



**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UMBB**

**Cours Master2 Réseau et
Télécommunications**

Technologies du web

Samira_mechid@yahoo.fr

2025-2026



Chapitre 4 : JavaScript

Définitions

- Le JavaScript, créé en 1995 par Brendan Eich
- C'est un langage de programmation de scripts orienté objet
- JavaScript est un langage qui n'a absolument pas les prétentions de Java.
- Java est un langage de programmation à part entière (comme le C, le C++, le Pascal...). Son terrain d'action ne se limite pas au Web.

Définitions

- JavaScript est profondément intégré à HTML
- JavaScript est un langage de script ➡ **un langage interprété**
- Il possède de nombreuses caractéristiques des langages objet, mais il ne peut pas prétendre être un véritable langage objet puisqu'il n'implémente pas la notion d'héritage et possède une notion de classe très simpliste



Définitions

■ Les scripts permettent une interaction entre la page et l'utilisateur.

■ Les scripts côté client peuvent être générés par un script côté serveur.

(Exécuté côté client, ce qui signifie dans un navigateur Web. Mais peut aussi être employé côté serveur)



Définitions

Quoi faire avec JavaScript ?

- ❑ Permet de valider les formulaires côté client.
- ❑ Attention : ne dispense pas de la validation côté serveur, qui est indispensable (sécurité).

On peut accéder aux formulaires et aux champs par leurs id ou bien par leur nom en utilisant des fonctions javascript à certains événements (on quitte un champ, on valide le formulaire, on presse une touche...)



Définitions

Quoi faire avec JavaScript ?

- ☐ Faire apparaître et cacher des éléments HTML.
 - ☐ Afficher des boîtes de dialogues (avertissement, confirmation,...).
 - ☐ Gérer les événements liés à la souris et au clavier.
- Communication en temps réel (Web Sockets).....

Définition du Script JavaScript

1. Dans l'élément <script> : (interne)

<html>

<head>

<title>Cours JavaScript</title>

</head>

<script>

alert("Cours DAW!");

</script>

<body>

<h2>Mon premier script JS</h2>

</body>

</html>



Où définir le script js (interne)

1. Dans l'élément head :

Exemple :

```
<html>
<head>
<title>Cours JavaScript</title>
<script>
alert("Cours Développement Web!");
</script>
</head>
<body>
<h2>Mon premier script JS-M2R-</h2>
</body>
</html>
```



Définitions

```
<html>
<head>
<title>Cours JavaScript</title>
<script>
alert(" Cours Développement Web!");
</script>
</head>
<body>
<h2>Mon premier script JS-M2R-</h2>
</body>
</html>
```

Où définir le script js (interne)

The screenshot illustrates the process of defining an internal JavaScript script. It shows a web browser window with the title 'Cours JavaScript' and a file path 'C:/Users/PC/Desktop/COURS-S...'. An alert box is displayed with the message 'Cours Développement Web!' and an 'OK' button. A large red arrow points from the 'OK' button to a code editor window titled 'Cours JavaScript'. The code editor shows the following HTML code:

```
Mon premier script JS-M2R-
```

Below the code editor, there is a Windows taskbar with the search bar 'Taper ici pour rechercher' and the system tray showing the date '25/10/2024' and time '20:35'.

Cliquant sur OK



Où définir le script js (interne)

1. Dans l'élément body :

Exemple :

```
<html>
<head>
<title>Cours JavaScript</title>
</head>
<body>
<h2>Mon premier script JS-M2R-</h2>
<script>
alert("Cours Développement Web!");
</script>
<h1>L'ordre compte</h1>

</body>
</html>
```



Définitions

Où définir le script js (interne)

????????????????



Définitions

Définition du Script JavaScript

2. Dans l'élément <script> : (externe)

<html>

<head>

<title>Cours JavaScript</title>

</head>

<script src="mon_script.JS">

</script>

<body>

<h2>Mon premier script JS</h2>

</body>

</html>

Définition du Script JavaScript

3. À travers des liens ou bouton : (Attributs d'événements)

```
<body>
```

```
<a href="javascript: (alert('Cours DAW'))">
```


```
Cliquez ici </a>
```

```
</body>
```

```
<body>
```

```
<input type="button" value= "Cliquez" onclick="alert('Cours')">
```

```
</body>
```



Attribut d'évènement

Exemples d'attributs d'évènements:

onload: Après le chargement de la page.

onclick: Après avoir cliqué sur l'élément (un bouton, champ input,...).

ondblclick: Après un double clic sur l'élément.

oninput: Après avoir saisi une valeur dans l'élément.

