



Atelier 3

Initiation à JavaScript

JavaScript

(Ne doit pas être confondu avec Java)

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web.

Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript dédié pour l'interpréter, indépendamment des considérations de sécurité qui peuvent se poser le cas échéant.

Intégration du JS dans le HTML

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
    initialscale=1, shrink-to-fit=no">
  <title>Cours JS</title>
</head>
<body>
  <h1>Cours JS</h1>
  <script>

  </script>
</body>
</html>
```

Déclaration de constantes et variables

```
let adresse ;
```

```
let age = 30;
```

```
console.log(age);
```

```
age = 35;
```

```
console.log(age);
```

```
console.log("Mon age est : "+age); // affichage soigné
```

Les types de données

```
const nom="Julien";  
const age = 33;  
const moyenne= 18.5;  
const isCool=true;  
const x=null;  
const y=undefined;
```

Les chaines de caractères

Concaténation

```
//affichage classique
console.log("Ma note en JS est : " + moyenne);
// JS moderne ES6
console.log(`Ma note en JS est : ${moyenne}`);
```

La longueur

```
console.log(`Mon nom est composé de ${nom.length} caractères`);
```

Fonctions de chaines de caractères

```
// nom en maj
console.log(`Mon nom en maj : ${nom.toUpperCase()}`);
// nom en min
console.log(`Mon nom en min : ${nom.toLowerCase()}`);
// extrait une sous chaine, Les index de 0 à 4 ==> 5 caract
console.log(`Sous chaine : ${nom.substring(0,4)}`);
```

(suite)

```
// prend les 4 caract et les transforme en maj  
console.log(`Sous chaine : ${nom.substring(0,4).toUpperCase()}`);
```

➔ chainage de fonctions : appliquer une fonction à la suite de l'autre

```
// decoupe la chaine en caractères  
console.log(nom.split(''));
```

```
const nomComplet = "Julien La Garde";
```

```
// decoupe la chaine selon un séparateur (dans ce cas, c'est l'espace)  
) ➔ 3 mots
```

```
console.log(nomComplet.split(' '));
```

Les tableaux

```
// decla de tableaux
const numbers = new Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
const fruits = ["mangue", "pastèque", "poire", "orange", "ananas"];

// acces a un element du tableau
console.log(fruits[0]); // acces au 1ere element
console.log(fruits[7]); // acces au 8eme element
fruits.unshift("fraise"); // ajout au debut du tableau
fruits.push("banane"); // ajout en fin de tableau
fruits.shift(); // supprimer le 1er element
fruits.pop(); // supprimer le dernier element
console.log(fruits.indexOf("orange")); // retourne la position de l'
element orange
console.log(Array.isArray(fruits)); // true
```

liste complète des fonctions de tableaux

https://www.w3schools.com/jsref/jsref_obj_array.asp

Les objets

```
const personne = {  
  prenom: "Julien",  
  nom: "La garde",  
  age: 30,  
  passions: ["foot", "natation", "jeux videos"],  
  adresse : {  
    rue : "rue Nationale",  
    ville : "Lille",  
    pays : "France"  
  }  
};
```

```
// afficher un objet  
console.log(personne);  
// acces aux attributs de l'objet  
console.log(personne.prenom); // acces au prenom
```

(suite)

```
console.log(personne.passions[0]); // acces au 1er element des passions
```

```
console.log(personne.adresse.rue); // acces a la rue (qui est lui meme un attribut de l'objet adresse')
```

```
personne.prenom="Julie"; // modifie la valeur d'un attribut
```

```
console.log(personne.prenom); // afficher la nouvelle valeur
```

```
personne.email="julie@example.com"; // ajout d'un nouvel attribut
```

```
console.log(personne); // affiche le nouvel attribut
```

(suite)

```
const todos = [
  {
    id: 1,
    text: "Faire les courses",
    isCompleted: true,
  },
  {
    id: 2,
    text: "Balade au vieux Lille"
  },
  {
    id: 3,
    text: "Préparer le diner",
    isCompleted: false,
  },
  {
    id: 4,
    text: "Regarder la TV",
    isCompleted: false,
  },
];

// afficher un tableau d'objets
console.log(todos);

// afficher l'attribut text du
1er objet du tableau
console.log(todos[0].text);
```

Les boucles

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

```
let i = 0;  
while (i < todos.length) {  
  document.write(todos[i].text + "<br>");  
  i++;  
}
```

```
todos.forEach(function (todo) {  
  document.write(todo.text + "<br>");  
});
```

```
const listeTodo=todos.map(function (todo) {  
  return todo.text;  
});
```

Les structures de contrôle

If / else

Syntaxe générale :

```
if (condition1) {  
    // traitement si  
    condition 1 est vraie  
} else if  
(condition2) {  
    // traitement si  
    condition 2 est vraie  
} else {  
    // Le reste des  
    cas  
}
```

```
const x = 14;  
const y = 10;
```

```
if (x == y) {  
    console.log("les deux nombres sont  
    égaux");  
} else if (x > y) {  
    console.log("x est supérieur à y");  
}  
else {  
    console.log("x est inférieur à y");  
}
```

```
if (x > y && x < 20) {  
    console.log("nombre autorisé");  
}
```

(suite)

L'opérateur ternaire

```
const message = x > 10 ? "commande permise" : "commande non permise";  
console.log(message);
```

Switch

Syntaxe

