➔ **12**
```
document.write(s.indexOf("Benjamin")) ;
```
➔ **17**
```
document.write(s.lastIndexOf("a")) ;
```
➔ **21**
```
document.write(s.match(/Benjamin$/)) ;
```
➔ **Benjamin(*null si non trouvé*)**

Chaînes : Méthodes

- Méthodes
  - **replace(*expr_reg, nouvelle_chaine*)**
  - **search(*expr_reg*)**
  - **slice(*debut*[, *fin*])**
  - **split(*separateur*[, *limite*])**
  - **substr(*debut*[, *taille*])**
  - **substring(*debut, fin*)**
  - **toLowerCase()**
  - **toUpperCase()**
- Opérateurs
  - **+**

Chaînes : Exemples

```
var s = "Bon anniversaire Benjamin" ;
document.write(s.replace(/i/g, 'I')) ;
```
➔ Bon annIversaIre BenjamIn
```
document.write(s.search(/n{2}/i)) ;
```
➔ 5
```
document.write(s.slice(17)) ;
```
➔ Benjamin
```
document.write(s.split(" ")) ;
```
➔ Bon,anniversaire,Benjamin
```
document.write(s.substr(4, 12)) ;
```
➔ anniversaire
```
document.write(s.substring(4, 16)) ;
```
➔ anniversaire
```
document.write(s.toUpperCase()+s.toLowerCase()) ;
```
➔ BON ANNIVERSAIRE BENJAMINbon anniversaire benjamin

Objet Math

- Propriétés
  - **E, LN10, LN2, LOG10E, LOG2E, PI, SQRT1_2, SQRT2**
- Méthodes
  - **abs(*val*)**
  - **acos(*val*), cos(*val*), …**
  - **exp(*val*), log(*val*)**
  - **floor(*val*), round(*val*), ceil(*val*)**
  - **max(*val1*, *val2*), min(*val1*, *val2*)**
  - **pow(*val*, *puiss*), sqrt(*val*)**
  - **random() // 0 → 1**

Objet Math : Exemples

```
document.write(115.04+15) ;
➔ 130.04000000000002  (Euh ?...)
document.write(Math.PI) ;
➔ 3.141592653589793
document.write(Math.abs(-12.34)) ;
➔ 12.34
document.write(Math.floor(12.54)) ;
➔ 12
document.write(Math.round(12.54)) ;
➔ 13
document.write(Math.ceil(12.54)) ;
➔ 13
document.write(Math.random()) ;
➔ 0.394555831655689
```

Propriétés & Fonctions supérieures

- Propriétés
  - **`Infinity, NaN, undefined`**
- Fonctions
  - **`eval(`*`chaine`*`)`**
  - **`isFinite(`*`nombre`*`)`**
  - **`isNaN(`*`objet`*`)`**
  - **`parseFloat(`*`chaine`*`)`**
  - **`parseInt(`*`chaine`*`)`**
  - **`escape(`*`chaine`*`)`**
  - **`unescape(`*`chaine`*`)`**

Propriétés & Fonctions supérieures

```
document.write(eval("Math.pow(3+2, 2)")) ;
```
➔ 25
```
document.write(isFinite(Math.log(0))) ;
```
➔ false
```
document.write(isNaN("abcd")) ;
```
➔ true
```
document.write("12.34"+2) ;
```
➔ 12.342
```
document.write(parseFloat("12.34")+2) ;
```
➔ 14.34
```
document.write(escape("Bon anniversaire")) ;
```
➔ Bon%20anniversaire
```
document.write(unescape("Bon%20anniversaire")) ;
```
➔ Bon anniversaire

Tableaux

- Objet Array
- Déclaration

```
var tab1 = new Array(taille) ;
var tab2 = new Array(1, "a", 9, …) ;
index → 0   1   2   …
```

- Utilisation

```
window.alert(tab2[0]) ; // 1
tab2[2] = 6 // 6 remplace 9
```

- Accroissement automatique de la taille

```
var tab1 = new Array(2) ;
tab1[200] = 5 ;
```

Tableaux

- Parcours

```
var tab2 = new Array(1, "a", 9) ;
tab2[200] = 12 ;
for (i in tab2)
  window.alert("tab2[" + i + "] = "
  + tab2[i]) ;
// tab2[0] = 1
// tab2[1] = a
// tab2[2] = 9
// tab2[200] = 12
```

Tableaux : Propriétés & Méthodes

- Propriétés
  - **`length`**
  - …
- Méthodes
  - **`concat(`*`tab2`*`, `*`tab3`*`, …)`**
  - **`join(`*`sépar`*`)`**
  - **`pop()`**
  - **`push(`*`val1`*`, `*`val2`*`, …)`**
  - **`shift()`**
  - **`unshift(`*`val1`*`, `*`val2`*`, …)`**
  - **`slice(`*`début`*`[, `*`fin`*`])`**

Contrôle de formulaires

- Vérifier la cohérence de la saisie
- Contrôles sur le client
- Évite les transmissions client / serveur
- Contrôles possibles:
  - Présence de valeur
  - Numérique / Chaîne
  - Expressions régulières
- Événement **`onSubmit`**

Contrôle de formulaires

```
<html><head><title>Contrôle</title>
<script type="text/javascript">
function verif() {
  if (document.formu.txt.value != '')
     return window.confirm('Envoyer ?') ;
  return false ; }
</script></head><body>
<form name="formu" action="URI"
  method="GET" onSubmit="return verif() ;">
  <input type="text" name="txt">
  <input type="submit" value="Envoyer">
</form></body></html>
```

Formulaires : Propriétés & Méthodes

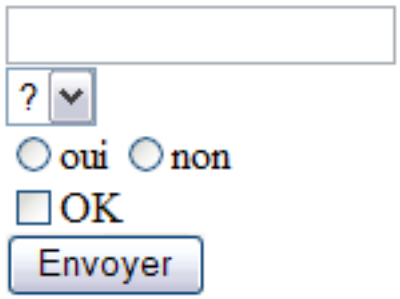
- Propriétés
  - `action`
  - `elements`
  - `encoding`
  - `length`
  - `method`
  - `name`
  - `target`
- Méthodes
  - `reset()`
  - `submit()`

