

Introduction au langage Javascript

Spécialité NSI 1ères

Lycée Général Joseph Saverne

Web

Introduction

Introduction

Nous avons déjà pu nous familiariser avec le couple HTML-CSS, en faite, le couple est plutôt un trio, car aujourd'hui un développeur web ne peut pas faire l'impasse sur le JavaScript .

Nous avons ensuite regardé quelques exemples d'utilisation du JavaScript, notamment dans le cas des interactions entre un utilisateur et une page web.

Dans ce cours je vous propose une introduction au langage Javascript.

Rappel & infos

Quelques rappels sur javascript :

- Introduit dans le **navigateur Netscape** in 1995.
- JavaScript a été créé en dix jours par **Brendan Eich en 1995**.
- Appelé ainsi pour profiter de la popularité du nom du langage java.
- C'est un **langage standardisé** sur la base de la norme ECMAScript.
- La référence la plus complète et à jour est le site Mozilla Developer Network [MDN](#).
- **97 %** de tous les sites Web utilisent Javascript

ECMAScript est un ensemble de normes concernant les langages de programmation de type script et standardisées par Ecma International dans le cadre de la spécification ECMA-262. Il s'agit donc d'un standard, dont les spécifications sont mises en œuvre dans différents langages de script, comme JavaScript ou ActionScript.

<https://fr.wikipedia.org/wiki/ECMAScript>

Objectifs de la séance

- Repérer, dans un nouveau langage de programmation, les traits communs et les traits particuliers à ce langage. *(Comme pour notre introduction au langage C)*
- Écrire des codes Javascript simples syntaxiquement correct sur machine.
- Lire et expliquer un code JS simple sur écran ou sur papier.

Plan

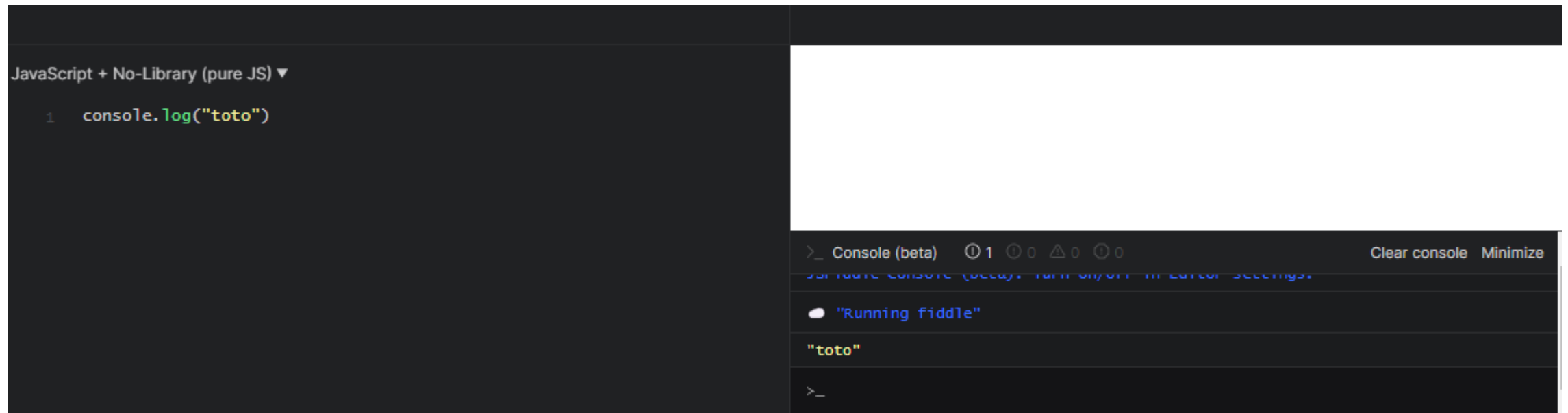
1. Variables
2. Types de données
3. Opérateurs
4. Instructions conditionnelles
5. Strings
6. Collections
7. Fonctions
8. Interactions utilisateurs
9. On code !!!!
10. Tips

Environnement de travail

Je vous rappelle l'adresse du site sur lequel nous allons travailler: <https://jsfiddle.net/>

Nous avons ici la console à droite et la fenêtre javascript à gauche dans laquelle nous allons écrire notre code.

