

# Cours JavaScript

---

Réalisé par Ahmed EL ECHI

(Vous êtes libre de le modifier, l'utiliser et/ou mettre votre nom dessus)

S'il y a une erreur, veuillez me contacter sur [ahmedelechi@yahoo.fr](mailto:ahmedelechi@yahoo.fr)

## Table des matières

1	Extrait du programme officiel année 2021-2022 (4SI-STI) .....	2
2	Déclaration des variables en JavaScript .....	2
3	Affectation, affichage et saisie.....	2
4	Opérateurs de comparaison .....	2
5	Opérateurs logiques.....	3
6	Opérateurs arithmétiques .....	3
7	Fonctions sur les nombres.....	3
8	Fonctions sur les chaînes .....	5
9	Fonctions sur les tableaux .....	7
10	Fonctions diverses.....	8
11	Porté des variables.....	8
12	Quelques erreurs dans le livre scolaire (4eme SI, ancien régime, TIC) .....	11

# 1 Extrait du programme officiel année 2021-2022 (4SI-STI)

- Les méthodes JAVASCRIPT prédéfinies à utiliser

	Objet	Méthodes prédéfinies
Number	Nombre	<ul style="list-style-type: none"> <li>• Number, isNaN</li> <li>• toString ; valueOf ; parseInt ; parseFloat ; eval</li> </ul>
String,	Chaîne de caractères	<ul style="list-style-type: none"> <li>• length ; indexOf ; lastIndexOf ; substr ; substring ; replace ; toUpperCase ; toLowerCase ; concat ; trim ; charAt ; split</li> </ul>
Array	Tableau	<ul style="list-style-type: none"> <li>• length, toString</li> </ul>

## 2 Déclaration des variables en JavaScript

fonctionnalité	Signe/Fonction en javascript
Déclaration implicite (ضمني) (obligatoirement avec affectation)	<b>x = 10</b> ← Déclaration <b>x = x+1</b>
Déclaration explicite (صریح), avec affectation	<b>var x = 10</b> ← Déclaration <b>x = x+1</b>
Déclaration explicite, sans affectation	<b>var x</b> ← Déclaration <b>x=10</b> <b>x = x+1</b>

## 3 Affectation, affichage et saisie

fonctionnalité	Signe/Fonction en javascript	Exemples
Affectation	=	email = "ahmed@gmail.com"
Affichage dans la console du navigateur	<b>console.log( ... )</b>	<b>console.log</b> ("Bonjour")
Affichage dans une alerte	<b>alert(...)</b>	<b>alert</b> ("Bonjour")
saisie d'une valeur donnée par l'utilisateur	<b>variable = prompt(...)</b>	n = <b>prompt</b> ('Donner un nombre')

## 4 Opérateurs de comparaison

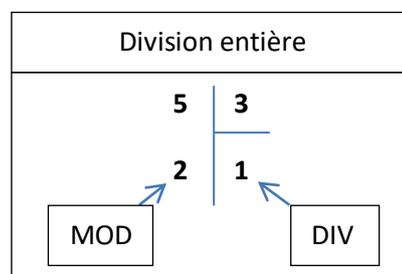
fonctionnalité	Signe en javascript	Exemples
Egalité	<b>==</b>	if(x == 5) ...
Différent	<b>!=</b>	if(x != 5) ...
Supérieur ou égale	<b>&gt;=</b>	if(x >= 5) ...
...	...	...

## 5 Opérateurs logiques

fonctionnalité	Signe en javascript	Exemples
Et	<b>&amp;&amp;</b>	if(x == true <b>&amp;&amp;</b> y == false ) ...
Ou	<b>  </b>	if(x == true <b>  </b> y == false) ...
Non	<b>!</b>	if( <b>!</b> x ) ...

## 6 Opérateurs arithmétiques

fonctionnalité	Signe en javascript	Exemples	Résultat
division réelle	/	x = 5 / 3	x vaut 1.66
Reste de la division entière (MOD)	%	x = 5 % 3	x vaut 2
Résultat (Quotient) de la division entière (DIV)	<u>1<sup>ère</sup> méthode :</u> Math.trunc( .../... )  <u>2<sup>ème</sup> méthode :</u> ~~(... / ...)	<u>1<sup>ère</sup> méthode :</u> x = <b>Math.trunc</b> (5 / 3)  <u>2<sup>ème</sup> méthode :</u> x = <b>~~</b> (5 / 3)	x vaut 1



