

PRATIQUER LA PROGRAMMATION EN JAVASCRIPT



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Scripting

- Les langages de script sont généralement exécutés à partir de fichiers contenant le code source du programme qui sera interprété
- Les langages de script les plus connus sont :
 - Script shell : sh ; bash ; ksh ; zsh ; csh ; tcsh ; fish pour GNU/Linux et Unix ;
 - Cmd (anciennement Command), Windows PowerShell pour le monde Windows ;
 - JavaScript ;
 - VBScript ;
 - Matlab
 - Perl
 - Python



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Scripting

- JavaScript est un langage de programmation qui ajoute de l'interactivité à votre site web

- Exemple :
 - Jeux,
 - Réponses quand on clique sur un bouton
 - Validation des données entrées dans des formulaires,
 - Composition dynamique,
 - Animations



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Scripting

La syntaxe du langage Javascript s'appuie sur le modèle de Java et C

- L'insertion des espaces peut s'effectuée n'importe où dans le script
- Chaque commande doit être terminée par un point-virgule (;).
- Un nombre à virgule est séparé par un point (.) et non par une virgule
- Le langage Javascript y est **sensible à la casse**



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Scripting

Activer JavaScript

- Pour Google Chrome :
Accédez au menu Chrome et sélectionnez Préférences.
Sélectionnez l'onglet Options avancée.
Sous Confidentialité, cliquez sur Paramètres de contenu.
Sélectionnez l'onglet JavaScript et choisissez l'option Autoriser tous les sites à utiliser JavaScript.
Fermez cette fenêtre.
- Pour les autres navigateur
.....
.....

<https://www.enable-javascript.com/fr/>



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Types et déclarations des variables

Déclaration des variables

- Les variables sont des conteneurs dans lesquels on peut stocker des variables.
- Pour déclarer une variable:
`let maVariable;`
- Affectation d'une valeur à une variable:
`maVariable = 'test';`
- On peut déclarer une variable comme suite: `let maVariable = 'test';`



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Types et déclarations des variables

Déclaration des variables

- La portée:

- **Variables locales**

Une variable est dite « locale » lorsqu'elle est définie à l'intérieur d'une fonction (donc localement).

- **Variables globales**

Une variable est globale si elle est définie dans l'espace global d'un script, c'est-à-dire en dehors de toute fonction.



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Types et déclarations des variables

Déclaration des variables

- « **let** » permet de déclarer des variables dont la portée est limitée à celle du bloc dans lequel elles sont déclarées.
- « **var** » permet de définir une variable globale ou locale à une fonction (sans distinction des blocs utilisés dans la fonction).

Exemple1

```
for (let i = 0; i < 10; i++) {  
  console.log(i); // 0, 1, 2, 3, 4 ... 9  
}  
console.log(i); // i n'est pas défini!!!
```

Exemple2

```
var a = 5; var b = 10;  
if (a === 5) {  
  let a = 4;  
  var b = 1;  
  console.log(a); // 4  
  console.log(b); // 1  
}  
console.log(a); // 5  
console.log(b); // 1
```





LES ÉLÉMENTS DE BASE DE JAVASCRIPT



Types et déclarations des variables

Déclaration des constants


- Les variables définies avec **const** se comportent comme des variables **let**, sauf qu'elles ne peuvent pas être réaffectées.
- Exemples:

`const PI = 3.141592653589793;` 

`const PI;`
`PI = 3.14159265359;` 

`PI = 3.14;` // erreur 
`PI = PI + 10;` // erreur 

Objet | | Array

`const car = {type:"Fiat", model:"500", color:"white"};` `car.color = "red";` 

`car = {type:"Volvo", model:"EX60", color:"red"};` 



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Les commentaires

- De la même façon qu'en CSS, il est possible d'intégrer des commentaires dans du code JavaScript :
 - `/* Voici un commentaire qui peut être sur plusieurs lignes */`
 - `// Voici un commentaire`



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Les opérateurs

- Un opérateur est un symbole mathématique qui produit un résultat en fonction de plusieurs valeurs (la plupart du temps on utilise deux valeurs et un opérateur).
- Le tableau suivant liste certains des opérateurs les plus simples ainsi que des exemples que vous pouvez tester dans votre console JavaScript



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Les opérateurs

Opérateur	Explication	Symbole(s)	Exemple
Somme / Concaténation	Il peut être utilisé pour calculer la somme de deux nombres ou pour concaténer (coller) deux chaînes ensemble.	+	6 + 9; "Coucou " + "monde !";
Soustraction, multiplication, division	Les opérations mathématiques de base.	-, *, /	9 - 3; 8 * 2; // pour multiplier, on utilise une astérisque 9 / 3;
Opérateur d'affectation	il permet d'affecter une valeur à une variable.	=	let maVariable = 'Bob';
Opérateur d'identité	Il permet de tester si deux valeurs sont égales et il renvoie un booléen true/false comme résultat.	===	let maVariable = 3; maVariable === 4;
Opérateur de négation et opérateur d'inégalité	Souvent utilisé avec l'opérateur d'égalité, l'opérateur de négation est l'équivalent, en JavaScript, d'un NON logique (il transforme la valeur true en false et vice versa)	!, !==	let myVariable = 3; !(myVariable === 3); let maVariable = 3; maVariable !== 3;



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Types de données

Variable	Explication	Exemple
chaîne de caractères	Une chaîne de texte. Pour déclarer une chaîne de caractères, il faut placer la valeur entre des quotes (').	<code>let maVariable = 'Bob';</code>
Nombre	Un nombre. Les nombres n'ont pas de quote.	<code>let maVariable = 10;</code>
Booléen	Une valeur qui signifie vrai ou faux. true/false sont des mots-clés spéciaux en JavaScript, il n'est pas nécessaire d'utiliser des quotes.	<code>let maVariable = true;</code>
Tableau	Une structure qui permet de stocker plusieurs valeurs dans une seule variable.	<code>let maVariable = [1,'Bob','Patrick',10];</code>
Objet	Un objet peut être n'importe quoi. Un objet est un ensemble de propriétés qui peut être enregistré dans une variable. C'est la structure la plus utilisée et la plus flexible.	<code>let maVariable = document.querySelector('h1');</code>

Le type d'une variable dépend de la valeur stockée dans cette variable. Pas de déclaration de type.



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Types de données

JavaScript inclut aussi deux types de données spéciaux :

- **Null** : possède une seule valeur, **null**, qui signifie l'absence de données dans une variable
- **Undefined** : possède une seule valeur, **undefined**. Une variable dont le contenu n'est pas clair car elle n'a jamais stocké de valeur, pas même **null** est dite non définie (undefined).



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

L'instruction if

- L'instruction **if** exécute une instruction si une condition donnée est vraie.
- Si la condition n'est pas vérifiée, il est possible d'utiliser une autre instruction.

```
if (condition)
  instruction1
[else instruction2]
```

```
if (condition1)
  instruction1
else
  if (condition2)
    instruction2
  else
    if (condition3)
      ...
    ...
```

```
if (condition1)
  instruction1
else if (condition2)
  instruction2
else if (condition3)
  instruction3
...
else
  instructionN
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

L'instruction switch

L'instruction «switch» permet de choisir le traitement approprié en fonction de la valeur d'une expression.

