JavaScript

Jean-Claude Charr

Maître de conférences

IUT de Belfort – Montbéliard
Université de Franche Comté

Description générale

- JavaScript rend les pages html interactives
- JavaScript est un langage de programmation de scripts
- Il est en général ajouté aux pages html ou dans un fichier avec une extension .js
- Il est interprété et par suite exécuté sans être compilé
- Java et JavaScript deux langages complètement différents

JavaScript dans un fichier HTML

```
<html>
    <head>
        <script type="text/javascript">
            function message(){
                alert("la fonction message a été appelée"); }
        </script>
    </head>
    <body>
        <script type="text/javascript">
                alert("message écrit par JavaScript");
                message();
        </script>
    </body>
</html>
```

JavaScript dans un fichier externe

Fichier HTML contenant :

```
<html>
    <head>
        <script type="text/javascript" src="script.js"></script>
    </head>
    <body onload="popup()">
    </body>
</html>
```

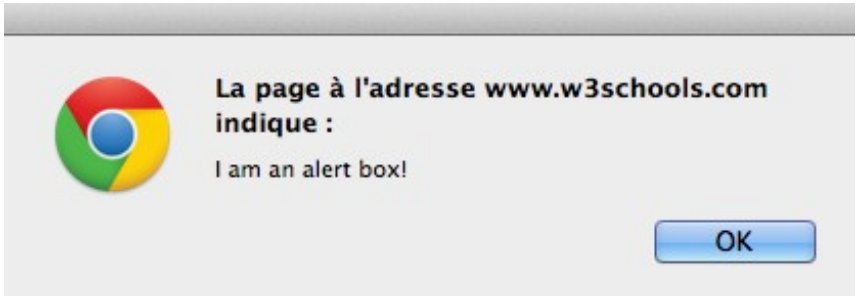
le fichier script.js contenant :

```
function popup() {
    alert("Hello World") ;
}
```

Popup boxes

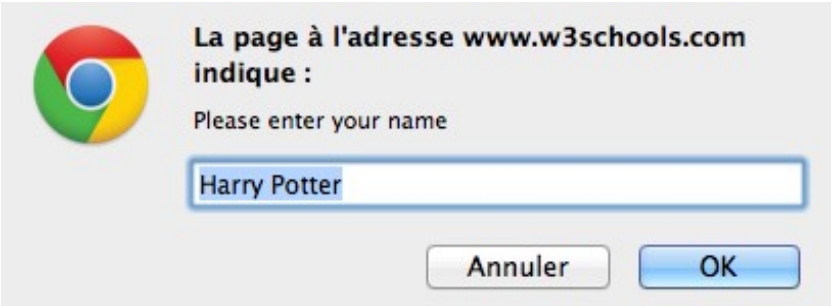
Alert box :

```
alert("I am an alert box!");
```



Confirm box :

```
var r=confirm("Press a button") ;
```



Prompt box :

```
var name=prompt("Please enter your name","Harry
    Potter");
```

Syntaxe

- Javascript est sensible à la casse.
- Les instructions sont exécutées en séquentiel
- Une commande JS termine par un ; ou un retour à la ligne
- { } encapsulent un bloc de JavaScript
- Commenter une ligne : // commentaire
- Commenter un paragraphe : /*...commentaire
  */

Variables

- Déclarer une variable avec l'instruction var ou let:
  Ex : var x; let y; (var hoisted)
  var nom;

- Affecter une valeur à une variable avec l'opérateur = :
  Ex : x=5; nom="Dupont";
  var prenom="Rita";

- Ajouter \ pour écrie un caractère spécial dans le texte :
  \', \", \&, \n, \\, \t ...
  Ex : document.write ("\"Rita\" \& I are singing!");

Opérateurs

- Opérateurs arithmétiques binaires : +, -, *, /, %, +=, ...
  Ex : x= ; y=x+ ; y-=x ;

- Opérateurs arithmétiques unaires : ++, -- ;
  Ex : x=5; x-- ;

- Opérateur de concaténation + :
  Ex : NomComplet= "Rita"+" Dupont" ;

- Opérateurs de comparaisons : ==, ===, !=, <, >, <=, >=
  Ex : if(age<18) alert("Trop jeune") ;

- Opérateurs logiques : &&, ||, !
  Ex : if(x<5 && x>3) alert("x=4");

L'instruction if()

```
<script type="text/javascript">
        var d = new Date()
        var time = d.getHours()
        if (time<10) {
            alert("Good morning");
        }
        else if (time>10 && time<16) {
                alert("Good day");
            }
            else {
                alert("Hello World!");
            }
    </script>
```