Définitions

Exemples d'attributs d'évènements:

onchange: Après avoir changé la valeur de l'élément.

onselect: Après avoir sélectionné une partie du texte de l'élément.

onsubmit: Après avoir cliqué sur le bouton submit d'un formulaire

onreset: Après avoir cliqué sur le bouton reset d'un formulaire.

onkeypress: Après avoir tapé un caractère sur le clavier.

onkeydown: Après avoir cliqué sur n'importe quel bouton du clavier.

.....

Exemple

```
<html>  
  <head><title>Cours JavaScript</title>  
  
  </head>  
  <body onload="alert('Page JS')">  
  
    <h2>Mon premier script JS-M2R-</h2>  
  
  </body>  
</html>
```

Exemple

```
<html>
  <head><title>Cours JavaScript</title>
  </head>
  <body onload="alert('Page JS')">
<input type="text" onselect="alert('texte sélectionné')" >
<br>
<input type="text" oncopy="alert('texte copié')" >
<br>
<input type="text" onclick="alert('un seul click')" >
<br>
<input type="text" ondblclick="alert('double click')" >
<br>
<input type="text" onmouseover="alert('sourisxxxx')" >
</body>
</html>
```


Remarques

De préférence, d'écrire le code Javascript dans un fichier externe, portant l'extension **js**

Ce fichier est ensuite appelé depuis la page Web au moyen de l'élément **<script>** comme suit :

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

Remarques

- *JavaScript fait la différence entre lettres minuscules et majuscules : les mots AA et aa peuvent représenter deux variables différentes.*
- Pour rendre un programme plus lisible, il est conseillé d'y mettre des commentaires ;



Eléments de bases

Les commentaires

Les commentaires commencent avec la séquence barre oblique - étoile. Les commentaires se terminent avec la séquence inverse : une étoile suivie d'une barre oblique.

Exemple

```
/* Ceci est un commentaire JS */
```

Ces commentaires peuvent s'étendre sur plusieurs lignes



Eléments de bases

Les commentaires Les commentaires de fin de ligne commencent par la séquence constituée de deux barres obliques (ils n'ont pas de séquence de terminaison, puisqu'ils ne se terminent qu'à la fin de la ligne courante).

Exemple :
action quelconque // Ceci est un commentaire JS
action suivante



Eléments de base

La boîte de dialogue alert()

alert() est une fonction, qui permet d'afficher une boîte de dialogue contenant un message.

Exemple :

```
alert('Cours JavaScript')
```

La boîte de dialogue alert()

Cette page indique

Cours JavaScript

OK



Eléments de base

La syntaxe Javascript

Les instructions doivent être séparées par un point-virgule que l'on place à la fin de chaque instruction :

<script>

instruction_1;

instruction_2;

instruction_3;

</script>



Eléments de base

La syntaxe d'une fonction est la suivante :

```
<script>  
myFunction();  
</script>
```

Par exemple :

```
<script>  
alert('Bonjour');  
</script>
```




Eléments de base

Bases des variables

```
<script>  
var myVariable;  
myVariable = 2;  
</script>
```

Ou

```
<script>  
var myVariable = 2;  
</script>
```



Eléments de base

Bases des variables (Remarques)

- Une variable peut être de type numérique, mais aussi une chaîne de caractères ou peut être de type booléen (boolean) ainsi que Null et Undefined
- Il existe en JavaScript une notion de variable globale et locale. Les variables locales sont déclarées au sein d'une fonction. Les variables globales sont les variables déclarées en dehors du corps d'une fonction; (►voir exemples plus loin)



Bases des variables (Remarques)

JavaScript est un langage **non typé**. Cela signifie que le type d'une variable est défini **uniquement au moment de l'exécution**

Objets référencés:

Voir plus loin

Conversion :

X='12'; X = parseInt(X) alors X=12

Typage dynamique des variables: Le type d'une variable est déterminé automatiquement selon la valeur affectée à cette variable.