```
switch ( expression )  
{  
  [ case etiquette :  
    [ instruction ]... ]...  
  [ default] :  
    [ instruction ]... ]  
}
```

```
<script>  
  var heure = 10;  
  switch(heure){  
    case 10 :  
      alert("Il est 10h");  
      break;  
    case 11 :  
      alert("Il est 11h");  
      break;  
    case 12 :  
      alert("Il est 12h");  
      break;  
    default :  
      alert("Je ne reconnais pas cette heure");  
  }  
</script>
```




LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Les boucles

- L'instruction do...while
- L'instruction for
- L'instruction for...in



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Les boucles

L'instruction **do...while**

- Permet de créer une boucle qui exécute une instruction jusqu'à ce qu'une condition de test ne soit plus vérifiée.
- La condition est testée après que l'instruction soit exécutée, l'instruction définie dans la boucle est donc exécutée au moins une fois.

```
Do  
    instruction  
while (condition);
```

- Exemple

```
var i = 0;  
do {  
    i += 1;  
    console.log(i);  
} while (i < 5);
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Les boucles

L'instruction for

- L'instruction **for** crée une boucle composée de trois expressions optionnelles séparées par des points-virgules et encadrées entre des parenthèses qui sont suivies par une instruction (généralement une instruction de bloc) à exécuter dans la boucle.

```
for ([initialisation]; [condition]; [expression_finale])  
    instruction
```

- Exemple

```
var str = "";  
for (var i = 0; i < 9; i++)  
    { str = str + i;}  
console.log(str);
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Les boucles

L'instruction for...in

- permet d'itérer sur les propriétés énumérables d'un objet. Pour chaque propriété obtenue, on exécute une instruction (ou plusieurs grâce à un [bloc](#) d'instructions).

```
for (variable in objet)
    { ... }
```

- Exemple:

```
var string1 = "";
var object1 = {a: 1, b: 2, c: 3};
for (var property1 in object1) {
    string1 = string1 + object1[property1];
}
console.log(string1);
// expected output: "123"
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Les boucles

Break/continue

- L'instruction **break** permet de terminer la boucle en cours ou l'instruction [switch](#) ou [label](#) en cours et de passer le contrôle du programme à l'instruction suivant l'instruction terminée.
- L'instruction **continue** arrête l'exécution des instructions pour l'itération de la boucle courante ou de la boucle étiquetée. L'exécution est reprise à l'itération suivante.

- Exemples:

```
var i = 0;
while (i < 6)
    { if (i === 3) { break; }
      i = i + 1;}
console.log(i);
```

```
for (var i = 0; i < 10; i++)
{
    if (i === 3) { continue; }
    text = text + i
;}
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Fonctions

- La déclaration **function** permet de définir une fonction et les paramètres que celle-ci utilise.

```
function nom([param1,[, param2,[..., paramN]]]) {  
  [instructions]  
}
```

- Exemple

Déclaration d'une
fonction :

```
function cube (x)  
{  
  return x*x*x;  
}
```

Déclaration d'une procédure

```
function bienVnu(titre, nom)  
{  
  alert("Bienvenue à " + titre + " " + nom + " !");  
}
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Fonctions

- Valeur par défaut

```
function myFunction(x, y = 10)
{
    // y est 10 si n'est pas passé en paramètres ou or undefined
    return x + y;
}
```

```
myFunction(5); // valeur de retour est 15
myFunction(5,3); // valeur de retour est 8
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Fonctions

Les fonctions auto-invoquées

- Les fonctions auto-invoquées (self-invoked functions) vont être des fonctions qui vont s'appeler elles-mêmes.
- Exemple

```
<body>
  <h1>Les fonctions</h1>
  <script>
    (function(x){
      x = 10;
      alert('Carré de ' + x + ' = ' + x*x);
    })();
  </script>
</body>
```

rajouter un couple de parenthèses autour de la fonction et un second après le code de la fonction.



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Fonctions

Les fonctions récursives

```
function factoriel(n){  
    if(n==1){  
        return 1;  
    }  
    return n*factoriel(n-1);  
}  
document.write(factoriel(3)); // retourne 3*2*1
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Fonctions

Les fonctions globales

Fonctions	Description
eval()	Evalue une chaîne
isFinite ()	Détermine si une valeur est un nombre légal fini
isNaN ()	Détermine si une valeur est un nombre illégal
Number()	Convertit la valeur d'un objet en nombre
parseFloat()	Analyse une chaîne et retourne un nombre à virgule flottante
parseInt ()	Analyse une chaîne et retourne un entier
String ()	Convertit la valeur d'un objet en chaîne



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Fonctions

Les fonctions globales

Fonctions	Description
<code>decodeURI()</code>	décode un URI
<code>decodeURIComponent()</code>	décode un composant URI
<code>encodeURIComponent ()</code>	Encode un URI
<code>encodeURIComponent()</code>	Encode un composant URI
<code>escape()</code>	Déconseillé dans la version 1.5. Utilisez plutôt <code>encodeURIComponent ()</code> ou <code>encodeURIComponent ()</code>
<code>unescape ()</code>	Déconseillé dans la version 1.5. Utilisez <code>decodeURI ()</code> ou <code>decodeURIComponent ()</code> à la place



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Fonctions

Les fonctions globales

```
var x = 2;  
var y = 39;  
var z = "42";  
eval("x + y + 1");
```

```
let nb= 1/0;  
document.write(isFinite(nb));
```

```
let nb= 1/"dan";  
document.write(isNaN(nb));
```

```
let saisie= "3";document.write(parseInt(saisie)+6);
```

```
let nb1= "3.0";  
let nb2= "3.33"  
document.write(parseFloat(nb1) + '<br>');  
document.write(parseFloat(nb2));
```

```
let monUrl= "//monsite/première page.html";  
document.write(encodeURIComponent(monUrl));
```

```
let monUrl=  
"http://monsite/premi%C3%A8re%20page.html";  
document.write(decodeURI(monUrl));
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Intégration du script dans du code HTML

Il y a deux manières d'ajouter du code JavaScript dans une page :

- en liant depuis la page HTML un fichier externe, dans lequel sont placées les instructions JavaScript :

```
<script type="text/javascript" src="monscript.js"></script>
```

- en ajoutant le code JavaScript à l'intérieur de l'élément script :

```
<script type="text/javascript"> // Mon code Javascript ... </script>
```

➔ Traditionnellement il était d'usage de placer la balise `<script>` entre les tags `<head>` et `</head>`, actuellement recommandé de **la placer en fin de document juste avant `</body>`** pour ne pas bloquer le chargement de la page, et exécuter les scripts uniquement lorsque le DOM est prêt.



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Intégration du script dans du code HTML

Insertion pour exécution directe

Le code s'exécute automatiquement lors du chargement de la page HTML dans le navigateur en même temps que le contenu de la page HTML s'affiche à l'écran.