Nous utiliserons la méthode `console.log(« Message »)` pour voir nos résultats,



The screenshot shows the jsfiddle.net editor interface. On the left, the code editor is titled "JavaScript + No-Library (pure JS)" and contains a single line of code: `console.log("toto")`. On the right, the console is open and shows the output of the code: `"Running fiddle"` and `"toto"`. The console also includes a "Clear console" button and a "Minimize" button.

Les variables



Web

Déclaration de variable

Qu'observez-vous?

```
var sheldonQI= 187;  
let leonardQI= 173;  
const PI=3.141592653589793;
```

Déclaration de variables

Pour déclarer une variable on peut utiliser le mot-clé var ou const (pour une constante).

var nom_variable ou **const nom_variable**

```
var sheldonQI= 187;  
let leonardQI= 173;  
const PI=3.141592653589793;
```

Précisions

Les noms de variables :

- Les caractères valides sont : a-z, A-Z, 0-9, _ et \$ •

Ne doit pas commencer par un chiffre (le \$ au début est déconseillé).

- Les noms sont sensibles à la casse : toto et Toto sont deux variables distinctes

Les commentaires / Espaces

```
// ceci est un commentaire.
```

```
/* Un commentaire
```

Sur plusieurs

Lignes

```
*/
```

Dans les déclarations:

- Les espaces et nouvelles lignes ne sont pas prises en compte
- Une déclaration termine par un point virgule

```
x = 3 +  
    5 * 4; console.log(  
x);
```

- *Énoncé : Déclarez une variable age et initialisez-la avec votre âge. Affichez ensuite cette variable dans la console

Type de données

Web

Précisions

Javascript est un langage dont le typage est :

- Faible : pas nécessaire de déclarer le type d'une variable avant de l'utiliser
- Dynamique : la même variable peut avoir différents types au cours de son existence

Précisions

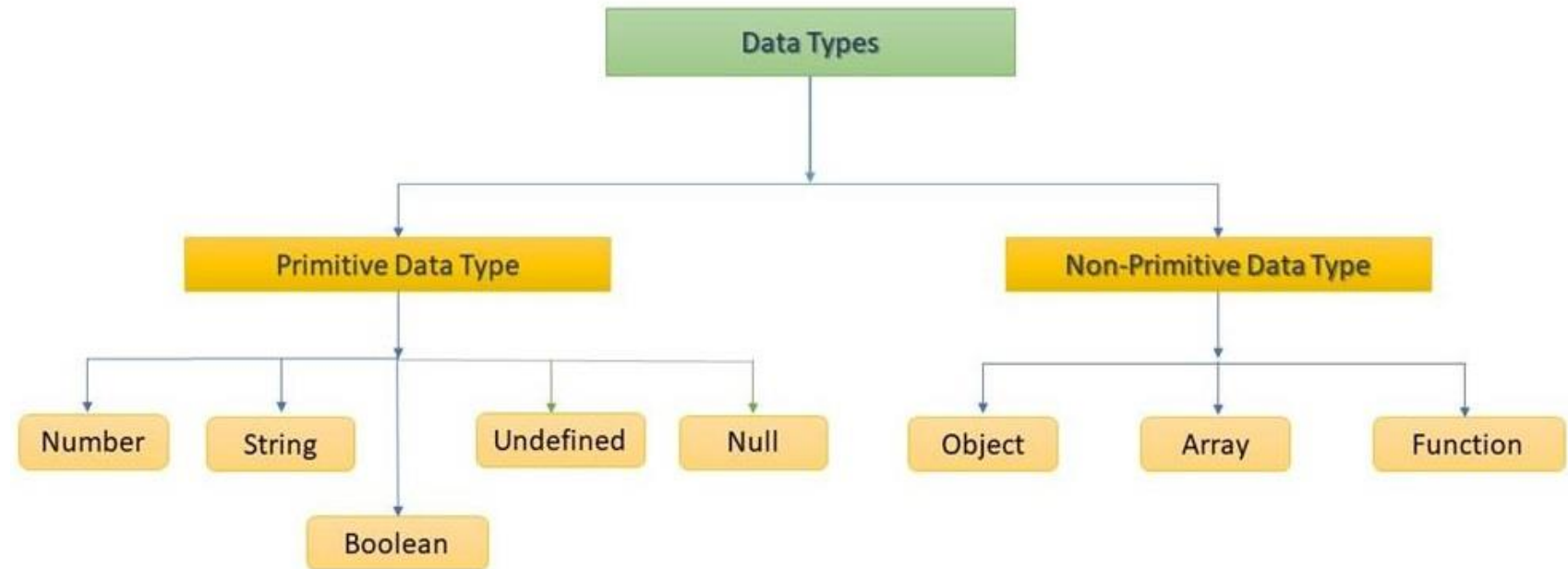
Number : Valeur numérique.

