



Université Assane Seck de Ziguinchor

JavaScript

UFR ST – Département Informatique
Licence 2 Ingénierie Informatique 2021/2022

Marie NDIAYE

C'est quoi JavaScript ?

- ▶ Un langage de programmation qui peut être inclus dans des pages HTML.
- ▶ destinées aux navigateurs WWW les plus courants.
- ▶ Offre la possibilité de créer des pages interactives.
- ▶ Contrairement à l'utilisation du CGI, les réponses aux événements déclenchés par le lecteur de la page se font au niveau du navigateur, sans nécessité d'une transmission vers le serveur d'où provient la page.

JavaScript ne doit pas être confondu avec Java, qui est un langage beaucoup plus complexe permettant de créer des applications autonomes.

Les éléments du langage

Instructions et commentaires

► Instructions

- Les instructions JavaScript sont composées de : valeurs, opérateurs, expressions, mots clés et commentaires.
- La plupart des programmes JavaScript contiennent de nombreuses instructions JavaScript.
- Les instructions sont exécutées, une par une, dans le même ordre qu'elles sont écrites.
- Les points-virgules (;) séparent les instructions JavaScript.

► Commentaires

► Sur une ligne

► Exemple :

```
// Commentaire sur une ligne.
```

► Sur plusieurs lignes

► Exemple

```
/* Commentaire sur  
   plusieurs ligne */
```

Identificateurs

► Identificateurs

- Toutes les variables JavaScript doivent être identifiées par des noms uniques (des identificateurs).
- Les identificateurs peuvent être des noms courts (comme `x` et `y`) ou des noms plus descriptifs (`age`, `sum`, `totalVolume`).
- Les règles générales de construction des noms de variables (identificateurs uniques) sont :
 - Les noms peuvent contenir des lettres, des chiffres, des traits de soulignement et des signes dollar.
 - Les noms doivent commencer par une lettre ou par \$
 - Les noms sont sensibles à la casse (`y` et `Y` sont des variables différentes). Les mots réservés (comme les mots clés JavaScript) ne peuvent pas être utilisés comme noms.
- JavaScript est sensible à la casse.
- Les mots réservés de JavaScript :
https://www.w3schools.com/js/js_reserved.asp

Types de variables

- ▶ En JavaScript, il n'y a pas besoin de déclarer le type de variables que l'on utilise, contrairement à des langages évolués tels que le langage C ou Java pour lesquels il faut préciser s'il s'agit d'entier (`int`), de nombre à virgule flottante (`float`), de caractères (`char`), etc.
- ▶ En fait, JavaScript n'autorise la manipulation que de 4 types de données :
 - ▶ des nombres : entiers ou à virgules
 - ▶ des chaînes de caractères (`string`) : une suite de caractères
 - ▶ des booléens : des variables à deux états permettant de vérifier une condition :
 - ▶ *false* : lors d'un résultat faux
 - ▶ *true* : si le résultat est vrai
 - ▶ des variables de type *null* : un mot caractéristique pour indiquer qu'il n'y a pas de données.

Déclaration de variable et de constante

- ▶ La déclaration de variable se fait en utilisant le mot clé **var** ou **let** :
 - ▶ Exemple
 - ▶ `var uneVariable;`
 - ▶ `let uneVariable;`
- ▶ Après la déclaration, la variable n'a pas de valeur; techniquement elle est indéfinie.
- ▶ Une constante se déclare avec le mot clé **const**
 - ▶ Exemple : `const pi = 3.14;`
- ▶ Si une variable est déclarée dans un bloc avec **var**, cette variable reste visible en dehors du bloc.
- ▶ L'utilisation de **let** ou de **const** permet de créer des variables et des constantes qui ont une portée locale.
- ▶ Les variables définies avec **let** ne peuvent pas être redéclarées.
- ▶ Les variables définies avec **let** doivent être déclarées avant utilisation.

Opérateurs

► Les opérateurs de base

► Les opérateurs de comparaison

égal à	==
différent de	!=
inférieur ou égal à	<=
supérieur à	>
supérieur ou égal à	>=
inférieur ou égal à	<

+	Addition
-	Soustraction
*	Multiplication
**	exponentiation
/	Division
%	Modulo
++	Incrémentation
--	Décrémentation

► Les opérateurs logiques

ET logique	&&
OU logique	
NON logique	!

Assignement

- ▶ En utilisant l'assignement on attribue une valeur à une variable.
- ▶ La structure générale de l'assignement est :
target = source.