Objets *commandes* de formulaires

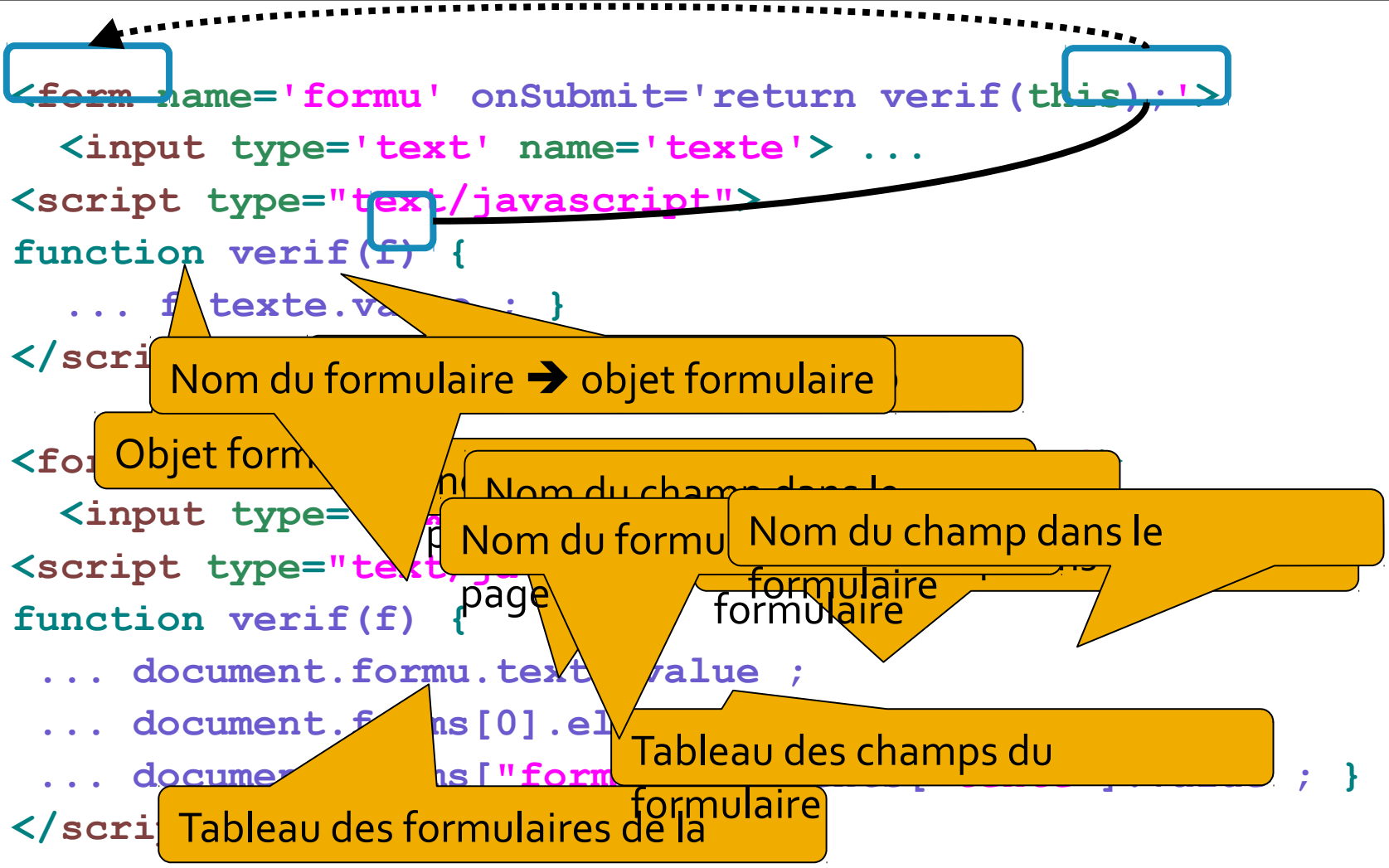
- **Text**
- **Textarea**
- **Hidden**
- **Password**
- **CheckBox**
- **Radio (/!\ tableau de /!\)**
- **Submit / Reset**
- **Select**
- **Option**
- **FileUpload**

Formulaires : Exemple

```
<form name='formu' onSubmit='return verif
  <input type='text' name='texte'><br>
  <select name='sel'>
    <option>?
    <option value=1>Un
    <option value=2>Deux
</select><br>
  <input type='radio' name='rad' id='rad1'>
  <label for='rad1'>oui</label>
  <input type='radio' name='rad' id='rad2'>
    <label for='rad2'>non</label><br>
  <input type='checkbox' name='chk' id='chk1'>
    <label for='chk1'>OK</label><br>
  <input type='submit' value='Envoyer'>
</form>
```

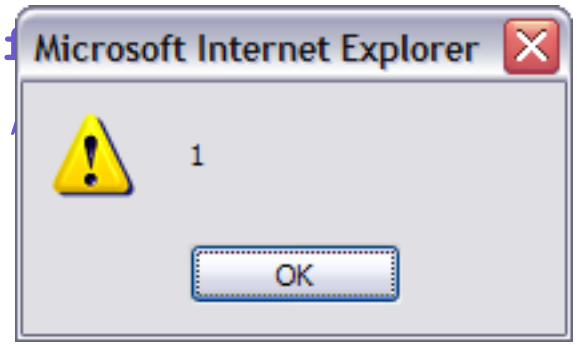


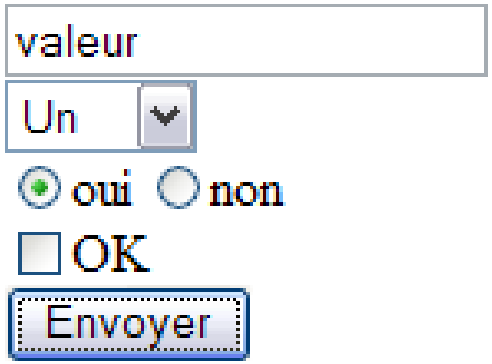
Formulaires : accès aux champs



Formulaires : Exemple

```
<script type="text/javascript">
function verif(f)
{
  window.alert(f.texte.value) ;
  window.alert(f.sel.selectedIndex) ;
  window.alert(f.sel[f.sel.selectedIndex].text) ;
  window.alert(f.sel[f.sel.selectedIndex].value) ;
  window.alert(f.rad[0].checked) ;
  window.alert(f
  return false ;
}
</script>
```