```
<html>
<head>
<title>..... </title>
</head>
<body>
    <script language="JavaScript">
        alert('bonjour');
    </script>
</body>
</html>
```

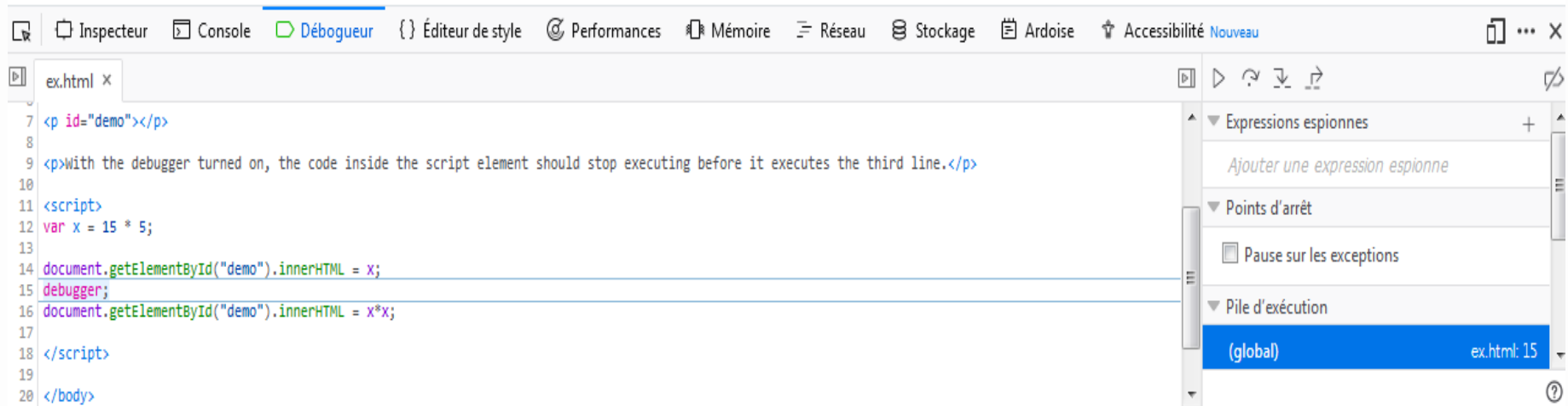


LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Débogage

75

With the debugger turned on, the code inside the script element should stop executing before it executes the third line.





LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Débogage

- L'instruction de débogage arrête l'exécution de JavaScript et appelle (si disponible) la fonction de débogage.
- L'utilisation de l'instruction de débogage a la même fonction que la définition d'un point d'arrêt dans le code.
- Quand on active le débogage dans votre navigateur avec la touche F12, puis sélectionnez "Console" dans le menu du débogueur.
- Si aucun débogage n'est disponible, l'instruction de débogage n'a aucun effet.



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Gestion des exceptions

L'instruction **try...catch** regroupe des instructions à exécuter et définit une réponse si l'une de ces instructions provoque une exception.

```
try {  
    instructions_try  
}  
[catch (exception_var_1 if  
condition_1) { // non-standard  
    instructions_catch_1  
}]  
...  
[catch (exception_var_2) {  
    instructions_catch_2  
}]  
[finally {  
    instructions_finally  
}]
```

L'instruction **try** permet de tester un bloc de code pour rechercher des erreurs.

L'instruction **catch** permet de gérer l'erreur.

L'instruction **throw** permet de créer des erreurs personnalisées.

L'instruction **finally** permet d'exécuter du code, après avoir essayé et intercepté, quel que soit le résultat.



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Gestion des exceptions

Exemple2

```
<body>
<p>Donner une valeur dans l'intervalle [5,100]:</p>
<input id="txtValeur" type="text">
<button type="button" onclick="myFunction()">Tester Try</button>
<p id="message"></p>
<script>
    function myFunction() {
        var message, x;
        message = document.getElementById("message");
        message.innerHTML = "";
        x = document.getElementById("txtValeur").value;
        try {
            if(x == "") throw "vide";
            if(isNaN(x)) throw "ce n'est pas un nombre";
            x = Number(x);
            if((x < 5) || (x > 10)) throw "valeur n'appartient pas à l'intervalle [5,10] ";
        }
        catch(err) {
            message.innerHTML = "Input is " + err;
        }
    }
</script>
</body>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Création et syntaxe d'un tableau

```
var colors = new Array(); // tableau vide  
var colors2 = new Array(20); // tableau avec 20 cases  
var colors3 = new Array("red","blue","green"); // 3 cases  
var colors4 = ["red","blue","green"]; // notation littérale
```

```
var prenom=[ 'Pierre', 'Victor', 'Julia', 'Claire'], p="";  
//On stocke les prénoms un par un dans p  
for(var i = 0; i < prenom.length; i++){  
    p += 'Prénom n°' + (i+1) + ' : ' +prenom[i] + '\n';  
}  
//On affiche le tout  
alert(p);
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Des tableaux avec des index textuels

```
<script>
    //On crée un objet littéral

    var prenom1 = {
        prenom1 : 'Pierre',
        prenom2 : 'Victor',
        prenom3 : 'Julia',
        prenom4 : 'Claire'
    };

    //On affiche la valeur associée à "prenom1"
    alert(prenoms.prenom1);
</script>
```

```
for (var clefs in prenom1)
{
    p += clefs + ' : ' + prenom1[clefs] + '\n';
}
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet Array

- Propriétés
 - **Array.length** : retourne le nombre d'éléments d'un tableau
- Méthodes
 - **Array.join()** : converti un tableau en chaîne
 - **Array.pop()** : supprime le dernier élément d'un tableau
 - **Array.push()** : ajoute un élément en fin de tableau
 - **Array.shift()** : supprime le premier élément d'un tableau
 - **Array.unshift()** : ajoute un élément en début de tableau
 - **Array.reverse()** : inverse l'ordre des éléments d'un tableau
 - **Array.sort()** : tri les éléments d'un tableau par ordre croissant



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet String

■ Propriétés

- Cet objet ne possède qu'une seule propriété, `length`, qui renvoie sa longueur.
- **Exemple:** `chaine="Bonjour!";alert(chaine.length);` renvoie 8.

■ Méthodes

Les méthodes de cet objet permettent de manipuler les chaînes de caractères. Elles s'appliquent à toute variable de ce type.

- **`indexOf`** renvoie la position de la première occurrence d'un caractère donné dans une chaîne.

Exemple: `chaine="Bonjour!";alert(chaine.indexOf('o'));` renvoie 1, la numérotation commençant à 0. Si le caractère n'est pas trouvé, la méthode renvoie -1.

- **`lastIndexOf`** renvoie la position de la dernière occurrence d'un caractère donné dans une chaîne.

Exemple: `chaine="Bonjour!";alert(chaine.lastIndexOf('o'));` renvoie 4.

- **`charAt`** renvoie le caractère présent à une certaine position.

Exemple: `chaine="Bonjour!";alert(chaine.charAt(5));` renvoie « u ».



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet String

■ Méthode

- **charCodeAt** : renvoie le code Unicode du caractère présent à une certaine position.
Exemple: `chaine="Bonjour!"; alert(chaine.charCodeAt(5));` renvoie « 117 ».
- **fromCharCode** : renvoie une chaîne correspondant à la séquence de codes Unicode passée en paramètre.
Exemple:
`alert(String.fromCharCode(66, 111, 110, 106, 111, 117, 114, 33))` renvoie « Bonjour! ».
- **toUpperCase** : convertit la chaîne en majuscules.
Exemple: `chaine="Bonjour!"; alert(chaine.toUpperCase);` affiche « BONJOUR! ».
- **toLowerCase** : convertit la chaîne en minuscules.
Exemple: `chaine="Bonjour!"; alert(chaine.toLowerCase);` affiche « bonjour! ».



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet String

■ Méthode

- **substring** extrait une sous-chaîne de la chaîne à laquelle elle s'applique. Cette méthode prend deux entiers positifs en paramètres.