L'instruction switch

```
<script type="text/javascript">
    var d=new Date();
    theDay=d.getDay();
    switch (theDay){
        case 5 :   alert("Finally Friday");
                    break;
        case 6 :   alert("Super Saturday");
                    break;
        case 0 :   alert("Sleepy Sunday");
                    break;
        default :   alert("I'm looking forward to this weekend!");
    }
</script>
```

Fonctions

```
<html>
    <head>
        <script type="text/javascript">
            function product(a,b){
                return a*b;
            }
        </script>
    </head>
    <body>
        <script type="text/javascript">
            alert(product(4,3)); //appel
        </script>
    </body>
</html>
```

Les boucles

La boucle for :

```
Ex : for (i=0;i<=5;i++){
        console.log("The number is " + i) ;
     }
```

La boucle while :

```
     while (i<=5){
         console.log("The number is " + i) ;
         i++;
      }
```

La boucle do … while :

```
     do{
         console.log("The number is " + i);
         i++;
        } while (i<=5);
```

Les instructions break et continue

Exemple :

```
for (i=0;i<=10;i++){
      if (i==3)
          continue;
      else if( i==5)
          break;
      console.log("The number is " + i);
      console.log("<br />");
}
```

Évènements

- Permettent de créer des pages dynamiques
- Peuvent déclencher l'exécution de fonctions JS
- Chaque élément HTML peut générer plusieurs évènements
- Exemple d'évènements :
  onload et onunload : lors de chargement de la page
  onfocus, onchange, onblur : concernant un textfield
  onmouseover : la souris est mise sur un élément

```
<input type="text" size="30" id="email" onchange="checkEmail()">
```

Évènements modèle en ligne

```
<html>
   <head>
      <script type="text/javascript"> // Animation
         function over(){
              document.getElementById("b1").src ="b_blue.gif";
         }
         function out(){
              document.getElementById("b1").src ="b_pink.gif";
         }
     </script>
   </head>
   <body>
      <a href="http://www.google.com" target="_blank">
         <img border="0" alt="Visit Google!" src="b_pink.gif" id="b1"
         onmouseover="over()" onmouseout="out()" /> </a>
</body>
</html>
```

Évènements modèle traditionnel

```
<script type="text/javascript">
   function over(){
        document.getElementById("b1").src ="b_blue.gif";
   }
   function out(){
       document.getElementById("b1").src ="b_pink.gif";
   }
   function eventRegister(){
       document.getElementById("b1").onmouseover=over ;
       document.getElementById("b1").onmouseout=out ;
   }
</script>
</head>
<body onload="eventRegister()">
 <a href="http://www.google.com" target="_blank">
     <img border="0" alt="Visit Google!" src="b_pink.gif" id="b1"/>
</a>
```

L'objet event

Permet de récupérer des informations sur l'évènement déclenché : type, ctrlkey, clientX, clientY, keyCode ...

Ex : document.onmouseover = mouseOver;

```
    function mouseOver( e ){
        var src ;
        src=e.target;
        src.style.color="red";
    }
}
```

Les objets

- JS est un langage orienté objet
- Un objet est composé d'attributs et méthodes
- Il existe des objets prédéfinis dans JS comme String

Exemple :

```
script type="text/javascript">
    var txt="Hello World!";
    document.write(txt.length+"\n") ; txt.toUpperCase();
    txt.big(); txt.bold(); txt.fontcolor("green");
    txt.link("http://www.google.com");
    document.write(txt+" "+txt.replace("World","everyone")) ;
    document.write("\n"+txt.indexOf("d"));
```

L'objet Math

- Contient plusieurs constantes et méthodes/ opérateurs mathématiques.

Exemples : Math PI;
Math.sqrt(16);
Math.cos(90);
Math.round(4.7);
Math.random();
Math.max(0,150,30,20,38);
Math.pow(4,5);

L'objet String

L'objet String fournit plusieurs méthodes pour manipuler les chaînes de caractères

Exemples :

```
var txt="Java script";
txt.charAt(2);
txt.concat(" is the best");
txt.split(" ");
txt.toLowerCase();
```

L'objet Date (1/2)

- Permet d'avoir la date et le temps courant ou de préciser une date et un temps spécifique.