String : séquence de caractère

Boolean : True ou False.

Undefined : Valeur indéfinie.

Null : aucune valeur



Opérateurs

Web

Les opérateurs

► Opérateurs arithmétiques

`+`, `-`, `*`, `/`, `%` (modulo), `++` (incrément), `--` (décrément)

NB: `+` permet aussi de concaténer (ou fusionner) deux string

► Opérateurs d'assignation

`=` (affectation), `+=` (ajouter à), `-=`

NB: `+=` permet aussi de concaténer deux string

NB : il en existe d'autres mais ce cours est une introduction

Opérateurs

► Opérateurs logiques

&& (et),

|| (ou),

! (non)

► Opérateurs de comparaison

==,

<, <=,

>, >=,

!= (différent),

```
if (true && true)
{
    console.log(true);
}
```

NB: permettent aussi de comparer deux string en ordre lexicographique.

NB: === (égalité incluant le type)

NB: !== (inégalité incluant le type)

Les array

Javascript supporte à la fois :

- les **tableaux linéaires** avec des **crochets []**
- les **structures associatives** avec des **accolades {}**

```
sports=["golf","tennis","ping"];
```

```
balls = {"golf": "Golf balls",  
         "tennis": "Tennis balls",  
         "ping": "Ping Pong balls"};
```

NB :(les structures associatives sont des objets mais nous en reparlerons l'an prochain)

Instructions conditionnelles

Web

If else.

Qu'observez-vous

```
var sheldonQi = 187, leonardQi = 173;
if(sheldonQi > leonardQi){
    console.log('Bazinga');
}else if(sheldonQi == leonardQi){
    console.log('IMPOSSIBRU!!');
}else{
    console.log('My mother had me tested');
}
```

If else.

On utilise des accolades {} (comme vu en c)

```
var sheldonQi = 187, leonardQi = 173;
if(sheldonQi > leonardQi){
    console.log('Bazinga');
}else if(sheldonQi == leonardQi){
    console.log('IMPOSSIBRU!!');
}else{
    console.log('My mother had me tested');
}
```

À vous de jouer

Faites les exercices débutants 1 et 2

Strings

Web

Observez

```
var nom = 'Jean';  
var prenom = "Dupont";  
var message = `Bonjour, ${prenom} ${nom} !`;
```

Strings

Déclarations de chaîne :

```
var nom = 'Jean';  
var prenom = "Dupont";  
var message = `Bonjour, ${prenom} ${nom} !`;
```

La concaténation de chaînes consiste à joindre plusieurs chaînes pour former une seule chaîne. En JavaScript, cela peut être fait avec l'opérateur '+' ou avec les **backticks** (`` ALTGR + 7) pour l'interpolation de chaînes.

```
var nom = 'Jean';  
var prenom = 'Dupont';  
var nomComplet = prenom + ' ' + nom; // Utilisation de l'opérateur '+'  
console.log(nomComplet); // Affiche "Jean Dupont"  
  
var message = `Bonjour, ${prenom} ${nom} !`; // Utilisation des backticks pour l'interpolation  
console.log(message); // Affiche "Bonjour, Jean Dupont !"
```

Quelques précisions

La longueur d'une chaîne peut être obtenue en utilisant la propriété **length** :

```
var texte = 'Bonjour';  
console.log(texte.length); // Affiche 7
```

Les caractères individuels d'une chaîne peuvent être accédés en utilisant la notation des crochets

'[]' avec l'indice du caractère.

```
var mot = 'JavaScript';  
console.log(mot[0]); // Affiche 'J'  
console.log(mot[2]); // Affiche 'v'
```

Quelques précisions

JavaScript fournit plusieurs méthodes pour manipuler les chaînes de caractères, telles que **toUpperCase()**, **toLowerCase()**, **charAt()**, **indexOf()**, **substring()**, **split()**,

```
var texte = 'Bonjour';  
console.log(texte.toUpperCase()); // Affiche "BONJOUR"  
console.log(texte.indexOf('o')); // Affiche 1  
console.log(texte.substring(0, 3)); // Affiche "Bon"
```

Les collections

Web

Les boucles en bref

On utilise des accolades `{}` (comme vu en c)

`while` et `for` s'écrivent comme en php

Dans les boucles les instructions `break` et `continue` s'écrivent comme en php.