- ▶ Exemples :

- ▶ $a = 3 ;$
 - ▶ $b = \text{"hello world"} ;$
 - ▶ $c = \text{'hello from Mars'} ;$
 - ▶ $a = b ;$
 - ▶ $a = 3 * 4 * 3 + 2 ;$
- ▶ Il existe des notations abrégées pour certains types d'assignements :

Notation abrégée	Signification
$x += y$	$x = x + y$
$x -= y$	$x = x - y$
$x *= y$	$x = x * y$
$x /= y$	$x = x / y$
$x \%= y$	$x = x \% y$
$x **= y$	$x = x ** y$

Opérateurs et assignement pour les chaines de caractères

- ▶ L'opérateur **+** peut également être utilisé pour concaténer des chaînes de caractères.
- ▶ L'opérateur d'affectation **+=** peut également être utilisé pour concaténer des chaînes d caractères.
- ▶ Pour assigner une chaîne de caractères, on la place entre guillemets doubles ("...") ou simples ('...').
- ▶ **Exemple**
 - ▶ `let y = "5" + 5;`
 - ▶ `let z = "Hello" + 5;`

Branchements conditionnels

- ▶ En JavaScript, nous avons les instructions conditionnelles suivantes :
 - ▶ **if** : spécifie un bloc de code à exécuter, si une condition spécifiée est vraie.
 - ▶ **else** : spécifie un bloc de code à exécuter, si la même condition est fausse.
 - ▶ **else if** : spécifie une nouvelle condition à tester, si la première condition est fausse.
 - ▶ **Switch** : spécifie de nombreux blocs de code alternatifs à exécuter.

- ▶ **Exemple :**

```
if (heure < 10) {  
    salutation = "Good morning";  
}  
else if (time < 20) {  
    salutation = "Good day";  
}  
else {  
    salutation = "Good evening";  
}
```

Exemple : Switch

```
► switch (new Date().getDay()) {  
    case 0:  
        jour = "Sunday";  
        break;  
    case 1:  
        jour = "Monday";  
        break;  
    case 2:  
        jour = "Tuesday";  
        break;  
    case 3:  
        jour = "Wednesday";  
        break;  
    case 4:  
        jour = "Thursday";  
        break;  
    case 5:  
        jour = "Friday";  
        break;  
    case 6:  
        jour = "Saturday";  
}
```

Les boucles : for, for in, for of

► For

```
for (let i = 0; i < cars.length; i++) {  
  text += cars[i] + "<br>";  
}
```

► For in

```
const numbers = [45, 4, 9, 16, 25];  
let txt = "";  
for (let x in numbers) {  
  txt += numbers[x];  
}
```

► For of

```
const cars = ["BMW", "Volvo", "Mini"];  
  
let text = "";  
for (let x of cars) {  
  text += x;  
}
```

Les boucles : while, do ... while

► while

```
while (i < 10) {  
    text += "The number is " + i;  
    i++;  
}
```

► do while

```
do {  
    text += "The number is " + i;  
    i++;  
}  
while (i < 10);
```

Fonctions

► Définition de fonction

- Une fonction JavaScript est définie avec le mot-clé `function`, suivi d'un nom, suivi de parenthèses `()`.
- Les noms de fonction peuvent contenir des lettres, des chiffres, des traits de soulignement et des signes dollar (mêmes règles que les variables).
- Les parenthèses peuvent inclure des noms de paramètres séparés par des virgules : (
 - `paramètre1, paramètre2, ...`)
- Le code à exécuter, par la fonction, est placé entre accolades : `{ }`

► Appel de fonction

- Le code à l'intérieur de la fonction s'exécutera lorsque "quelque chose" invoque (appelle) la fonction :
- Lorsqu'un événement se produit (lorsqu'un utilisateur clique sur un bouton)
- Lorsqu'il est invoqué (appelé) à partir du code JavaScript
Automatiquement (auto-invoqué)

Exemple

- ▶ Sans retour de valeur

- ▶ `function myFunction() { alert("Hello World!"); }`

- ▶ Avec retour de valeur

- ▶ `function toCelsius(fahrenheit) {
 return (5/9) * (fahrenheit-32);
}`

- `document.getElementById("demo").innerHTML = toCelsius (70);`

Exercice d'application 1

- ▶ Écrivez un script qui affiche le message "Hello World !" sur la page.

Exercice d'application 2

- ▶ Déclarez trois variables a, b et c comme "1", "2" et a+b.
- ▶ Afficher le type et la valeur de chaque variable.
- ▶ Ensuite, forcez le type de a et b à entier et exécuter à nouveau a+b.
- ▶ Afficher le type et la valeur de chaque variable.

Exercice d'application 3

- ▶ Créez une fonction qui fait la somme d'un nombre indéterminé d'entiers.
- ▶ Écrivez un script qui va tester cette fonction.