Exemple: *Typage dynamique*

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

```
var A ;
```

```
alert ( typeof (A) ) ; ← Affiche: undefined
```

```
A = -10.56 ;
```

```
alert ( typeof (A) ) ; ← Affiche: number
```

```
A = "Bonjour";
```

```
alert ( typeof (A) ) ; ← Affiche: string
```

```
A = false;
```

```
alert ( typeof (A) ) ; ← Affiche: boolean
```

```
A = [12, 34];
```

```
alert ( typeof (A) ) ; ← Affiche: object
```

```
</script>
```




Eléments de base

Les opérateurs

- **Opérateurs unaires: Signe(+)(-)**
- **Incrémentation(++), décrémentation(--)**



Eléments de base

Les opérateurs

Les opérateurs arithmétiques

On peut utiliser 5 opérateurs arithmétiques : l'addition (+), la soustraction (-), la multiplication (*), la division (/) et le modulo (%). Le modulo est le reste d'une division.

Exemple

```
<Script>  
var x = 3;  
  x+= 5;  
  alert(x);  
</script>
```




Éléments de base

Opérateurs de comparaison

$==$: égal à

$!=$: différent de

$>$ supérieur à

$>=$ supérieur ou égal à

$<$: inférieur à

$<=$: inférieur ou égal à



Eléments de base

Les opérateurs logiques

&& qui signifie : **valeur₁ && valeur₂**

|| qui signifie OU : **valeur₁ || valeur₂**

! qui signifie NON : **!valeur**



Eléments de base

La fonction prompt()

Exemple

```
<script>
```

```
var start = 'Salut ', name, end = ' !', result;
```

```
name = prompt('Quel est votre prénom ?');
```

```
result = start + name + end;
```

```
alert(result);
```

```
</script>
```



Éléments de base

La fonction prompt()

Cette page indique

Quel est votre prénom ?

OK

Annuler

La fonction prompt()

Cette page indique

Salut ZAKARIA !

OK



Eléments de base

La fonction prompt()

Exemple

```
<script>
```

```
var first, second, result;
```

```
first = prompt('Entrez le premier chiffre :');
```

```
second = prompt('Entrez le second chiffre :');
```

```
result = parseInt(first) + parseInt(second); /* la fonction  
parseInt() transforme la chaîne de caractères en nombre */
```

```
alert(result);
```

```
</script>
```




Eléments de base

La condition if else

La fonction **confirm()** permet une interaction de l'utilisateur à l'exécution du code

(true si OK, false si Annuler) :

Exemple

```
<script>
```

```
if (confirm(Réaliser le projet avec java ?')) {
```

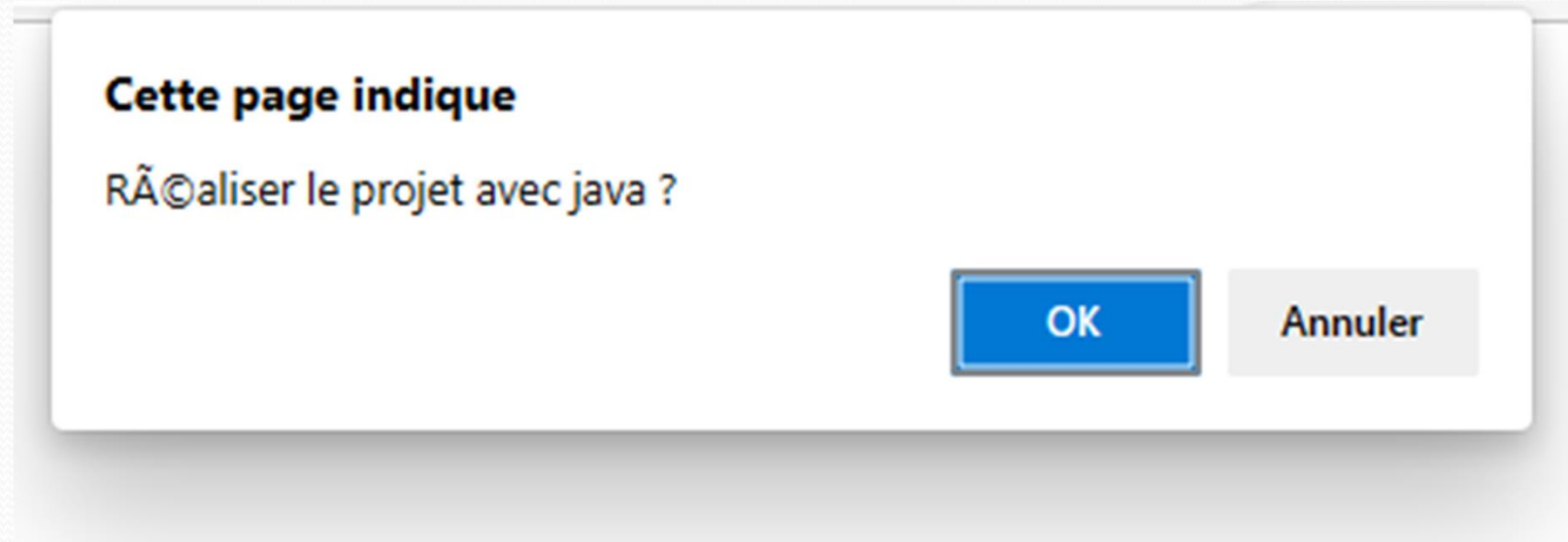
```
  alert('Le bon choix !');
```

```
}</script>
```