Un exemple la boucle for

Un parcours de structure. Ça rappelle le python

```
balls = {"golf": "Golf balls",
         "tennis": "Tennis balls",
         "ping": "Ping Pong balls"};

for (var ball in balls) {
  console.log(ball+' = '+balls[ball]);
}
```

```
let sports=["golf","tennis","ping"];

for (var i=0; i<sports.length; i++){
  console.log(sports[i]);
}
```

Fonctions

Web

Les fonctions

Syntaxe de la définition d'une fonction

Avec une instruction :

```
function nomDeLaFonction( [arg_1 [, arg_2, ...] ] ) {  
    séquence d'instructions;  
    [return expression;]  
}
```

```
function sayHelloTo (nom) {  
    return "Hello "+nom+" ";  
}
```

```
function sayHello () {  
    return "Hello world";  
}
```

Les fonctions : l'appel

On l'appelle par son nom!

```
var retour=sayHelloTo("Vianney");  
alert(retour);  
console.log(retour);
```

```
"Hello Vianney "
```

Une page intégrée à l'adresse fiddle.jshell.net indique

Hello Vianney

OK

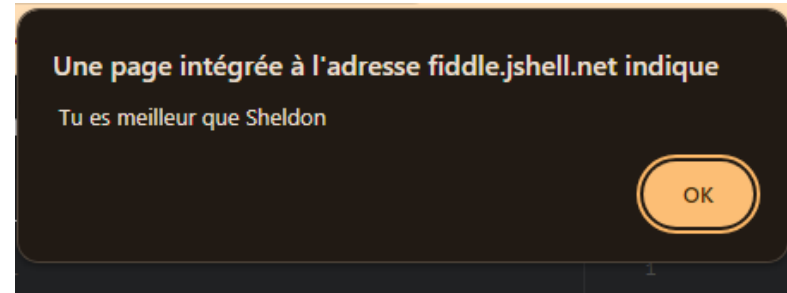
Interactions avec l'utilisateur

Web

Prompt et alert

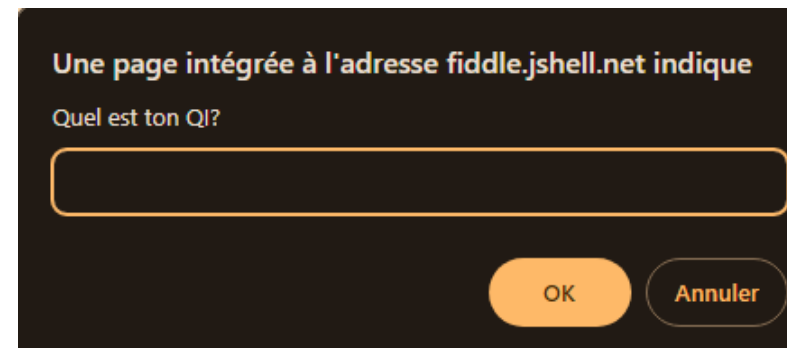
Alert nous l'avons déjà vu.

```
alert(" Tu es meilleur que Sheldon");
```



Prompt : permet de récupérer une entrée utilisateur :

```
var qi = prompt("Quel est ton QI?")
```



Ça code!!!

Web

A faire vous même 2

Vous avez dans le fichier exercicesJsEleves des exercices de différent niveau. Essayez de faire les niveau débutants / moyen

Ou moyen / confirmé

Tips

Web

La cheatsheet

<https://htmlcheatsheet.com/js>

Debug du code

Dans js fiddle vous pouvez faire ceci pour voir : `document.write(ce que vous voulez voir)`

```
document.write("<p>Hello World</p>");
```

Utiliser le débbuger dans Chrome

- Aller voir la source (F12).
- Cliquer sur la ligne du script où vous voulez mettre le point d'arrêt.
- Rechargez la page.