## 7 Fonctions sur les nombres

fonctionnalité	Fonction en JavaScript	Exemples	Résultat
Verifier si une valeur n'est pas un nombre (is Not a Number)	<b>isNaN( ... )</b>	a = isNaN("Bonjour")  b = isNaN( "3" ) c = isNaN( " 3 " ) d = isNaN( " 3 6 " ) e = isNaN( 3 ) f = isNaN(-1.23)	a vaut <b>true</b>  b vaut false c vaut false d vaut <b>true</b> e vaut false f vaut false

Convertir une chaîne en un nombre	<b>Number(ch)</b>	n1 = Number("14.50") n2 = Number(" 14.50 ") n3 = Number(" 14.50 5 ") n4 = Number("14.50 abc")	n1 vaut 14.5 n2 vaut 14.5 n3 vaut <b>NaN</b> n4 vaut <b>NaN</b>
Convertir un nombre en une chaîne	<b>n.toString( )</b>	x = 5 ch1 = x.toString( )  y = 18.75 ch2 = y.toString( )	ch1 vaut "5"  ch2 vaut "18.75"
Convertir une chaîne en un entier	<b>parseInt(ch)</b> <del>Math.parseInt(ch)</del>	a = parseInt("10") b = parseInt(" 10 ") c = parseInt("10.33") d = parseInt("34 45 66") e = parseInt("30 dinars") f = parseInt("30.5 dinars") g = parseInt("j'ai 30 dinars")	a vaut 10 b vaut 10 c vaut 10 d vaut 34 e vaut <b>30</b> f vaut <b>30</b> g vaut <b>NaN</b>
Convertir une chaîne en un réel	<b>parseFloat(ch)</b> <del>Math.parseFloat(ch)</del>	a = parseFloat("10") b = parseFloat(" 10 ") c = parseFloat("10.33") d = parseFloat("34 45 66") e = parseFloat("30 dinars") f = parseFloat("30.5 dinars") g = parseFloat("j'ai 30 dinars")	a vaut 10 b vaut 10 c vaut <b>10.33</b> d vaut 34 e vaut <b>30</b> f vaut <b>30.5</b> g vaut <b>NaN</b>

## 8 Fonctions sur les chaînes

<i>fonctionnalité</i>	<i>Fonction en JavaScript</i>	<i>Exemples</i>	<i>Résultat</i>
Longueur d'une chaîne ch	<code>ch.length</code>	<pre>ch="bjr" n = ch.length</pre>	n vaut 3
Le caractère d'indice i de la chaîne ch. Renvoie une chaîne vide dans le cas où l'indice déborde	<code>ch.charAt(i)</code>	<pre>ch="bjr" c1 = ch.charAt(1)  ch="bjr" c2 = ch.charAt(10)</pre>	<p>c1 vaut 'j'</p> <p>c2 vaut ''</p>
Le caractère d'indice i de la chaîne ch. Renvoie une <u>undefined</u> dans le cas où l'indice déborde	<code>ch[i]</code> <b>(hors programme bac 2021-2022)</b>	<pre>ch="bjr" c1 = ch[1]  ch="bjr" c2 = ch[10]</pre>	<p>c1 vaut 'j'</p> <p>c2 vaut <b>undefined</b></p>
Retourne la <b>première</b> position d'une sous-chaîne <b>Sch</b> dans la chaîne <b>Ch</b> , en commençant la recherche à partir de la position <b>debut</b> . Elle renvoie -1 dans le cas contraire ou il n'existe pas.  Si l'argument <b>debut</b> est omis, la recherche débutera à la position 0	<code>P=Ch.indexOf(Sch, debut)</code>	<pre>Ch="Javascript" Sch="a" P1=Ch.indexOf(Sch)  Ch="position" Sch="o" P2=Ch.indexOf(Sch,3)  Ch="Javascript" Sch="@ " P3=Ch.indexOf(Sch)  Ch="Javascript" Sch="script" P4=Ch.indexOf(Sch)</pre>	<p>P1 vaut 1</p> <p>P2 vaut 6</p> <p>P3 vaut -1</p> <p>P4 vaut 4</p>
Retourne la <b>dernière</b> position d'une sous-chaîne <b>Sch</b> dans la chaîne <b>Ch</b> , en commençant la recherche à <b>partir 0 jusqu'à la position fin</b> . Elle renvoie -1 dans le cas contraire ou il n'existe pas.  Si l'argument	<code>P=Ch.lastIndexOf(Sch, fin)</code>	<pre>Ch="Javascript" Sch="a" P3=Ch.lastIndexOf(Sch)  Ch="Javascript" Sch="i" P4=Ch.lastIndexOf(Sch, 5)  Ch="Javascript" Sch="a" P1=Ch.lastIndexOf(Sch,2)</pre>	<p>P3 vaut 3</p> <p>P4 vaut -1</p> <p>P1 vaut 1</p>