La condition if else

La fonction **confirm()** permet une interaction de l'utilisateur à l'exécution du code

(true si OK, false si Annuler) :





Eléments de base

La condition if else

Exemple

```
<script>
var nbre ;
  nbre= prompt("Entrez un nombre ");
nbre= parseInt(nbre);
if (nbre == 0) {
  alert('un nombre null. ');
}
  else if ( nbre <= 0)
  alert("NOMBRE NEGATIF ");
  else {
  alert("NOMBRE POSITIF.");
}</script>
```



Eléments de base

Les boucles

La boucle while

Syntaxe

<script>

while (condition) {

instruction_1; instruction_2; instruction_3;

} </script>



Eléments de base

Les boucles

La boucle do while (peu utile)

Exemple

```
<script>
```

```
do {
```

```
instruction_1; instruction_2; instruction_3;
```

```
} while (condition);
```

```
</script>
```



Eléments de base

Les boucles

La boucle for (très utile)

`<script>`

`for (initialisation; condition; incrémentation) {`

`instruction_1;`

`instruction_2;`

`instruction_3;`

`} </script>`



Les fonctions

Les fonctions

- ❑ Pour définir une fonction qui ne retourne pas de valeur, la syntaxe est la suivante :

***function Nom_Function (Arg1, ..., ArgN) {
Instruction 1 InstructionM}***

- ❑ Pour définir une fonction qui retourne une valeur, la syntaxe est la suivante :

***function Nom_Function (Arg1,..., ArgN) {
Instruction 1
InstructionM
return valeur }***



Les fonctions

Exemple

Une fonction peut retourner une seule valeur, stockée dans une variable :

```
<script>
```

```
function bonjour() {  
  return 'Bonjour !';  
}
```

```
alert(bonjour());
```

```
</script>
```



Les fonctions

Exemple

<head>

<title>Déclaration des fonctions</title>

<script>

function Cours(){

alert("Cours DAW!");

}

Cours();

</script>

</head>

Cours DAW



Les fonctions

Exemple

<head>

<title>Déclaration des fonctions</title>

<script>

function CR(Cours, heure){

alert("Le cours "+Cours+"démarre à"+heure);

}

CR("DAW" , 9);

</script>

</head>

**Le cours DAW
démarre à 9**



Les fonctions

Exemple

<head>

<title>Déclaration des fonctions</title>

<script>

function CR(Cours, heure){

alert("Le cours "+Cours+" démarre à "+heure);

}

CR("DAW");

</script>

</head>

**Le cours DAW démarre à
undefined**



Les fonctions

Exemple

<head>

<title>Déclaration des fonctions</title>

<script>

function CR(Cours, heure){

if(typeof(heure) == "undefined"){

alert("Le cours "+Cours);

else

alert("Le cours "+Cours+" démarre à "+heure);

}}

CR("DAW"); CR("DAW" ,9); </script> </head>

Exemple

<head>

<title>Déclaration des fonctions</title>

<script>

function nom(){

return prompt("Entrez votre nom");

}

function Prenom(){

return prompt("Entrez votre Prenom");

}

alert("Je suis "+nom()+Prenom());

</script> **</head>**



Les fonctions

Les fonctions prédéfinies

`parseFloat`

La fonction *parseFloat* analyse la chaîne de caractères qu'on lui passe en paramètre et, dans la mesure du possible, la traduit en valeur numérique réelle. Par exemple, la chaîne '1.414' sera traduite en valeur numérique 1.414



Les fonctions

Les fonctions prédéfinies

IsNaN

En JavaScript, lorsqu'une expression numérique ne peut pas être évaluée, elle prend la valeur *NaN* (pour Not a Number). C'est le cas, par exemple, lors d'une division par 0 ou lorsqu'on essaie de transformer en nombre une chaîne de caractères qui n'a rien de numérique.