Exemple: `chaine="Bonjour!";alert(chaine.substring(1,4));` renvoie « onj ».

- **substr** : extrait aussi une sous-chaîne, mais elle fonctionne différemment. Le premier paramètre désigne la position de départ, et le second indique la longueur de la sous-chaîne à extraire.

Exemple: `chaine="Bonjour!";alert(chaine.substr(1,4));` renvoie « onjo ».

- **slice** : extrait aussi une sous-chaîne. le premier paramètre désigne la position de départ, et le second indique la position de fin de la sous-chaîne à extraire.

Exemple: `chaine="Bonjour!";alert(chaine.substr(2,4));` renvoie « njo ».



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet String

■ Méthode

- **split** : permet de scinder une chaîne en sous-chaînes. Le résultat est un tableau de chaînes de caractères.

Exemple: `chaine="Bonjour!";tab=chaine.split('o');` renvoie un tableau à une dimension dont les éléments sont respectivement "B", "nj" et "ur!".

- **concat** : concatène la chaîne avec la ou les chaînes passées en argument.

Exemple: `chaine="Bonjour!";alert(chaine.concat(" et à ", "bientôt..."));` renvoie « Bonjour! et à bientôt... ».

- **replace** : remplace un morceau de la chaîne indiqué en paramètre par une autre chaîne indiquée comme second paramètre.

Exemple: `chaine="Bonjour!";alert(chaine.replace("jour", "soir"));` renvoie « Bonsoir! ».

- **includes** teste si une chaîne de caractères donnée est incluse dans une autre, et renvoie un booléen. Par exemple...



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet Date

- Date est un type d'objet qui permet de gérer non seulement les dates, mais aussi les heures et durées. Le constructeur est ce qui permet de créer une nouvelle « instance » de la classe Date.
- Syntaxe :

```
var nouvelleDate = new Date(annee, mois, jour [, heure, minute, seconde, milliseconde]) ;
```
- Les paramètres heure, minute, seconde et milliseconde sont optionnels.
Exemple,: une variable stockant la date du 11 octobre 2003 à 9h32 passée de 34 secondes et 123 millisecondes est créée par...

```
var nouvelleDate = new Date(2003, 9, 11, 9, 32, 34, 123) ;
```
- Si aucun paramètre n'est utilisé, par exemple `auj=new Date;`, la date renvoyée est la date courante.



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet Date

■ Méthodes

- **getFullYear** : renvoie l'année pour la date spécifiée.
 - ✓ **Exemple** : `alert(nouvelleDate.getFullYear())` renvoie 2003.
 - ✓ Cette méthode est complétée par `getUTCFullYear` qui donne l'année dans le Temps Universel en tenant compte du décalage horaire.
- **getMonth** renvoie le mois pour la date spécifiée.
 - ✓ **Exemple** : `alert(nouvelleDate.getMonth())` renvoie 9, la numérotation des mois commençant en janvier à 0.
 - ✓ Cette méthode est complétée par `getUTCMonth` qui donne l'année dans le Temps Universel en tenant compte du décalage horaire.
- **getDate** renvoie le quantième dans le mois pour la date spécifiée.
 - Exemple** : `alert(nouvelleDate.getDate())` renvoie 11. Cette méthode est complétée par `getUTCDate` qui donne l'année dans le Temps Universel en tenant compte du décalage horaire.



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet Date

■ Méthodes

- **getDay** renvoie le jour de la semaine pour la date spécifiée (0=dimanche, 6=samedi).
Exemple: `alert(nouvelleDate.getDay())` renvoie 6. Cette méthode est complétée par `getUTCDay` qui donne l'année dans le Temps Universel en tenant compte du décalage horaire.
- **getHours** renvoie l'heure dans le jour pour la date spécifiée (entre 0 et 23).
Exemple: `alert(nouvelleDate.getHours())` renvoie 9. Cette méthode est complétée par `getUTCHours` qui donne l'année dans le Temps Universel en tenant compte du décalage horaire.
- **getMinutes** renvoie le nombre de minutes pour la date spécifiée.
Exemple: `alert(nouvelleDate.getMinutes())` renvoie 32. Cette méthode est complétée par `getUTCMinutes` qui donne l'année dans le Temps Universel en tenant compte du décalage horaire.
- **getSeconds** renvoie le nombre de secondes pour la date spécifiée.
Exemple: `alert(nouvelleDate.getSeconds())` renvoie 34. Cette méthode est complétée par `getUTCSeconds` qui donne l'année dans le Temps Universel en tenant compte du décalage horaire.



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet Date

■ Méthodes

- **getMilliseconds** renvoie le nombre de millisecondes pour la date spécifiée.
Exemple, `alert(nouvelleDate.getMilliseconds())` renvoie 123. Cette méthode est complétée par `getUTCMilliseconds` qui donne l'année dans le Temps Universel en tenant compte du décalage horaire.
- **getTime** renvoie le nombre de millisecondes écoulées depuis le 1er janvier 1970 jusqu'à la date spécifiée.
- **getTimezoneOffset** renvoie le décalage horaire en minutes par rapport au Temps Universel tel qu'il est défini sur le système d'exploitation. Par exemple, si l'on se trouve dans une zone à GMT+10, la valeur retournée est -600.



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet Number

Méthode	Méthode Description
isFinite ()	Vérifie si une valeur est un nombre fini
isInteger ()	Vérifie si une valeur est un entier
isNaN ()	Vérifie si une valeur est Number.NaN
isSafeInteger ()	Vérifie si une valeur est un entier sûr
toExponential (x)	Convertit un nombre en une notation exponentielle. Ex: 10 → 1.000e+1
toFixed (x)	Formate un nombre avec x nombres de chiffres après le point décimal. (5.56789 → 5.56)
toPrecision (x)	Formate un nombre en x longueur
toString ()	Convertit un nombre en chaîne
valueOf ()	Retourne la valeur primitive d'un nombre



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet Number

Propriété	Description
constructor	Retourne la fonction qui a créé le prototype Number de JavaScript.
MAX_VALUE	Renvoie le plus grand nombre possible en JavaScript
MIN_VALUE	Retourne le plus petit nombre possible en JavaScript
NEGATIVE_INFINITY	Représente l'infini négatif (retourné au débordement)
NaN	Représente une valeur "Not-a-Number"
POSITIVE_INFINITY	Représente l'infini (renvoyé en cas de dépassement de capacité)
prototype	Permet d'ajouter des propriétés et des méthodes à un objet.



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet Math

Propriété	Description
<code>abs (x)</code>	Retourne la valeur absolue de x
<code>acos (x)</code>	Retourne l'arccosine de x, en radians
<code>acosh (x)</code>	Renvoie l'arccosine hyperbolique de x
<code>asin (x)</code>	Retourne l'arcsin de x, en radians
<code>asinh (x)</code>	Renvoie l'arcsine hyperbolique de x
<code>atan (x)</code>	Renvoie l'arctangente de x sous forme de valeur numérique comprise entre $-\pi / 2$ et $\pi / 2$ radians.
<code>atan2 (y, x)</code>	Retourne l'arctangente du quotient de ses arguments
<code>atanh (x)</code>	Retourne l'arctangente hyperbolique de x
<code>cbrt (x)</code>	retourne la racine cubique de x



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet Math

Propriété	Description
ceil (x)	Retourne x, arrondi au nombre entier le plus proche
cos (x)	retourne le cosinus de x (x est en radians)
cosh (x)	renvoie le cosinus hyperbolique de x
exp (x)	Retourne la valeur de E^x
floor (x)	Retourne x, arrondi au nombre entier le plus proche
log (x)	Retourne le logarithme naturel (base E) de x
max (x, y, z, ..., n)	Retourne le nombre avec la valeur la plus élevée
min (x, y, z, ..., n)	Retourne le nombre avec la valeur la plus basse
pow (x, y)	Renvoie la valeur de x à la puissance de y



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet Math

Propriété	Description
random ()	Retourne un nombre aléatoire compris entre 0 et 1
round (x)	arrondit x au nombre entier le plus proche
sin (x)	retourne le sinus de x (x est en radians)
sinh (x)	renvoie le sinus hyperbolique de x
sqrt (x)	retourne la racine carrée de x
tan (x)	Renvoie la tangente d'un angle
tanh (x)	Renvoie la tangente hyperbolique d'un nombre
trunc (x)	Retourne la partie entière d'un nombre (x).