Expressions rationnelles

- Expressions rationnelles / régulières
  - `/modele/drapeaux  (drapeaux: g, i, gi)`
  - `new RegExp("modele"[, "drapeaux"])`
- Modèles
  - `^` : début de
  - `$` : fin de
  - `*` : 0 à n fois
  - `+` : 1 à n fois
  - `?` : 0 à 1 fois
  - `.` : un caractère sauf retour chariot
  - `|` : ou

Expressions rationnelles

- Modèles
  - **(x)** : **x** et mémorise
  - **{n}** : **n** fois
  - **{n,}** : au moins **n** fois
  - **{n,m}** : de **n** à **m** fois
  - **[xyz]** : 1 élément de la liste
  - **[a-z]** : 1 élément de la série
  - **[^xyz]** : 1 élément n'étant pas dans la liste
  - **[^a-z]** : 1 élément n'étant pas dans la série
  - **\b** : frontière de mot
  - **\B** : non frontière de mot

Expressions rationnelles

- Modèles
  - **`\d = [0-9]`** : chiffre
  - **`\D = [^0-9]`** : non chiffre
  - **`\n`** : retour à la ligne
  - **`\s`** : séparateur de mot
  - **`\S`** : non séparateur de mot
  - **`\t`** : tabulation
  - **`\w = [A-Za-z0-9_]`** : 1 caractère alphanumérique
- Méthodes
  - **`test(`*`chaine`*`)`**

Expressions rationnelles : Exemples

```
true
 <script type="text/javascript">
false
   document.write(/l/.test('Hello')) ;
false
   document.write(/^l/.test('Hello')) ;
true
   document.write(/^h/.test('Hello')) ;
true
   document.write(/^h/i.test('Hello')) ;
true
   document.write(/^Hel.o/.test('Hello')) ;
true
   document.write(/^Hel+o/.test('Hello')) ;
true
   document.write(/^He+llo/.test('Hello')) ;
true
   document.write(/^Hea*llo$/.test('Hello')) ;
true
   document.write(/^He(l|o)*$/.test('Hello')) ;
true
   document.write(/^H[leos]+/.test('Hello')) ;
false
   document.write(/^H[^leo]+/.test('Hello')) ;
true
   document.write(/^H[^kyz]+/.test('Hello')) ;
true
   document.write(/^H[a-z]*/.test('Hello')) ;
true
   document.write(/^H[a-z]*$/.test('Hello')) ;
 </script>
```

Dates : Propriétés & Méthodes

- Méthodes
  - Constructeur
  - getDay(), attention de 0 (dimanche) à 6 (samedi)…
  - getDate() / setDate()
  - getMonth() / setMonth(), attention de 0 à 11…
  - getHours()/ setHours()
  - getMinutes() / setMinutes()
  - getTime() / setTime()
  - getYear() / setYear() / getFullYear() / setFullYear()
  - parse()

Dates : Exemples

```
var today = new Date()
document.write(today) ;
var birthday = new Date("December 17, 1995
  03:24:00")
document.write(birthday) ;
birthday = new Date(95
document.write(birthday) ;
birthday = new Date(95, 11, 17, 3, 24, 0)
document.write(birthda
document.write(birthda
document.write(birthda
birthday.setDate(25) ;
document.write(birthda
```

Tue Nov 02 2010 00:11:36 GMT+0100

Sun Dec 17 1995 03:24:00 GMT+0100

Sun Dec 17 1995 00:00:00 GMT+0100

Sun Dec 17 1995 03:24:00 GMT+0100

0

17

Mon Dec 25 1995 03:24:00 GMT+0100

Dates : Exemples

```
document.write(birthda
birthday.setMonth(10)
document.write(birthda
document.write(birthday.getYear()) ;
birthday.setYear(96) ;
document.write(birthda
document.write(birthday.getFullYear()) ;
birthday.setFullYear(2010) ;
document.write(birthda
document.write(Date.pa
```

11

Sat Nov 25 1995 03:24:00 GMT+0100

95

Mon Nov 25 1996 03:24:00 GMT+0100

1996

Thu Nov 25 2010 03:24:00 GMT+0100

807919200000

Images : Propriétés & Méthodes

- Propriétés
  - **`complete`**
  - **`width`**
  - **`height`**
  - **`name`**
  - **`src`**
  - …
- Méthodes
  - **`constructeur`**
    - **`Image()`**
    - **`Image(`*`largeur, hauteur`*`)`**

Images: Exemples

```
<script type="text/javascript">
// Une image
rouge = new Image(100, 100) ;
rouge.src = 'rouge.gif' ; // Préchargement

// Une autre image
vert = new Image(100, 100) ;
vert.src = 'vert.gif' ; // Préchargement

// Modification de la 13e image de la page Web
document.images[12].src = rouge.src ;

// Modification de la 15e image de la page Web
document.images[14].src = rouge.src ;
</script>
```