Syntaxe : *isNaN(expression)*

expression est l'expression numérique qu'on souhaite tester.

isNaN retourne la valeur booléenne *true* ou *false*.



Les fonctions

Les fonctions prédéfinies(IsNaN)

Exemple :

```
<script>
var chaine1 = '3.14'
var chaine2 = 'Rien'
Res1=parseFloat(chaine1)
Res2=parseFloat(chaine2)
if (isNaN(Res1)==true)
document.write('La valeur obtenue pour parseFloat de la chaine
'+chaine1+ ' est non numerique.<BR>')
else
document.write ('La valeur numerique de parseFloat de la chaine
'+chaine1+ ' est numérique : ' + Res1 + '<BR>')
if (isNaN(Res2)==true)
document.write('La valeur obtenue pour parseFloat de la chaine '+chaine2+
' n\'est pas numerique.<BR>')
else
document.write ('La valeur numerique de parseFloat de la chaine
'+chaine1+ ' est numérique' + Res2 + '<BR>') </script>
```

Les fonctions

test.html



Fichier | C:/Users/PC/Desktop/TP2-DAW/test.html



La valeur numerique de parseFloat de la chaine 3.14 est numérique : 3.14.
La valeur obtenue pour parsefloat de la chaine Rien n'est pas numerique.



Les objets et les tableaux

Les OBJETS



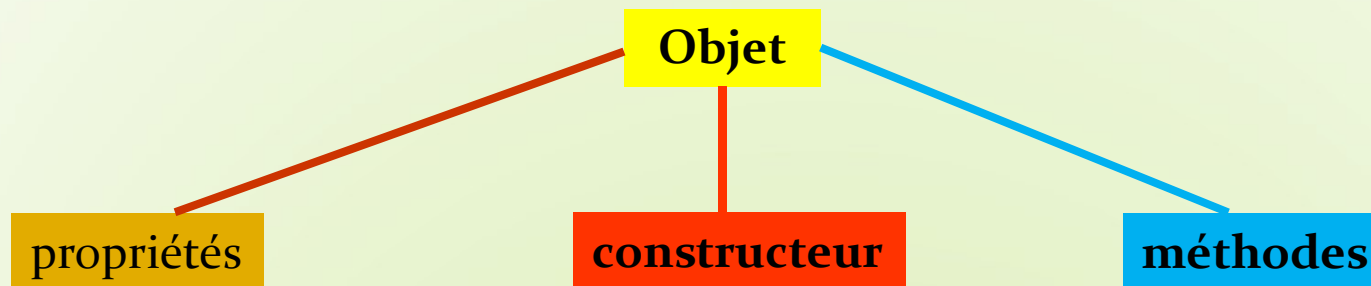
Les objets et les tableaux

- le JavaScript est un langage orienté objet.
- Le JavaScript met à notre disposition des objets natifs :
 - ✓ Number: l'objet qui gère les nombres ;
 - ✓ Boolean: l'objet qui gère les booléens ;
 - ✓ String: l'objet qui gère les chaînes de caractères.

Outre les objets natifs, le JavaScript nous permet de fabriquer nos propres objets.

Les objets et les tableaux

- Les objets en JavaScript vont contenir des **propriétés**, **méthodes** et **constructeur**,



Une propriété est une variable contenue dans l'objet, elle contient des informations nécessaires au fonctionnement de l'objet.



Une méthode est une fonction contenue dans l'objet, et qui permet de réaliser des opérations sur le contenu de l'objet.