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Interaction avec les objets prédéfinis

Objet Math

Property	Description
E	Returns Euler's number (approx. 2.718)
LN2	Returns the natural logarithm of 2 (approx. 0.693)
LN10	Returns the natural logarithm of 10 (approx. 2.302)
LOG2E	Returns the base-2 logarithm of E (approx. 1.442)
LOG10E	Returns the base-10 logarithm of E (approx. 0.434)
PI	Returns PI (approx. 3.14)
SQRT1_2	Returns the square root of 1/2 (approx. 0.707)
SQRT2	Returns the square root of 2 (approx. 1.414)



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Les boîtes de dialogues

Les boîtes de message ou boîtes de dialogue permettent :

- d'alerter,
- de confirmer,
- de recevoir des informations.

Exemple:

alert() (Affiche un message d'information dans une boîte de dialogue modale)

confirm() (Affiche une boîte dialogue avec un bouton OK et un bouton Annuler)

prompt() (Ouvre une boîte de dialogue avec zone de saisie de texte)



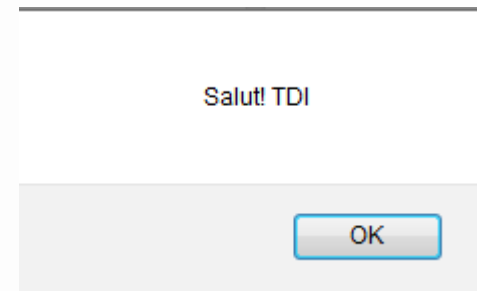
LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Les boites de dialogues

Alerter

- La méthode javascript **Alert(texte)** permet d'avertir l'utilisateur en ouvrant un boite de dialogue avec un bouton "OK".
- Exemple :

```
var message="Salut TDI";  
alert(message);
```





LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Les boites de dialogues

Confirmer

- Affiche le paramètre texte dans un message d'alerte avec 2 boutons de confirmation.
 - Retourne true si le visiteur clique sur Ok.
 - Retourne false si le visiteur clique sur Annuler ou appuye sur la touche Echap.

```
<form>
<input type="button" value="Changer la couleur de fond" onClick="ConfirmMessage()">
</form>
<script type="text/javascript">
  function ConfirmMessage() {
    if (confirm("Voulez-vous changer la couleur de fond de page ?")) {
      // Clic sur OK
      document.body.style.backgroundColor="#ccc";
    }
  }
</script>
```

Voulez-vous changer la couleur de fond de page ?