Relations entre code HTML et DOM

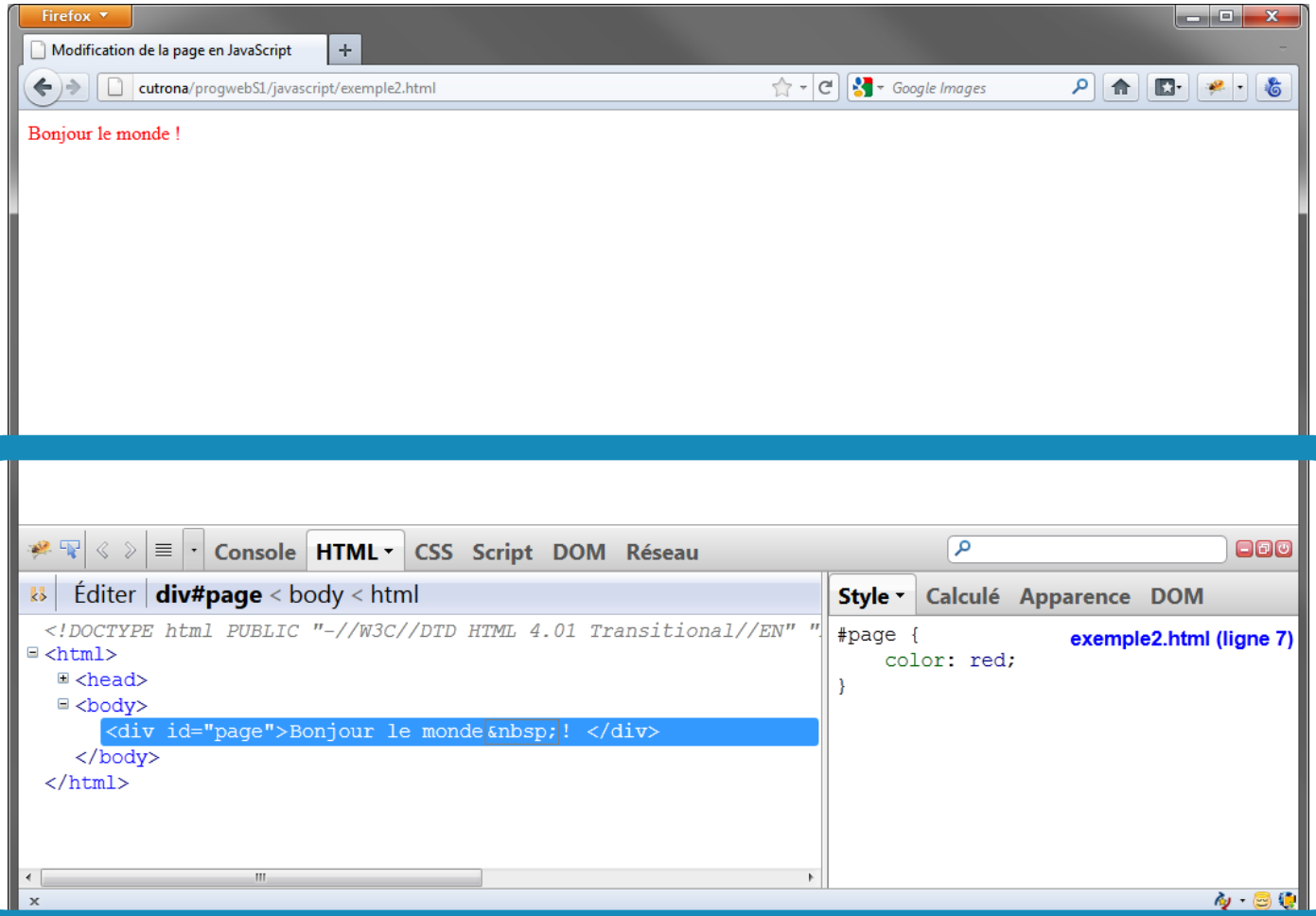
- Deux visions, normalement concordantes :
  - Le code HTML produit par le concepteur
  - L'interprétation qui en faite par le navigateur
- Discordances possibles :
  - Erreurs dans le code (code non valide)
  - Balises non supportées (HTML 5, par exemple)
  - Modifications de la page par JavaScript

- Comment explorer l'état réel de l'interprétation du code HTML / JavaScript par la navigateur ?