- Permet de manipuler la date et le temps

Exemple :

```
today = new Date()
d1 = new Date("October 13, 1975 11:13:00");
d2 = new Date(79,5,24);
d3 = new Date(79,5,24,11,33,0);
today.setDate(today.getDate()+5) ;
d1.setFullYear(2010,0,14);
if (d1<today)
        alert("Today is after 14th January 2010");
```

L'objet Date (2/2)

```
<html>
   <head>
      <script type="text/javascript">
         function startTime(){
               var today=new Date() ;
               var h=today.getHours();
               var m=today.getMinutes() ;
               var s=today.getSeconds();
         document.getElementById('txt').innerHTML=h+":"+m+":"+s;
               t=setTimeout('startTime()',500);   }
      </script> //clearTimeout(t); pour arrêter
   </head>
   <body onload="startTime()">
      <div id="txt"></div>
   </body>
</html>
```

L'objet Array (vecteur)

- Permet de stocker plusieurs éléments dans une variable

- Créer et initialiser un vecteur :
  Ex : var myCars=new Array() ;
  var myCars=new Array("Saab","Volvo","BMW") ;
  var myCars=["Saab","Volvo","BMW"];

- Accéder et modifier un élément d'un vecteur :
  Ex : myCars[0]="Opel" ;

- Contient plusieurs méthodes pour arranger, ajouter et supprimer des éléments : sort(), join(), reverse(), pop(), push(), shift(), ...

L'objet Boolean

- Il peut seulement avoir les valeurs false ou true.

- Exemples de création et initialisation :

```
var myBoolean=new Boolean() ;
var myBoolean=new Boolean(true);
var myBoolean=new Boolean(0);
var myBoolean=new Boolean(null) ;
var myBoolean=new Boolean("true");
var myBoolean=new Boolean("");
var myBoolean=new Boolean(false) ;
var myBoolean=new Boolean("false");
var myBoolean=new Boolean(NaN) ;
var myBoolean=new Boolean("Richard");
```

Définir un objet et le manipuler (1/2)

- Définir la structure de l'objet (template) :

```
function person(firstname,lastname,age,eyecolor){
   this.firstname=firstname;      //les attributs
   this.lastname=lastname;
   this.age=age;
   this.eyecolor=eyecolor ;
   //les méthodes
   this.setFirstName=setFirstName;
   this.getFirstName=getFirstName;
}
function getFirstName(){
   return this.firstName() ;
}
```

Définir un objet et le manipuler (2/2)

- Créer une instance de l'objet défini :
  Ex : father=new person("John","Doe",50,"blue");

- Accéder aux attributs de l'objet instancié
  Ex : document.write(father.firstname);

- Appeler une méthode de l'objet instancié
  Ex : father.setFirstName("Jack");

Détecter le navigateur

- Permet d'afficher des informations différentes selon le navigateur utilisé par le client

- L'objet navigator :

```
document.write("Browser Name: "+navigator.appName)
document.write("Browser Version : " + navigator.appVersion)
document.write("Cookies Enabled: " + navigator.cookieEnabled)
```

Cookies

- Une cookie est une variable stockée sur l'ordinateur du Client.

- Elle peut contenir des informations concernant le client comme : nom, mot de passe, date du dernier accès...

Exemple :

```
function checkCookie(){
    username=getCookie('username');
      if (username!=null && username!="")
        alert('Welcome again '+username+'!') ;
      else{
        username=prompt('Please enter your name:',"");
      if (username!=null && username!="")
         setCookie('username',username) ; }
      }
```

Cookies

```
function setCookie(c_name,value){
     document.cookie=c_name+ "=" +escape(value)+365;
}

function getCookie(c_name){
    if (document.cookie.length>0){
       c_start=document.cookie.indexOf(c_name + "=");
       if (c_start!=-1){
              c_start=c_start + c_name.length+1;
              c_end=document.cookie.indexOf(";",c_start);
              if (c_end==-1) c_end=document.cookie.length;
          return unescape(document.cookie.substring(c_start,c_end));
       }
   }
return "";
}
```

JS pour valider un formulaire (1/2)

```
<html>
   <head>
      <script type="text/javascript">
         ...
      </script>
   </head>
   <body>
      <form action="submit.htm" onsubmit="return validate(this)"
      method="post">
         Email: <input type="text" name="email" size="30">
         <input type="submit" value="Submit">
      </form>
   </body>
</html>
```

JS pour valider un formulaire (2/2)

```
function validate(thisform){
    with (thisform){
       if (validate_required(email,"Email must be filled out!")==false){
          email.focus();  return false;}
    }
}

function validate_required(field,alertTxt){
     with (field){
          if (value==null || value==""){
               alert(alertTxt);  return false;}
          else
               return true;
     }
}
```