```
switch(expression) {  
  case x:  
    // traitement  
    break;  
  case y:  
    // traitement  
    break;  
  default:  
    // traitement  
}
```

(suite)

```
const age = 10;

switch (age) {
  case 18:
    console.log("majeur");
    break;
  case 60:
    console.log("retraité");
    break;
  default:
    console.log("autres");
}
```

```
switch (true) {
  case age > 0 && age <= 12:
    console.log("enfant");
    break;
  case age >= 13 && age < 18:
    console.log("adolescent");
    break;
  case age >= 18:
    console.log("majeur");
    break;
  default:
    console.log("erreur");
}
```

Les fonctions

```
// implementation d'une fonction  
// fonction avec param et retour de valeur  
function somme(a, b) {  
    return a + b;  
}  
  
// fonction sans param et sans retour de valeur  
function affichageDate() {  
    const date = new Date();  
    document.write(  
        `aujourd'hui il est le ${date.getDate()}/${${  
date.getMonth()}/${date.getFullYear()}`  
    );  
}  
  
x = 5;  
y = 10;
```


(suite)

// appel des fonctions

```
total = somme(x, y);  
document.write(total);
```

```
document.write(somme(x, y));  
document.write(somme(12, 20));
```

```
affichageDate();
```

Fonctions fléchées

Syntaxe

```
(param 1, param 2, ... param n) => {  
  // instructions  
}
```

(suite)

Exemple :

```
const voitures = [  
  'Ferrari',  
  'Mercedes',  
  'BMW',  
  'Maserati'  
];  
console.log(voitures.map((voiture) => voiture.length));  
// Array(4) [ 7, 8, 3, 8 ]
```

Fonction classique vs fonction fléchée

<pre><i>/* fonction classique :</i> let somme = function (a, b) { return a + b; }; */</pre>	<pre><i>//Equivalent en fonction fléchée : ES6</i> let somme = (a, b) => a + b; console.log(somme(1, 2));</pre>
---	--

Le DOM

Récupérer des éléments du DOM

// recuperer l'element qui a l'id my-form

```
const form=document.getElementById("my-form");  
console.log(form);
```

// recuperer une collection d'elements qui ont la classe container

```
const conteneurs = document.getElementsByClassName("container");  
console.log(conteneurs);
```

// recuperer une collection d'elements qui ont l'attribut name

```
const inputName = document.getElementsByName("name");
```

// recuperer une collection d'elements qui ont la balise ul

```
const ul = document.getElementsByTagName("ul");
```

(suite)

```
// recuperer le 1ere element qui repond au selecteur CSS spécifié en param
```

```
const sectionContainer = document.querySelector("section.container");
```

```
// recuperer une collection d'elements qui ont repondent au selecteur CSS spécifié en param
```

```
const allInputs = document.querySelectorAll("input");
```

Manipulation du DOM

```
// supprimer l'element
```

```
document.querySelector(".para1").remove();
```

```
console.log(document.querySelector(".para1").textContent);
```

```
// modifie le contenu textuel d'un element
```

```
document.querySelector(".para1").textContent = "para modifié";
```

(suite)

```
// modifie le contenu textuel d'un element  
document.querySelector(".para1").innerText = "para modifié 2eme fois";  
  
// modifie le contenu html (on peut mettre des balises) d'un element  
document.querySelector(".para1").innerHTML = "<span class='p2'>para de  
classe p2 <span>";
```

Liste des attributs et méthodes du DOM

https://www.w3schools.com/jsref/dom_obj_document.asp

Les évènements

Partie HTML

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
  <title>Cours JS</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1 class="titre">Cours JS</h1>
  <p class="para1">Lorem ipsum dolor sit, amet consectetur
adipisicing elit. Illo maxime explicabo cupiditate ipsam
    autem. Estnecessitatibus, architecto esse quaerat hic repellat
libero nisi ad eos ex vero at in velit.
  </p>
```

(suite)

```
<section class="container">
  <form id="my-form">
    <h1>Add User</h1>
    <div class="msg"></div>
    <div>
      <label for="name">Name:</label>
      <input type="text" id="name" name="name">
    </div>
    <div>
      <label for="email">Email:</label>
      <input type="text" id="email" name="email">
    </div>
    <input class="btn" type="submit" value="Submit">
  </form>
  <ul id="users"></ul>
</section>
```

(suite)

```
<script src="main.js"></script>  
</body>  
</html>
```

```
// USER FORM SCRIPT
```

Partie JS

```
// Mettre les elements du DOM dans des variables  
const myForm = document.querySelector("#my-form");  
const nameInput = document.querySelector("#name");  
const emailInput = document.querySelector("#email");  
const msg = document.querySelector(".msg");  
const userList = document.querySelector("#users");  
  
myForm.addEventListener("submit", onSubmit);
```


(suite)

```
function onSubmit(e) {
  e.preventDefault();
  if (nameInput.value === "" || emailInput.value === "") {
    msg.classList.add("error");
    msg.innerHTML = "Merci de compléter tous les champs";

    setTimeout(() => {
      msg.classList.remove("error");
      msg.innerHTML = "";
    }, 2000);
  }
  else {
    const li = document.createElement("li");
    // li.appendChild(
    //   document.createTextNode(`${nameInput.value}:${emailInput
    //     .value}`)
    // );
  }
}
```

(suite)

```
li.innerHTML = `<strong>${nameInput.value}</strong>
${emailInput.value}`;

    userList.appendChild(li);

    nameInput.value = "";
    emailInput.value = "";
  }
}
```