<p><b>debut</b> est omis, la recherche débutera à la position 0</p>		<pre>Ch="Javascript" Sch="a" ; P2=Ch.lastIndexOf(Sch,Ch.length)</pre>	P2 vaut 3
<p>permet d'extraire d'une chaîne donnée Ch, une sous chaîne de caractères Sch à partir d'une position déterminée <b>debut</b> et d'une longueur déterminée <b>Nbr</b>. Le paramètre <b>Nbr</b> peut être omis de façon à inclure tous les caractères jusqu'à la fin de la chaîne.</p>	<pre>Sch=Ch.substr(debut,Nbr)</pre>	<pre>Ch="Javascript" ; Sch1 = Ch.substr(3,4);  Ch="Javascript" ; Sch2 = Ch.substr(3);</pre>	<p>Sch1 vaut "ascr"</p> <p>Sch2 vaut "ascript"</p>
<p>permet d'extraire d'une chaîne donnée Ch, une sous chaîne de caractères Sch à partir d'une position <b>debut</b> jusqu'à la position <b>fin</b>. Le paramètre <b>fin</b> peut être omis de façon à inclure tous les caractères jusqu'à la fin de la chaîne.</p>	<pre>Sch=Ch.substring(debut,fin_exclu )</pre>	<pre>Ch="Javascript" ; Sch3 = Ch.substring (0,2);  Ch="Javascript" ; Sch1 = Ch.substring (3,4);  Ch="Javascript" ; Sch4 = Ch.substring (3,3);  Ch="Javascript" ; Sch2 = Ch.substring (3);</pre>	<p>Sch3 vaut 'Ja'</p> <p>Sch1 vaut 'a'</p> <p>Sch4 vaut ''</p> <p>Sch2 vaut 'ascript'</p>
<p>Concaténer ch1 et ch2</p>	<pre>Ch=Ch1.concat(Ch2)</pre>	<pre>Ch1="bon" Ch2="jour" Ch3=Ch1.concat(Ch2); Ch4=Ch2.concat(Ch1);</pre>	<p>Ch3 vaut "bonjour"</p> <p>Ch4 vaut "jourbon"</p>
<p>Concaténer ch1 et ch2</p>	<pre>Ch=Ch1 + Ch2</pre>	<pre>Ch1="bon" Ch2="jour" Ch3=Ch1 + Ch2 Ch4= Ch2 + Ch1</pre>	<p>Ch3 vaut "bonjour"</p> <p>Ch4 vaut "jourbon"</p>
<p>transforme une chaîne de caractères Ch en majuscule</p>	<pre>Ch.toUpperCase( )</pre>	<pre>Ch="JAVAscript" Ch1=Ch.toUpperCase( )</pre>	<p>Ch1 vaut "JAVASCRIPT"</p>
<p>transforme une chaîne de</p>	<pre>Ch.toLowerCase( )</pre>	<pre>Ch="JAVAscript" Ch1=Ch.toLowerCase( )</pre>	<p>Ch1 vaut "javascript"</p>

caractères Ch en minuscule			
Enlève les espaces du début et de la fin de la chaîne ch	Ch.trim( )	Ch=" bjr "	Ch1 vaut "bjr"
Remplace, dans ch, la <b>premiere</b> occurrence de ch1 par ch2	ch.replace(ch1,ch2)	ch = "Mr <b>Blue</b> has a <b>blue</b> house and a <b>blue</b> car";  result = ch.replace("blue", "red");	result vaut " Mr <b>Blue</b> has a <b>red</b> house and a <b>blue</b> car"
Découpe une chaîne ch en se basant sur la chaîne ch1 et range le résultat dans un tableau	Ch.split(ch1)	ch = "Mr Blue has a <b>blue</b> house and a <b>blue</b> car";  t = ch.split('blue')	t vaut Array( "Mr Blue has a ", " house and a ", " car" )

## 9 Fonctions sur les tableaux

<i>fonctionnalité</i>	<i>Fonction en JavaScript</i>	<i>Exemples</i>	<i>Résultat</i>
Nombre de cases du tableau t	<b>t.length</b>	t = Array("Banana", "Orange", "Apple");  n = t.length	n vaut <b>3</b>
Convertir un tableau en une chaîne contenant la concaténation de tous les éléments du tableau séparé par virgule	<b>t.toString( )</b>	t = Array("Banana", "Orange", "Apple");  ch = t.toString( )	Ch vaut " <b>Banana,Orange,Apple</b> "

## 10 Fonctions diverses

fonctionnalité	Fonction en JavaScript	Exemples	Résultat
évalue une expression écrite sous forme de chaîne de caractères	<b>eval(ch)</b>	<pre>a = 5.2 msg = "a+3" x=eval(msg)  b = '5.2' msg = " parseInt(b) +3 " y=eval(msg)  msg = "alert('Bonjour')"</pre>	<p>x vaut <b>8.2</b></p> <p>y vaut <b>8</b></p> <p>Le navigateur va afficher une alerte contenant le message "Bonjour "</p>

## 11 Porté des variables

La règle (voir page 85 du livre scolaire 4SI-TIC ancien régime)

**Règle 1 :** Les variables déclarées tout au début d'un script, en dehors et avant toute fonction, seront toujours globales, qu'elles soient déclarées de façon **implicite (sans le mot clé var)** ou **explicite (par le mot clé var)**. On pourra donc les exploiter partout dans le script.