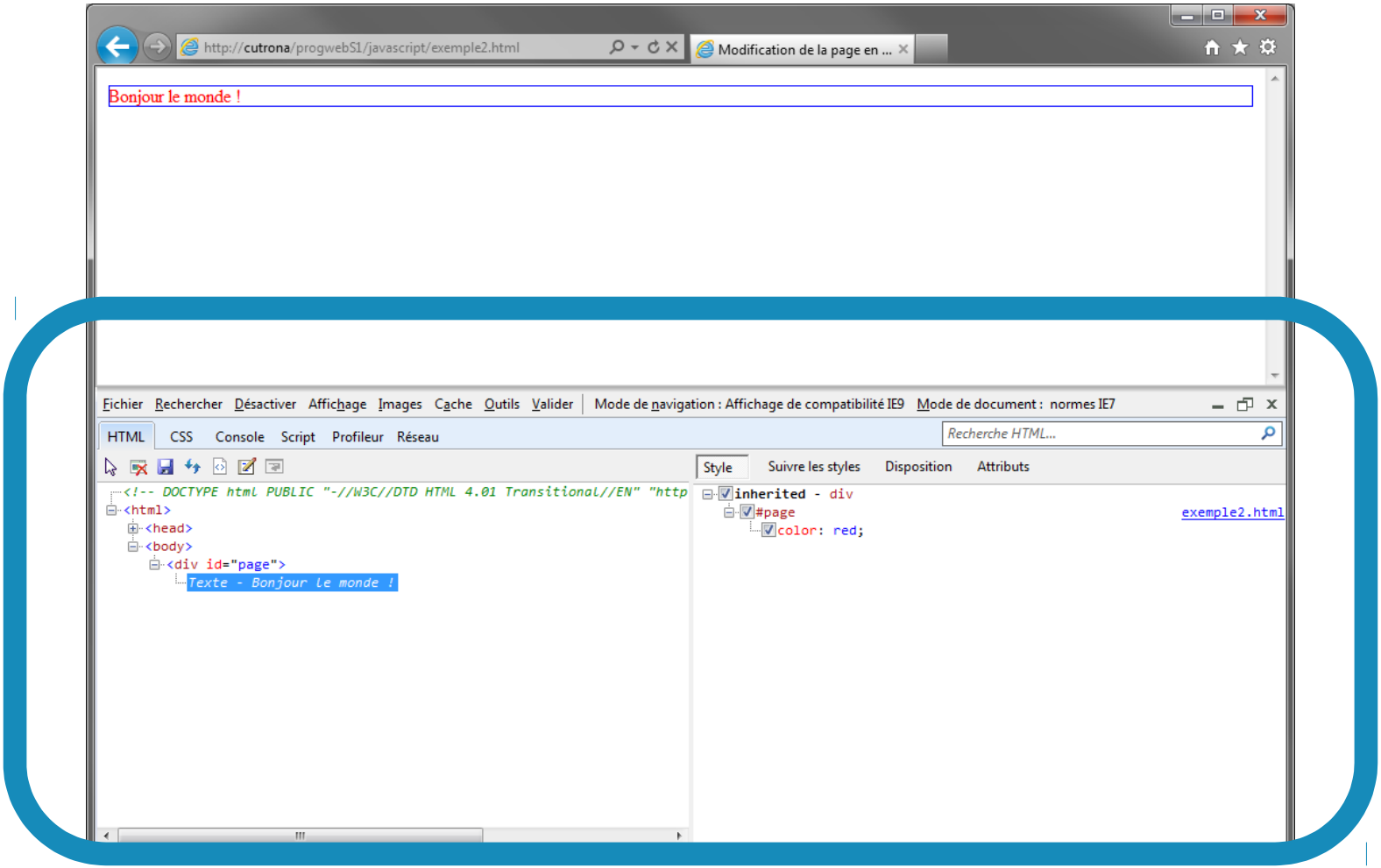
Firefox : Firebug

- Module complémentaire de Firefox
- https://addons.mozilla.org/fr/firefox/addon/firebug/
- Principales fonctionnalités :
  - Édition
  - Débogage
  - Modification
  - HTML
  - CSS
  - JavaScript
  - HTTP
- C'est l'outil indispensable du développeur Web