OK

Annuler



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Les boites de dialogues

Prompt

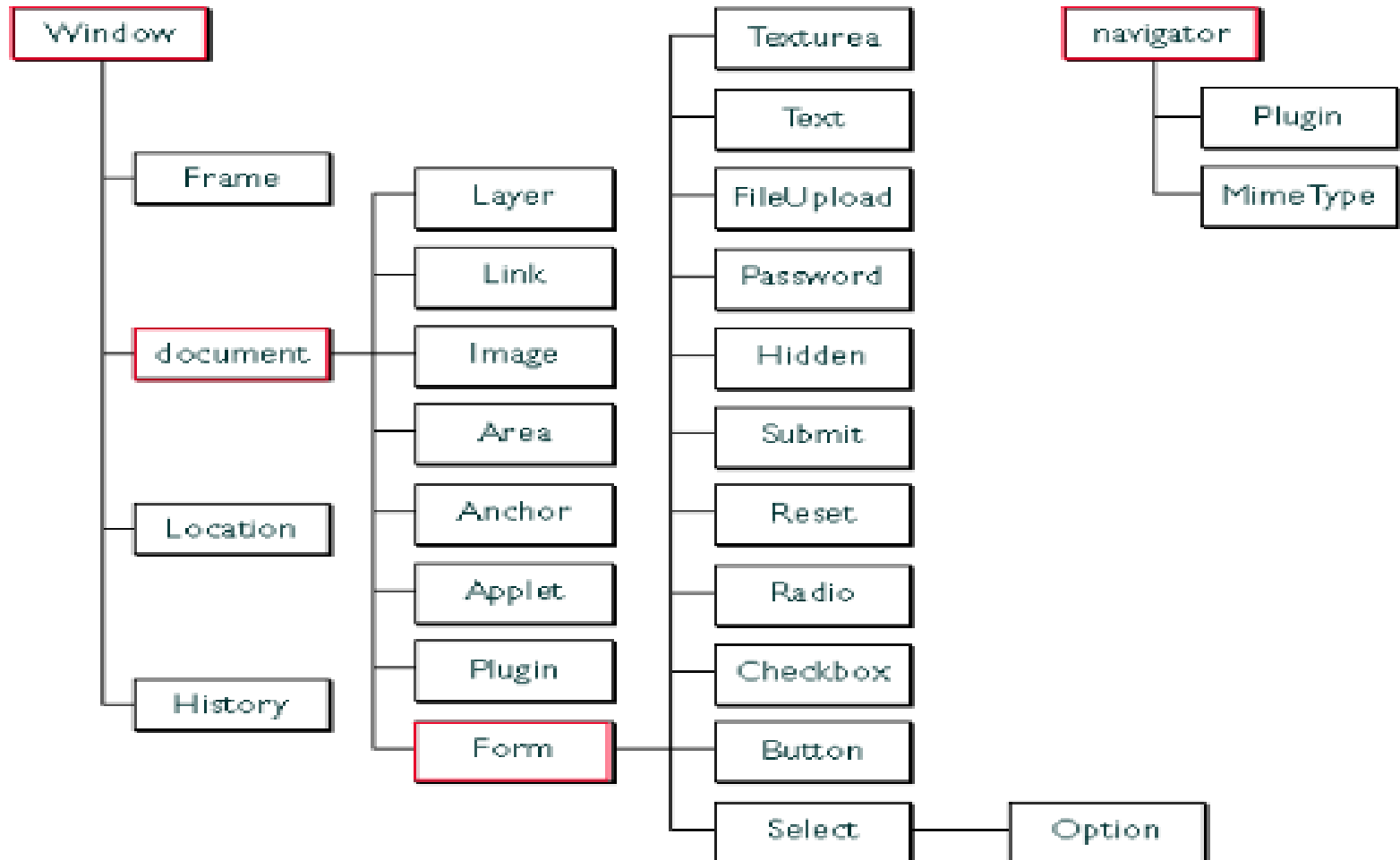
- Affiche une boîte de dialogue le paramètre message, une zone de saisie de texte avec le paramètre par défaut défaut et 2 boutons Valider et Annuler.
 - L'utilisateur peut saisir du texte.
 - la méthode retourne le texte saisi si le visiteur clique sur Valider et retourne null si le visiteur clique sur Annuler ou appuie sur la touche Echap.

```
<form name="mon_formulaire">
<input type="button" value="Saisir" onClick="PromptMessage()">
</form>
<script type="text/javascript">
  function PromptMessage() {
    var saisie = prompt("Saisissez votre texte :", "Texte par défaut")
    console.log(saisie);
  }
</script>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires





LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Objet window

- Propriétés
 - **closed** (Indicateur de fermeture de fenêtre)
 - **defaultStatus** (Texte par défaut de la barre de statut)
 - **name** (Nom de la fenêtre)
 - **opener** (Fenêtre ouvrante)
 - **screenLeft** (Position horizontale en pixels de la fenêtre navigateur)
 - **screenTop** (Position verticale en pixels de la fenêtre navigateur)
 - **status** (Texte de la barre de statut)



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Objet window

■ Méthodes

- **Alert():** Affiche un message d'information dans une boîte de dialogue modale
- **blur():** Enlève le focus à la fenêtre
- **clearInterval():** Arrête l'exécution d'un traitement à intervalle régulier
- **clearTimeout():** Suspend la minuterie identifiée par timer créée avec setTimeout()
- **close():** Ferme la fenêtre
- **confirm():** Affiche une boîte dialogue avec un bouton OK et un bouton Annuler
- **focus():** Donne le focus à la fenêtre
- **getComputedStyle():** Retourne le style CSS de l'élément HTML du DOM
- **getSelection():** Retourne le texte sélectionné par l'utilisateur dans la page
- **home():** Ouvre la page de démarrage du navigateur
- **moveBy():** Déplace la fenêtre
- **moveTo():** Positionne la fenêtre
- **open():** Ouvre un popup
- **print():** Imprime la page en cours



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Objet window

▪ Méthodes

- **print()** :Imprime la page en cours
- **prompt()** :Ouvre une boîte de dialogue avec zone de saisie de texte
- **resizeBy()** :Agrandit ou réduit la taille de la fenêtre de deltaX et deltaY pixels
- **resizeTo()** :Redimensionne une fenêtre
- **scrollBy()** :Déroule les ascenseurs
- **scrollTo()** :Positionne les ascenseurs
- **setInterval()** :Lance la fonction fonc() à intervalle régulier défini par delai, exprimé en millisecondes
- **setTimeout()** (Déclenche une minuterie et exécute fonc() après un délai exprimé en millisecondes
- **stop()** Arrête le chargement de la page



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Objet window

▪ Sous-objets

- **console** (Objet console pour afficher des informations dans la console du navigateur, en particulier avec log())
- **document** (Objet document visible sur la page qui donne accès à tout le DOM)
- **external** (Objet restreint pour Internet Explorer)
- **frames** (Tableaux des frames)
- **history** (Objet permettant de naviguer dans l'historique)
- **localStorage** (Objet de type Storage permettant l'accès au stockage de données permanent du navigateur)
- **location** (Objet contenant les informations sur l'adresse de la page)
- **parent** (Fenêtre mère du jeu de frames)
- **performance** (Objet pour la mesure et l'optimisation des performances du navigateur)
- **screen** (Information sur l'écran)
- **self** (Fenêtre en cours)
- **sessionStorage** (Objet de type Storage permettant l'accès au stockage de données de session du navigateur)
- **top** (Fenêtre initiale du jeu de frames)



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Object document

- Propriétés
 - **alinkColor** :Couleur des liens
 - **bgColor** :Couleur de fond de page)
 - **characterSet** :Propriété contenant le type d'encodage du document
 - **cookie** :Chaîne contenant les cookies du domaine
 - **defaultCharset** :Jeu de caractères
 - **domain** :Domaine de l'adresse de la page
 - **fgColor** :Couleur du texte
 - **fileSize** :Taille de la page en octets
 - **lastModified**:Date de dernière modification
 - **location** :Url de la page
 - **mimeType** :Type de document
 - **protocol** :Type de protocole de l'url
 - **referrer** :Adresse de la page d'origine
 - **title** :Titre de la page
 - **URL** :Adresse de la page
 - **URLUnencoded**:Adresse codée de la page
 - **vlinkColor** :Couleur des liens visités



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Object document

■ Méthodes

- **createElement()** :Crée et retourne un élément HTML à partir de sa balise (tag)
- **getElementById()** :Retourne un objet du DOM à partir de son identifiant
- **getElementsByClassName()** :Retourne la liste des éléments HTML à partir de leur classe
- **getElementsByName()** :Retourne un tableau d'éléments HTML ayant l'attribut name qui vaut nom
- **getElementsByTagName()** :Retourne la liste des éléments HTML à partir de leur balise
- **querySelector()** :Retourne le premier élément HTML correspondant au sélecteur CSS
- **querySelectorAll()** :Retourne une liste d'objets HTML correspondant au motif CSS du sélecteur
- **write()** :Ecrit une chaîne de caractères dans le document
- **writeln()** :Ecrit du texte dans le document

■ Sous-objets

- **body** :Objet HTML correspondant au corps de la page affichée
- **forms** :Tableau des formulaires du document
- **images** :Tableau des images de la page

<https://www.toutjavascript.com/reference/ref-window.document.php>



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Objet location

- propriétés
 - `location.href` : URL actuelle complète
 - `location.host` : nom de domaine de l'URL actuelle
 - `location.pathname` : chemin de l'URL actuelle sans le nom de domaine
- Méthodes
 - `location.reload()` : *recharger la page*

Objet history

- Propriétés
 - `history.length` : nombre d'URLs stockées dans l'historique
- méthodes
 - `history.back()` : revenir à la page précédente
 - `history.forward()` : aller à la page suivante



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Accès aux éléments HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le DOM HTML</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1 id="gros_titre">Le DOM</h1>
    <p class="para">Du texte et <a href="« page.html">un lien</a></p>
    <p class="para" paragraphe1</p>
    <script>
      var titre = document.getElementById('gros_titre');
      var texteTitre = titre.innerHTML;
      var textePara = document.querySelector('.para').textContent;
      alert('Titre : ' + texteTitre + '\nParagraphe : ' + textePara);
    </script>
  </body>
</html>
```




LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Modifier le contenu d'un élément HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le DOM HTML</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1 id="titreH1">Le DOM</h1>
    <p class="para">Text1<a href="page1.html">Text2</a></p>
    <p class="para">Un deuxième paragraphe</p>
    <script>
      document.getElementById('titreH1 ').innerHTML = 'Titre modifié !';
    </script>
  </body>
</html>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Modifier la valeur d'un attribut HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le DOM HTML</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1 id="titreH1">Le DOM</h1>
    <p class="para">Text1<a href="page1.html">Text2</a></p>
    <p class="para">Un deuxième paragraphe</p>
    <script>
      document.querySelector('a').href = 'page2.html';
    </script>
  </body>
</html>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Modifier la valeur d'un attribut HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le DOM HTML</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1 id="titreH1">Le DOM</h1>
    <p class="para">Text1<a href="page1.html">Text2</a></p>
    <p class="para">Un deuxième paragraphe</p>
    <script>
      document.querySelector('a').href = 'page2.html';
      document.querySelector('.para').className = 'para1';
    </script>
  </body>
</html>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Correspondances entre les propriétés JavaScript et CSS