Un constructeur est un code qui est exécuté quand on utilise un nouvel objet



L'objet tableaux

Les tableaux



L'objet tableaux

Les tableaux

- ❑ Après Number, String et Boolean, Array est un 4^{ème} objet natif de JavaScript,
- ❑ Un tableau (array en anglais) est une variable qui contient plusieurs valeurs, appelées items. Chaque item est accessible au moyen d'un indice dont la numérotation commence à partir de **0**
- ❑ Un tableau peut accueillir des éléments de types différents.



L'objet tableaux

Les tableaux

Exemple :

```
<script>  
var myArray = ['R', 'M', 'A', 'J', 'G'];  
    myArray[1] = 'P';  
    alert(myArray[1]);  
</script>
```




Les objets et les tableaux

Les tableaux

Une autre déclaration avec Array

```
var jours = new Array();  
var jours = new Array("Lundi", "Mardi", "Mercredi", "Jeudi",  
"Vendredi", "Samedi", "Dimanche");
```



Les objets et les tableaux

Les tableaux

Afficher un tableau :

```
alert(jours)
```

Taille d'un tableau :

```
alert(jours.length)
```

```
// Syntaxe : tableau.méthode();
```

Accéder aux items :

```
alert(jours[1]);
```




Les objets et les tableaux

Opérations sur les tableaux :

// Syntaxe : tableau.méthode();

On peut ajouter des items avec la méthode **push()** (ajout à la fin)

```
<script>
```

```
var myArray = ['R', 'M'];
```

```
myArray.push('A'); // Ajoute « A » à la fin du tableau
```

```
myArray.push('J', 'G'); // Ajoute « J » et « G » à la fin du tableau
```

```
</script>
```

Les objets et les tableaux

Opérations sur les tableaux

La méthode unshift() fonctionne comme push(), excepté que les items sont ajoutés au début du tableau. Les méthodes shift() et pop() retirent respectivement le premier et le dernier élément du tableau.

Exemple

```
<script>  
var myArray = ['R', 'M', 'A', 'J', 'G'];  
  myArray.shift(); // Retire « R »  
  myArray.pop(); // Retire « G »  
alert(myArray); // Affiche « M,A,J »  
</script>
```




Les objets et les tableaux

Opérations sur les tableaux

Inverser le tableau : *myArray*=['R', 'M', 'A', 'J', 'G'];

myArray = *myArray.reverse()*; //Résultat: 'G', 'J', 'A', 'M', 'R',

Sélectionner une partie :

myArray = *myArray.slice*(0, 2); //Résultat: 'G', 'J'

Trier un tableau :

myArray = *myArray.sort()*; //Résultat: 'G', 'J'

Les objets et les tableaux

Opérations sur les tableaux

L'inverse de `split()`, c'est-à-dire créer une chaîne de caractères depuis un tableau, se nomme **`join()`** :

Exemple

```
<script>  
var cString = cArray.join('-');  
alert(cString);  
</script>
```

La chaîne de caractères va être créée, séparée par un tiret.

Les objets et les tableaux

Parcourir un tableau

On peut parcourir un tableau avec for ;

Exemple

```
<script>  
var myArray = ['R', 'M', 'A', 'J', 'G'];  
for ( i = 0; i < myArray.length; i++)  
    alert(myArray[i]);  
</script>
```

Les objets et les tableaux

Les objets littéraux

On peut remplacer l'indice par un identifiant comme suit :

Exemple :

```
<script>
var module = {
  cours: 'tw',
  TD: 'creation de site',
  TP: 'realisation', };
alert(module['TP']);
alert(module.cours);
module['expose'] = 'sur les sites sociaux';
alert(module.expose); </script>
```


Les objets et les tableaux

Les objets littéraux

Parcourir un objet avec for in

On ne peut pas parcourir l'objet avec for, parceque for s'occupe d'incrémenter des variables numériques. Là on fournit une variable-clé pour le parcours

Exemple :

```
<script>
var module = {
  cours: 'tw',
  TD: 'creation de site',
  TP: 'realisation',
};
for (var id in module)
```

Les objets et les tableaux

Les objets littéraux

❑ pour récupérer la valeur d'un champ on peut utiliser:

`module.cours; ou module[cours];`

❑ Ajouter des items:

`module['Unite'] = 'IoT'; ou module.Unite = 'IoT';`



L'objet String

Les chaines de caractères

Les objets et les Chaines de caractères

Opérations sur les chaines de caractères

Exemple

```
<script>  
var cString = 'cours TD TP '  
</script>
```

Taille :