Firefox : Firebug



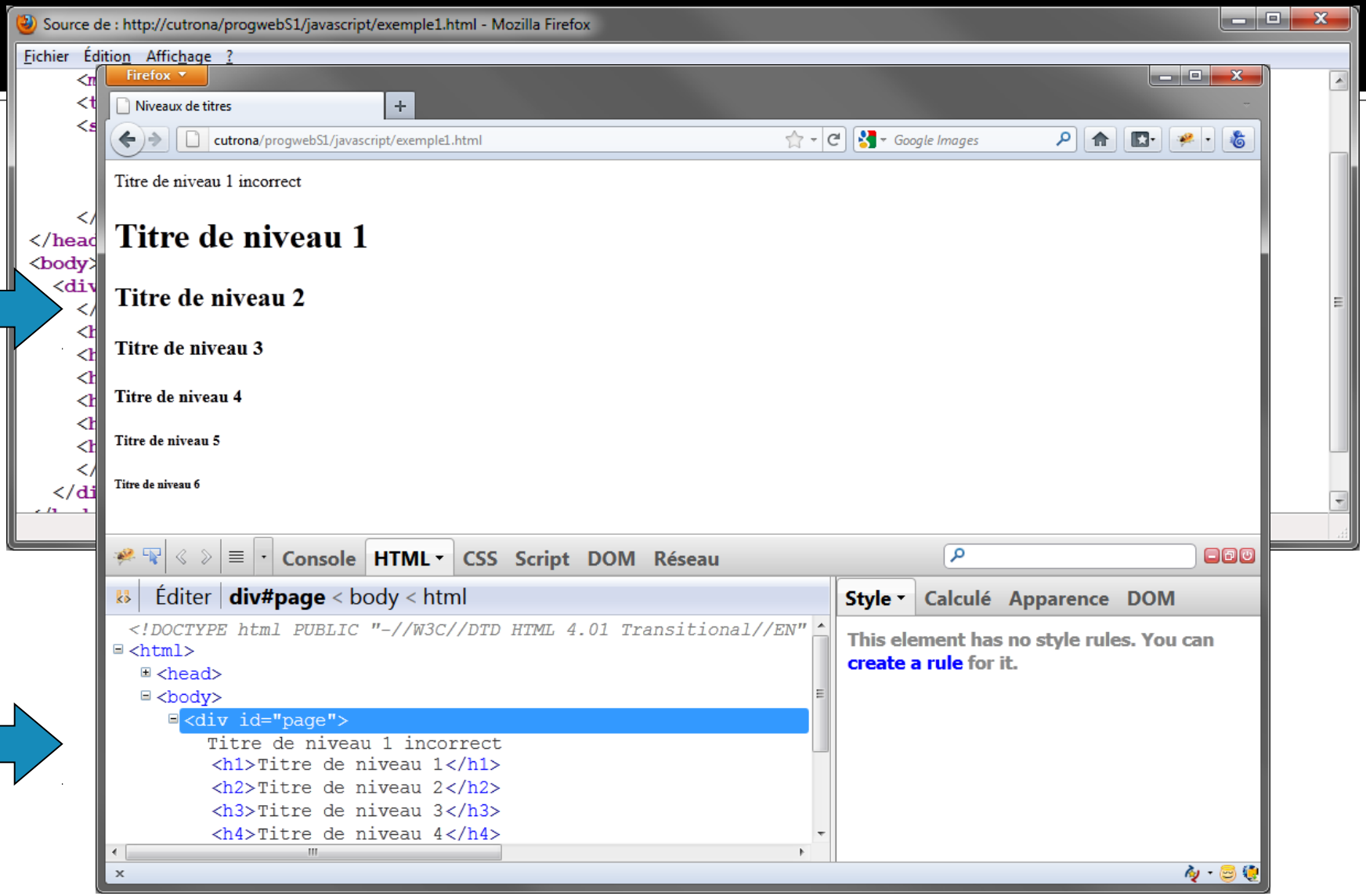
IE : outils de développement



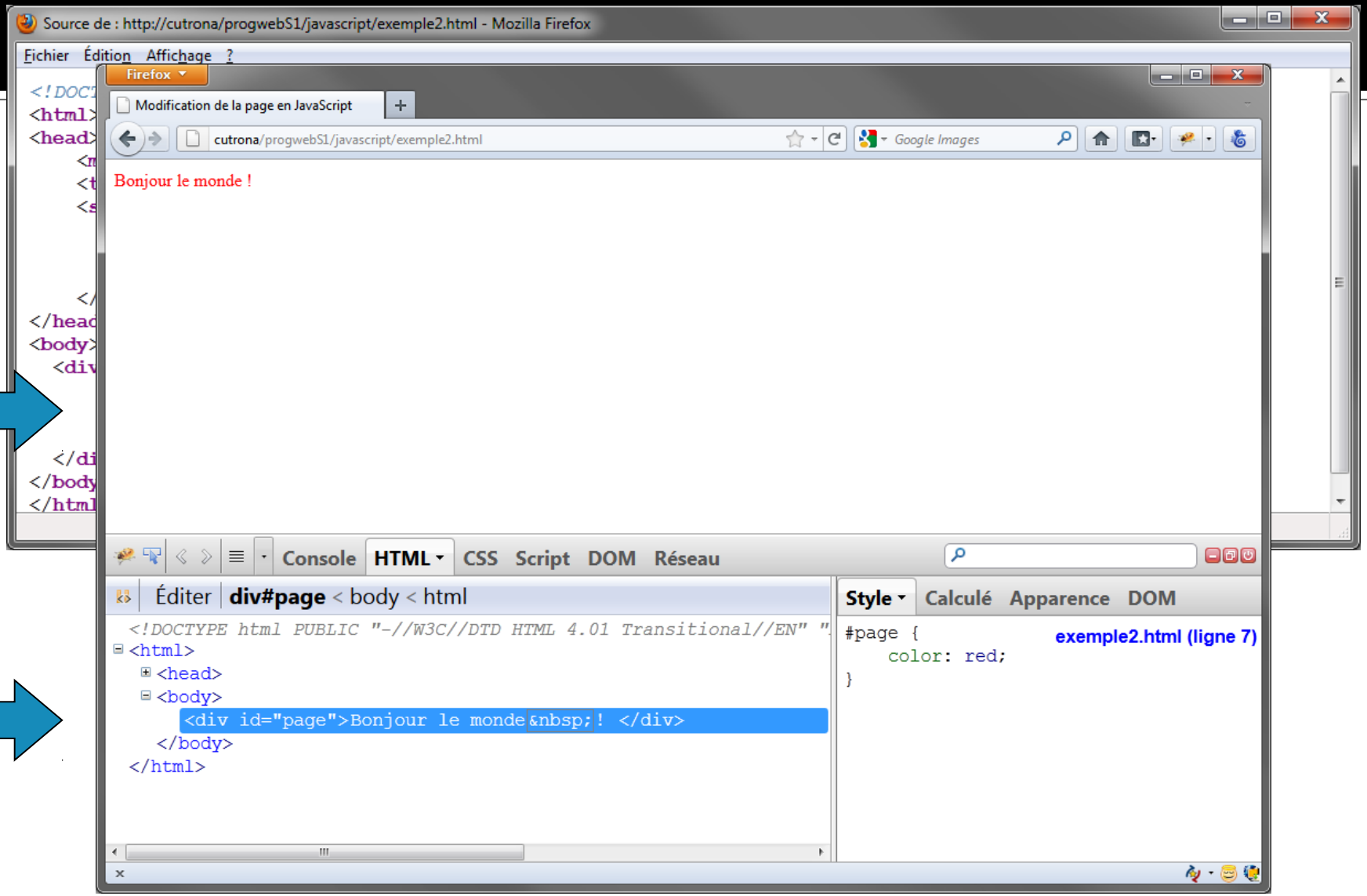
Et les autres navigateurs ?

- Opera
  - Widgets permettent d'obtenir des outils de développement
- Safari
  - Version allégée de Firebug
- Google Chrome
  - Outils de développement intégrés

Relations entre code HTML et



Relations entre code HTML et



Solutions pour modifier le DOM

- innerHTML
  - Construire une chaîne de caractères contenant du code HTML
  - Affecter cette chaîne de caractères à l'attribut **`innerHTML`** d'un élément de la page
  - Le navigateur interprète le contenu de la chaîne de caractères affectée pour modifier la structure du document
- DOM « pur »
  - Construire de nouveaux éléments logiques du document avec des méthodes JS
  - Construire les liens de parenté entre ces éléments

innerHTML : exemple

- Identifier un élément HTML

```
<div id="un_id"></div>
```

- Accéder à un élément

```
e = document.getElementById("un_id");
```

- Construire une chaîne contenant du HTML

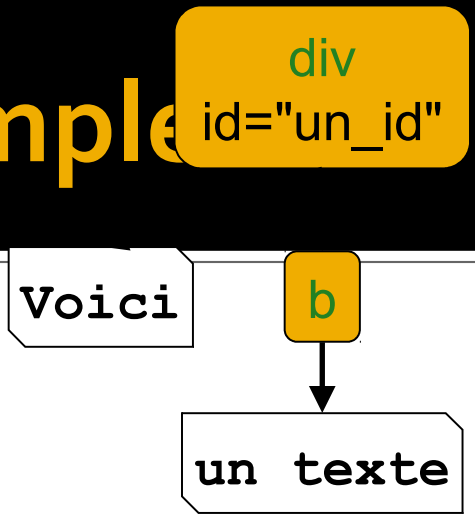
```
s = "Voici <b>un texte</b>";
```

- Modifier le contenu de l'élément

```
e.innerHTML = s;
```

- Interprétation « automatique » par le navigateur du nouveau contenu pour modifier le document

DOM « pur » : exemple



- Identifier un élément HTML

```
<div id="un_id"></div>
```

- Accéder à un élément

```
e = document.getElementById("un_id");
```

- Créer un nœud de type « texte »

```
t1 = document.createTextNode('Voici ');
t2 = document.createTextNode('un texte');
```

- Créer un nouveau nœud de type « balise »

```
b = document.createElement('b');
```

- Construire des liens de parenté

```
e.appendChild(t1);
b.appendChild(t2); e.appendChild(b);
```

Gestion des événements

- Interactions HTML / JavaScript
- Possibilité d'associer du code JavaScript à certains événements dans la page Web
- Evénements
  - OnMouseOver
  - OnMouseOut
  - OnClick
  - OnKeyDown
  - OnFocus
  - OnBlur
  - ...