Javascript	résultat
<pre>var x = 10 // déclarer une variable x avec le mot var y = 20    // déclarer une variable y sans le mot var  // déclaration d'une fonction test function test( ) {      // Dans la fonction test, afficher     // les variables déclarées dans le programme principal     console.log(x)     console.log(y) }  // appel de la fonction test test()</pre>	<p>afficher la valeur 10 afficher la valeur 20</p>

**Règle 2 :** Dans une fonction, une variable déclarée **par le mot clé var** aura une **portée limitée** à cette seule fonction

<i>JavaScript</i>	<i>résultat</i>
<pre>// déclaration d'une fonction test function test( ) {   var x = 2 // déclarer une variable avec le mot var }  // appel de la fonction test test()  // Dans le programme principal, afficher // la variable déclarée dans la fonction test console.log(x)</pre>	afficher une erreur ( <b>x is not defined</b> )

**Règle 3 :** Dans une fonction, une variable déclarée **sans le mot clé var** aura une **portée globale**

<i>JavaScript</i>	<i>résultat</i>
<pre>// déclaration d'une fonction test function test( ) {   x = 2 // déclarer une variable sans le mot var }  // appel de la fonction test test()  // Dans le programme principal, afficher // la variable déclarée dans la fonction test console.log(x)</pre>	afficher la valeur 2

**Règle 4 :** Déclarer une variable globale et une variable local avec le même nom (la règle est définie dans la colonne constatation, dans le tableau ci-dessous)

Javascript	résultat	Constatation
<pre>// déclarer x de façon explicite var x = 5  // déclaration d'une fonction test function test( ) {     // re-déclarer x de façon explicite     var x = 2 }  // appel de la fonction test test ( )  // Dans le programme principal, afficher // la variable x console.log(x)</pre>	<p>afficher la valeur 5</p>	<p>x à l'extérieur de la fonction et x à l'intérieur de la fonction <u>sont deux variables distinctes</u>.</p>
<pre>// déclarer x de façon explicite var x = 5  // déclaration d'une fonction test function test( ) {     // re-déclarer x de façon implicite     x = 2 }  // appel de la fonction test test ( )  // Dans le programme principal, afficher // la variable x console.log(x)</pre>	<p>afficher la valeur 2</p>	<p>x à l'extérieur de la fonction et x à l'intérieur de la fonction sont les mêmes. Donc la modification effectuée à l'intérieur de la fonction s'applique directement sur la variable globale.</p>

## 12 Quelques erreurs dans le livre scolaire (4eme SI, ancien régime, TIC)

### ▶ La méthode `lastIndexOf()`

La syntaxe :

**P=Ch.lastIndexOf(Sch,pos);**

Cette méthode recherche dans une chaîne de caractères **Ch** la dernière occurrence d'une sous chaîne **Sch** et retourne sa position si elle existe ou -1 dans le cas contraire.

Il est possible mais facultatif de transmettre comme deuxième paramètre la position **pos** à partir de laquelle, dans la chaîne de caractères, la recherche doit commencer.

Exemples	Résultats
<pre>Ch="Javascript" ; Sch="a" ; P=Ch.lastIndexOf(Sch,0);</pre>	<p><del>P=3</del></p>
<pre>Ch="Javascript" ; Sch="@" ; P=Ch.lastIndexOf(Sch);</pre>	<p>P=-1</p>

### ▶ La méthode `substr`

La syntaxe :

**Sch=Ch.substr(P,Nbr);**

Cette méthode permet d'extraire d'une chaîne donnée **Ch**, une sous chaîne de caractères **Sch** à partir d'une position déterminée **P** et d'une longueur déterminée **Nbr**.

Le paramètre **Nbr** peut être omis de façon à inclure tous les caractères jusqu'à la fin de la chaîne.



Exemples	Résultats
<pre>Ch="Javascript" ; Sch1 = Ch.substr(3,3); Sch2 = Ch.substr(1);</pre>	<p><del>Sch1 ="vas"</del> Sch2 = "avascript"</p>