JavaScript	CSS	JavaScript	CSS
marginBottom	margin-bottom	textDecoration	text-decoration
marginLeft	margin-left	textDecorationLineThrough	text-decoration: line-through
marginRight	margin-right	textDecorationNone	text-decoration: none
marginTop	margin-top	textDecorationUnderline	text-decoration: underline
lineHeight	line-height	textDecorationOverline	text-decoration: overline
letterSpacing	letter-spacing	textIndent	text-indent
verticalAlign	vertical-align	textTransform	text-transform
zIndex	z-index	styleFloat	float
backgroundAttachment	background-attachment	listStyle	list-style
backgroundColor	background-color	listStyleImage	list-style-image
backgroundImage	background-image	listStylePosition	list-style-position
backgroundPosition	background-position	listStyleType	list-style-type
backgroundRepeat	background-repeat		



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Correspondances entre les propriétés JavaScript et CSS

JavaScript	CSS
marginBottom	margin-bottom
marginLeft	margin-left
marginRight	margin-right
marginTop	margin-top
lineHeight	line-height
letterSpacing	letter-spacing
verticalAlign	vertical-align
zIndex	z-index
backgroundAttachment	background-attachment
backgroundColor	background-color
backgroundImage	background-image
backgroundPosition	background-position
backgroundRepeat	background-repeat

JavaScript	CSS
borderBottom	border-bottom
borderBottomColor	border-bottom-color
borderBottomStyle	border-bottom-style
borderBottomWidth	border-bottom-width
borderColor	border-color
borderLeft	border-left
borderLeftColor	border-left-color
borderLeftStyle	border-left-style
borderLeftWidth	border-left-width
borderRight	border-right
borderRightColor	border-right-color
borderRightStyle	border-right-style
borderRightWidth	border-right-width
borderStyle	border-style
borderTop	border-top
borderTopColor	border-top-color
borderTopStyle	border-top-style
borderTopWidth	border-top-width
borderWidth	border-width



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Correspondances entre les propriétés JavaScript et CSS

JavaScript	CSS
paddingBottom	padding-bottom
paddingLeft	padding-left
paddingRight	padding-right
paddingTop	padding-top
pageBreakAfter	page-break-after
pageBreakBefore	page-break-before
fontFamily	font-family
fontSize	font-size
fontWeight	font-weight
fontVariant	font-variant



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Validation: expression régulière

Une expression régulière est une séquence de caractères formant un motif de recherche. Le modèle de recherche peut être utilisé pour la recherche de texte et les opérations de remplacement de texte.

/pattern/modifiers;

Modifieur	Description
i	Effectuer une correspondance insensible à la casse
g	Effectuer une correspondance globale (rechercher toutes les correspondances plutôt que de s'arrêter après la première correspondance)
m	Effectuer une correspondance multiligne

```
var str = "Dev web cote Serveur";  
var res = str.replace(/serveur/i, "client");
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Validation: expression régulière

Exemple1

```
var message = 'Bonjour les TDI!';  
var regex = /les/;  
if(message.match(regex)) alert(' il y a plusieurs personnes ?');  
else alert('Tout seul...');
```

Exemple2

```
var message = 'Bonjour les gens !';  
var regex = new RegExp('les');  
if(message.match(regex)) alert('Tiens, il y a plusieurs personnes ?');  
else  
alert('Tout seul...');
```

Exemple3

```
var code = prompt('Le code svp ?', '');  
var error = false; if(!code.match(/^[0-9]{4}$/)) alert('Le code semble correct...');  
else alert('Mauvais format de code ! (4 chiffres)');
```




LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Validation: champ vide

Pour vérifier qu'un champ n'est pas vide, nous allons utiliser la propriété `valueMissing`, qui renvoie `true` si un champ possédant un attribut `required` est vide.

```
<form method="post" action="traitement.php">  
  <label for='prenom'>Entrez votre prénom svp : </label>  
  <input type='text' name='prenom' id='prenom' maxlength='20' required>  
  <!-- Afficher un message décrivant l'erreur à côté du champ-->  
  <span id='missPrenom'></span><br />
```

HTML

```
<label for='mail'>Entrez votre mail : </label>  
<input type='email' name='mail' id='mail' required><br />
```

```
<label for='tel'>Numéro de téléphone :</label>  
<input type='tel' name='tel' id='tel' required><br />
```

```
<!--Ajouter un id à votre input type="submit" afin de pouvoir y accéder plus facilement-->  
<input type='submit' value='Valider' id='bouton_envoi'>
```

```
</form>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Validation: champ vide

Pour vérifier qu'un champ n'est pas vide, nous allons utiliser la propriété `valueMissing`, qui renvoie `true` si un champ possédant un attribut `required` est vide.

```
<script>
  var formValid = document.getElementById('bouton_envoi');
  var prenom = document.getElementById('prenom');
  var missPrenom = document.getElementById('missPrenom');
  // 1 - Créer un gestionnaire d'évènement click contenant notre fonction de validation
  formValid.addEventListener('click', validation);
  // 2 - Vérifier la présence de données en utilisant la propriété validity de l'objet Element
  function validation(event){
    //Si le champ est vide alors valueMissing renvoie true et on exécute le code dans le if
    if (prenom.validity.valueMissing){
      event.preventDefault(); // Empêcher l'envoi du formulaire
      missPrenom.textContent = 'Prénom manquant'; // Afficher un message d'erreur
      missPrenom.style.color = 'red';
    }
  }
</script>
```

Js



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Validation : enable/disable

```
<script>
  function disable() {
    document.getElementById("nom").disabled=true;
  }
  function enable() {
    document.getElementById("nom").disabled=false;
  }
</script>
```

```
<form>
  <input type="texte" id="nom"/>
  <input type="button" onclick="disable()" value=" test">
</form>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Validation : onkeydown

```
<input type="text" size="50" onkeydown="keyCode(event)">
```

```
<script>
```

```
function keyCode(event) {  
    var x = event.keyCode;  
    if (x == 32) {  
        alert ("You pressed the Escape key!");  
        event.preventDefault();  
    }  
}
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Validation : checkbox

```
function checkForm(form)
{
    if(form.checkboxfield.checked) {
        alert("The checkbox IS checked");
    } else {
        alert("The checkbox IS NOT checked");
    }
    return false;
}
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Validation : Radio

```
function checkRadio(field)
    { for(var i=0; i < field.length; i++) { if(field[i].checked) return
    field[i].value; }
    return false; }
```

```
function checkForm(form)
{
    if(radiusValue = checkRadio(form.radiofield))
    { alert("You selected " + radiusValue); return true; }
    else
    { alert("Error: No value was selected!");
    return false; }
}
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Validation : Radio

```
<script>
    function testItem(form) {
        alert("Item "+form.mylist.selectedIndex+" is selected");
    }
</script>

<form action="" method="GET">
    <select size="3">
        <option>TDI201
        <option>TDI202
        <option>TDI203
    </select>
    <input value="Check Item" onClick="testItem(this.Form)">
</form>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Validation : Combiner plusieurs conditions

- On peut définir des conditions qui combinent plusieurs éléments.
- Exemple, une entrée de texte qui doit uniquement avoir une valeur si une case à cocher est cochée:

```
script type="text/javascript">
    function checkForm(form)
    {
        ...
        if(form.checkboxname.checked && (form.textinputname.value == ""))
        {
            alert("Error: error message");
            form.textinputname.focus();
            return false;
        }
        ...
        return true;
    }
</script>
```




LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Manipulation des contenus de formulaires

Validation : Combiner plusieurs conditions

- On peut définir des conditions qui combinent plusieurs éléments.
- Exemple, une entrée de texte qui doit uniquement avoir une valeur si une case à cocher est cochée:

```
function checkRadio(field)
{
    for(var i=0; i < field.length; i++)
    {if(field[i].checked) return
field[i].value;}
    return false;
}
```

```
function checkForm(form)
{
    ...
    var radioValue = checkRadio(radiofield);
    switch(radioValue)
    {
        case "Red":
            <conditions to apply if 'Red' is selected>
            break;
            ....
        default:
            <conditions to apply in all other cases>
    }
    ...
    return true;
}
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

onblur - Lorsqu'un utilisateur quitte un champ de saisie

```
<script>
  function METRE_EN_MJ() {
    var x = document.getElementById("Nom");
    x.value = x.value.toUpperCase();
  }
</script>

<body>
  Nom: <input type="text" id="Nom" onblur="METRE_EN_MJ()">
</body>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

- onchange - Lorsqu'un utilisateur modifie le contenu d'un champ de saisie
- Lorsqu'un utilisateur sélectionne une valeur de liste déroulante

HTML

Nom: `<input type="text" id="Nom"`

`onchange="METRE_A_MJ()"`

Choisir une valeur:

`<select id="listValeur"`

`onchange="Affich_Val_Select()"`

`<option value="A">A</option>`

`<option value="B">B</option>`

`</select>`

`</body>`

JS

```
<script>
```

```
function Affich_Val_Select() {
```

```
    valeur = document.forms[0].listValeur.value;
```

```
    alert("You prefer browsing internet with " + valeur);
```

```
}
```

```
function METRE_A_MJ() {
```

```
    var x = document.getElementById("Nom");
```

```
    x.value = x.value.toUpperCase();
```

```
    alert("valeur modifiée");
```

```
}
```

```
</script>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

onfocus - Lorsqu'un champ de saisie devient actif

```
<script>
function changer_couleur(x) {
  x.style.background = "yellow";
}
</script>
</head>
<body>
```

Nom: `<input type="text" onfocus="changer_couleur(this)">`



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

onselect - Lorsque le texte saisi est sélectionné

```
<script>
function Selection() {
    document.getElementById("demo").innerHTML = "Un text est selectionné";
}
</script>
</head>
<body>
```

Some text: <input type="text" value="mot1 mot 2" onselect="Selection()">

```
<p id="demo"></p>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

onsubmit - Lorsqu'un utilisateur clique sur le bouton d'envoi

```
<script>
function confirmInput() {
    nom_prenom = document.forms[0].name.value;
    alert("Salut " + nom_prenom + "! Redirection vers page2.html");
}
</script>
```

```
<form onsubmit="confirmInput()" action="page2.html">
    Nom&Prénom: <input id="name" type="text" size="20">
    <input type="submit">
</form>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

onreset - Lorsqu'un utilisateur clique sur le bouton de réinitialisation

```
<script>
    function message() {
        alert( document.forms[0].nom.value);
    }
</script>

<form onreset="message()">
    Enter your name: <input type="text" id="nom" size="20">
    <input type="reset">
</form>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

onkeydown - Lorsqu'un utilisateur appuie / maintient une touche enfoncée

```
<script>
function myFunction() {
    document.getElementById("parag").innerHTML=
document.getElementById("txtNom").value
}
</script>
</head>
<body>
```

```
<input type="text" onkeydown="myFunction()" id="txtNom"><p id="parag"></p>
```




LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

onkeypress - Lorsqu'un utilisateur appuie / maintient une touche enfoncée

```
<p> <input id="field" type="text" /></p>  
<label id="press">
```

```
<script>
```

```
    var press = document.getElementById('press');  
    document.addEventListener('keypress', function(e) {  
        press.innerHTML = e.keyCode;  
    });
```

```
</script>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

onkeyup - Lorsque l'utilisateur relâche une clé

```
Texte: <input type="text" id="mon_texte" onkeyup="myFunction(event)"/>
<p id="text_sans_espace"></p>
Texte sans espace: <input type="text" id="mon_texte2" />
<script>
```

```
function myFunction(event){
    if (event.keyCode==32 || event.which==32)
        {return false;}
        else
            document.getElementById("text_sans_espace").innerHTML =
document.getElementById("mon_texte").value;
            document.getElementById("mon_texte2").value =
document.getElementById("mon_texte").value;
};
</script>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

onkeyup - Lorsque l'utilisateur relâche une clé

Texte: `<input type="text" id="mon_texte" />`

`<p id="text_sans_espace"></p>`

Texte sans espace: `<input type="text" id="montexte2" />`

`<script>`

`document.getElementById("montexte").onkeyup=function(event){`

`if (event.keyCode==32 || event.which==32)`

`{return false;}`

`else`

`document.getElementById("montexte2").value =`

`document.getElementById("montexte").value;`

`};`

`</script>`



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

onmouseover / onmouseout - Lorsque la souris survole un élément

```
<!DOCTYPE html>
<html>
  <body>
    <h1 onmouseover="style.color='red'"
      onmouseout="style.color='black'">Mouse over this text</h1>
  </body>
</html>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

onmousedown / onmouseup - lorsque vous appuyez / relâchez un bouton de la souris

```
<!DOCTYPE html>
<html>
<head>
  <script>
    function myFunction(elmnt, clr) {   elmnt.style.color = clr;}
  </script>
</head>
<body>
  <p onmousedown="myFunction(this,'red')"
    onmouseup="myFunction(this,'green')"> TDI 201/203</p>
</body>
</html>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements d'entrée

onmousemove / onmouseout - Lorsque vous déplacez le pointeur de la souris sur une image

```
<script>
function myFunction(e) {
  x = e.clientX;
  y = e.clientY;
  coor = "Coordinates: (" + x + "," + y + ")";
  document.getElementById("demo").innerHTML = coor
}

function clearCoor() {
  document.getElementById("demo").innerHTML = "";
}
</script>
```

```
<body style="margin:0px;">

<div id="coordiv"
style="width:199px;height:99px;border:1px solid"
onmousemove="myFunction(event)"
onmouseout="clearCoor()"></div>

<p id="demo"></p>

</body>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

Événements Click/ dblclick

onclick - Quand le bouton est cliqué

ondblclick - Quand un texte est double-cliqué

```
<script>
    function AfficheBonjour() {
        document.getElementById("parag").innerHTML = "Bonjour TDI";
    }
    function AfficheBonsoir() {
        document.getElementById("parag").innerHTML = "Bonsoir TDI";
    }
</script>
</head>
<p ondblclick="AfficheBonjour()">Doubleclick ici</p>
<p onclick="AfficheBonsoir()">click ici</p>
<p id="parag"></p>
```



LES ÉLÉMENTS DE BASE DE JAVASCRIPT

Événements liés aux formulaires

- onload - Une fois la page chargée
- onload - Quand une image a été chargée
- onerror - Lorsqu'une erreur survient lors du chargement d'une image
- onunload - Lorsque le navigateur ferme le document
- onresize - Lorsque la fenêtre du navigateur est redimensionnée