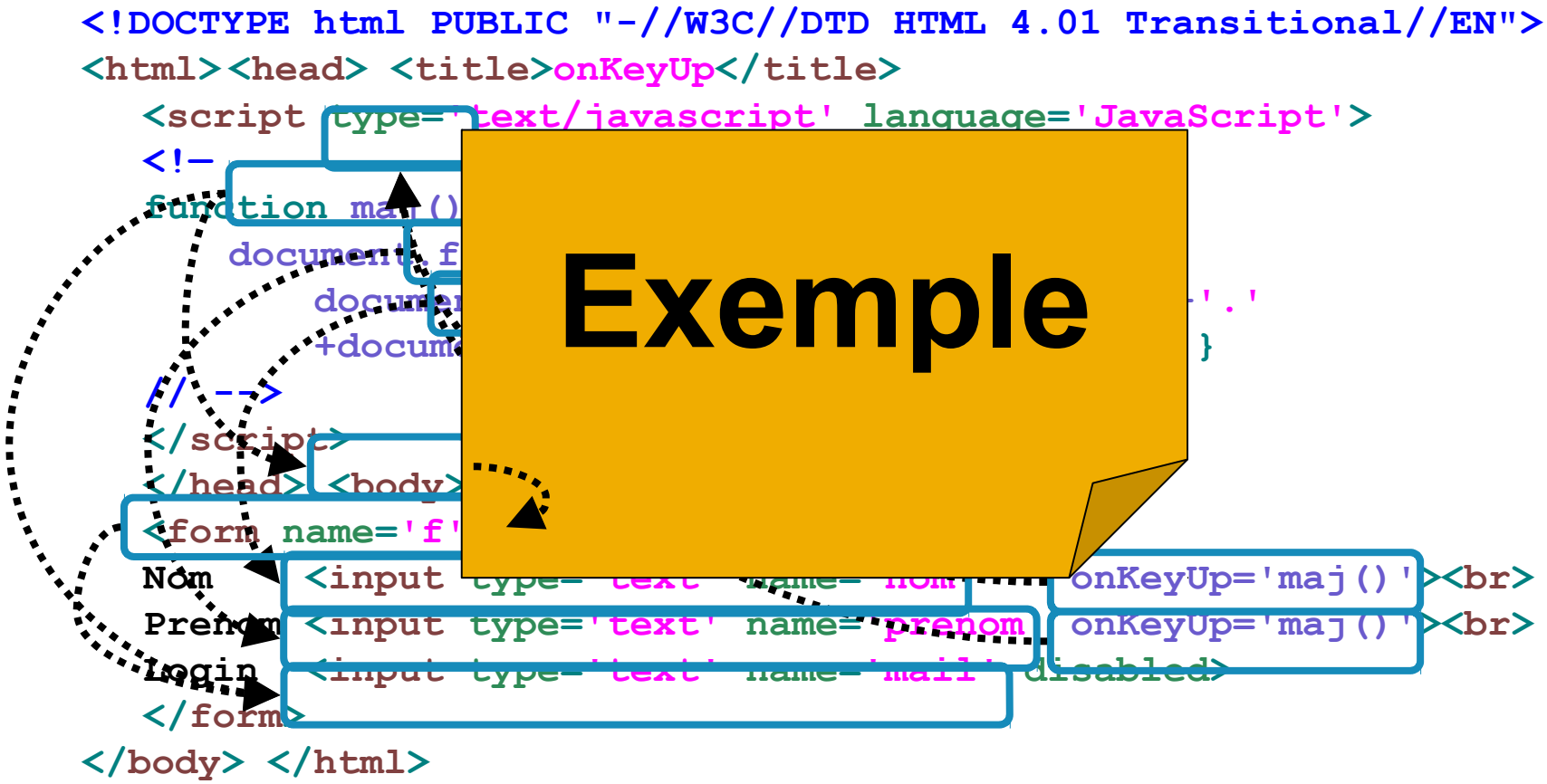
Mise en place des événements

```
<a href='URI cible'
   onMouseOver="code_javascript1"
   onMouseOut="code_javascript2">Support</a>
```

```
<a href='URI_cible'
   onMouseOver="a=1"
   onMouseOut="b=2">Support</a>
```

```
<a href='URI_cible'
   onMouseOver="allumer()"
   onMouseOut="eteindre()">Support</a>
```

Événement onKeyUp



Événement onMouseOver / Out

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head> <title>Rollover</title>
   <script type='text/javascript' language='JavaScript'>
   <!--
   function chang
      document.i
   // -->
   </script>
   </head>
   <body>
      <a href='
         onMouse                                    .gif')"
         onMouseOut= change( 'image1' , 'rouge.gif')">
        <img name='image1' width='100' height='100'
             src='rouge.gif' alt='disque'></a>
   </body>
</html>
```

Exemple

ages du
es par leur
indice

Événement onMouseOver / Out

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html> <head> <title>Rollover</title>
   <script type='text/javascript' language='JavaScript'>
     function cha
    document.imag
   </script> </hea
   <li><a href="#"                                    ert.gif')"
   onMouseOut="cha                                    e 1</a>
   <li><a href="#"                                    ert.gif')"
   onMouseOut="cha                                    e 2</a>
   <li><a href="#"                                    ert.gif')"
   onMouseOut="cha                                    que 3</a> </ul>
   <img name='imag                                    src='rouge.gif'
   alt='disque'>
   <img name='image2' width='100' height='100' src='rouge.gif'
   alt='disque'>
   <img name='image3' width='100' height='100' src='rouge.gif'
   alt='disque'>
</body> </html>
```