`alert(cString.length);` **Résultat** :11

Caractère à un indice donné :

`cString.charAt(0);` **Résultat** :c

Sélectionner une partie :

`cString.slice(2, 4);` **Resultat** :ur

Les objets et les Chaines de caractères

Opérations sur les chaines de caractères

Exemple

```
<script>  
var cString = 'cours TD TP '  
</script>
```

Écriture en minuscule :

```
cString = cString.toLowerCase();
```

Écriture en majuscule :

```
cString = cString.toUpperCase();
```

Les objets et les Chaines de caractères

Opérations sur les chaines de caractères

Les chaînes de caractères possèdent une méthode **split()** qui permet de les découper en un tableau, en fonction d'un séparateur.

Exemple

```
<script>
var cString = 'cours TD TP '
cArray = cString.split(' ');
alert(cString);
alert(cArray);
</script>
```


Les objets et les Chaines de caractères

Opérations sur les chaines de caractères : **charAt(x)**

La méthode **charAt(x)** (avec A majuscule) permet de retourner le caractère qui se trouve à la position x passé en paramètre

Exemple :

```
Str=" Bonjour " ;
```

```
Str.length; //7
```

```
Str.charAt(0) ; //B
```

Les objets et les Chaines de caractères

Opérations sur les chaines de caractères : **indexOf()**

- ✓ La méthode **indexOf()** renvoie l'indice de la première occurrence de la valeur cherchée au sein de la chaîne courante.
- ✓ La méthode **indexOf()** peut accueillir un deuxième paramètre qui est un entier qui indique à partir de quel position de la chaîne on commencera la recherche du caractère passé en premier paramètre.
- ✓ Elle renvoie **-1** si la valeur cherchée n'est pas trouvé

Les objets et les Chaines de caractères

Opérations sur les chaines de caractères : `indexOf()`

Exemple :

```
Str=" Bonjour " ;  
Str.indexOf(" x "); // -1  
Str.indexOf(" o"); // 1  
Str.indexOf(" o",2); // 4
```

Les objets et les Chaines de caractères

Opérations sur les chaines de caractères : substr et substring

Les deux fonctions sont utilisées pour obtenir la partie spécifiée de la string, mais il y a une légère différence entre elles.

La fonction str.substr() renvoie le nombre spécifié de caractères à partir de l'index spécifié à partir de la string donnée. Syntaxe:

str.substr(start, len)

La fonction str.substring() Cette méthode récupère les caractères d'une string, entre deux indices spécifiés, et renvoie la nouvelle string. Syntaxe:

str.substring(start, end)

Les objets et les Chaines de caractères

Opérations sur les chaines de caractères : substr et substring

Exemple :

Str=« Bonjour » ;

Str.substring(0,3); //Bon

Str.substring(3,6); //jou

Str.substr(0,3); //Bon

Str.substr(3,3); //4

Exemple d'application

Ecrire le script associé à A, B et C

A. Soit le tableau suivant

var tab = [AA,AB,AC,AD,AE,BA,BB,BC]

- a) Cette instruction contient des fautes. Lesquelles ?
- b) Donner une autre déclaration du tableau « tab »
- c) Comment retrouver le dernier élément de cette liste ?
- d) Partant de la liste ci-dessus que vous aurez corrigée, afficher tous les éléments
- e) Afficher tous les éléments en deux colonne

Exemple d 'application

Ecrire le script associé à A, B et C

B.

Soit une chaine de caractères `myString` = 'Ceci est une chaîne de caractères'

1. Donner la taille de cette chaine
2. Convertir la chaine en majuscule
3. Découper cette chaine en tableau (soit `TabString` ce tableau)
4. Ajouter « JavaScript » au tableau obtenu
5. Afficher le 6eme élément du tableau
6. Afficher le 3eme mot de la chaine `myString`

Exemple d 'application

Ecrire le script associé à A, B et C

C.

On veut enregistrer les noms des étudiants demandés par l'utilisateur, **écrire les scripts dans les deux cas suivants :**

1. Les noms seront stockés dans une chaîne de caractères séparée par un blanc
2. Les noms seront stockés dans un tableau et les affichés par la suite séparée par un blanc

Exemple d 'application

Remarques:

Différence entre slice et splice

??????????????



Fin de la 1^{ère} Partie Chapitre 4 « JavaScript »