Exemple

Boîte à outils (1/3)

```
// Détection du navigateur
// Netscape 4
var nava = document.layers ? true : false ;
// IE, Firefox, Netscape 6, Opera
var dom = document.getElementById ? true : false ;
// IE, Opera
var iex = document.all ? true : false ;
// Récupérer un objet identifié
function getobj(id)
{
  var obj ;
  if (nava)      { obj=document.id }
  else if (dom) { obj=document.getElementById(id) }
  else if (iex) { obj=id }
  return obj ;
}
```

Boîte à outils (2/3)

```
// Récupérer le style d'un objet identifié
function getstyle(id)
{
   var obj ;
   if (nava)
      { obj=document.id }
   else if (dom)
      { obj=document.getElementById(id).style }
   else if (iex)
      { obj=id.style }
   return obj ;
}
```

Boîte à outils (3/3)

```
// Écrire un contenu HTML dans un objet identifié
function writecontent(obj, content)
{
  if (nava) {
    var objet=getstyle(obj) ;
    objet.document.write(content) ;
    objet.document.close() ;
  }
  else if (dom) {
    document.getElementById(obj).innerHTML=content ;
  }
  else if (iex) {
    document.all(obj).innerHTML=content ;
  }
}
```

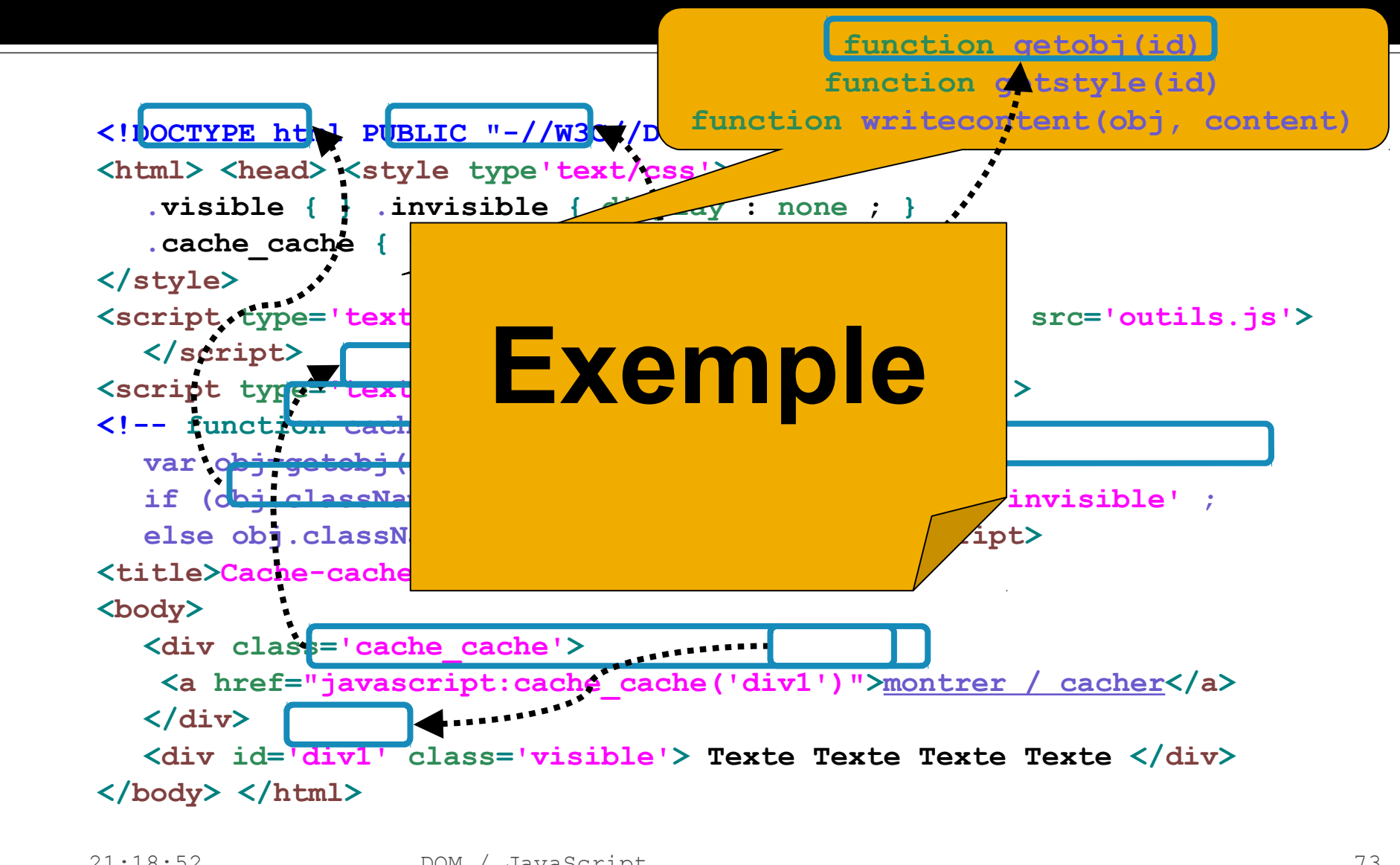
Événement onMouseOver / Out

```
function getobj(id)
function getstyle(id)
function writecontent(obj, content)
```

```
<!DOCTYPE html PUBLIC "-//W3C//D... 1.01 Transitional//EN">
<html> <head> <title>Image cliquable</title>
<script type='text/javascript' language='JavaScript' src='outils.js'>
   </script> </head
<h1>Survolez mon i...les ;-)</h1>
<img src='formes.g...'image sur laquelle
  il faut cliquer"
<map name='map'>
  <area href='#' a...circle'
  coords='94,67,4... 'Cercle')'
       onMouseOut
  <area href='#' a...shape='rect'
   coords="171,20,...ent('info',
   'Rectangle')"  ...writecontent('info', '')">
  <area href='#' alt='Etoile' title='Etoile' shape='poly'
  coords='116,159,120,180,151,185,136,202,138,227,116,217,94,227,96,203,
  80,184,103,180' onMouseOver="writecontent('info', 'Etoile')"
    onMouseOut="writecontent('info', '')">
</map> <span id='info'></span>
</body> </html>
```

Exemple

Modification dynamique de style



Modification dynamique de contenu