

Les types de données en JavaScript

En JavaScript, il existe 7 types de valeurs différents. Chaque valeur qu'on va pouvoir créer et manipuler en JavaScript va obligatoirement appartenir à l'un de ces types. Ces types sont les suivants :

- `String` ou « chaîne de caractères » en français ;
- `Number` ou « nombre » en français ;
- `Boolean` ou « booléen » en français ;
- `Null` ou « nul / vide » en français ;
- `Undefined` ou « indéfini » en français ;
- `Symbol` ou « symbole » en français ;
- `Object` ou « objet » en français ;

Ce que vous devez bien comprendre ici est que les données vont pouvoir être manipulées différemment en fonction de leur type et qu'il est donc essentiel de les connaître pour créer des scripts fonctionnels.

Les types de valeurs `Null` et `Undefined`

Les types de valeurs `Null` et `Undefined` sont des types un peu particuliers car ils ne contiennent qu'une valeur chacun : les valeurs `null` et `undefined`.

La valeur `null` correspond à l'absence de valeur ou du moins à l'absence de valeur connue. Pour qu'une variable contienne `null`, il va falloir stocker cette valeur qui représente donc l'absence de valeur de manière explicite.

La valeur `undefined` correspond à une variable « non définie », c'est-à-dire une variable à laquelle on n'a pas affecté de valeur.

Cette définition peut vous paraître similaire à celle de `null` et pourtant ces deux valeurs ont une signification différente. Si on déclare une variable sans lui attribuer de valeur, alors son type sera `Undefined`. Si on déclare une variable et qu'on lui passe `null`, alors son type sera `Object`.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours JavaScript</title>
    <meta charset="utf-8">
    <meta name="viewport"
      content="width=device-width, initial-scale=1, user-scalable=no">
    <link rel="stylesheet" href="cours.css">
    <script src='cours.js' async></script>
  </head>
  <body>
    <h1>Titre principal</h1>
    <p>Un paragraphe</p>
    <p id='p1'></p>
    <p id='p2'></p>
    <p id='p3'></p>
  </body>
</html>
```

Présentation des opérateurs de comparaison

Comme je l'ai précisé plus haut, nous allons souvent construire nos conditions autour de variables : selon la valeur d'une variable, nous allons exécuter tel bloc de code ou pas.

En pratique, nous allons donc comparer la valeur d'une variable à une certaine autre valeur donnée et selon le résultat de la comparaison exécuter un bloc de code ou pas. Pour comparer des valeurs, nous allons devoir utiliser des opérateurs de comparaison.

On va pouvoir utiliser les opérateurs de comparaison suivants en JavaScript :

Opérateur	Définition
==	Permet de tester l'égalité sur les valeurs
===	Permet de tester l'égalité en termes de valeurs et de types
!=	Permet de tester la différence en valeurs
<>	Permet également de tester la différence en valeurs
!==	Permet de tester la différence en valeurs ou en types
<	Permet de tester si une valeur est strictement inférieure à une autre
>	Permet de tester si une valeur est strictement supérieure à une autre
<=	Permet de tester si une valeur est inférieure ou égale à une autre
>=	Permet de tester si une valeur est supérieure ou égale à une autre

Certain de ces opérateurs nécessitent certainement une précision de ma part. La première chose à bien comprendre ici est que les opérateurs de comparaison ne nous servent pas à indiquer au JavaScript que tel opérande est supérieur, égal, ou inférieure à tel autre opérande.

```
let x = 4; //On stocke le chiffre 4 dans x

/*Les comparaisons sont effectuées avant l'affectation. Le JavaScript va donc
*commencer par comparer et renvoyer true ou false et nous allons stocker ce
*résultat dans nos variables test*/

let test1 = x == 4;
let test2 = x === 4;
let test3 = x == '4';
let test4 = x === '4';
let test5 = x != '4';
let test6 = x !== '4';
let test7 = x > 4;
let test8 = x >= 4;
let test9 = x < 4;

alert('Valeur dans x égale à 4 (en valeur) ? : ' + test1 +
  '\nValeur dans x égale à 4 (valeur & type) ? : ' + test2 +
  '\nValeur dans x égale à "4" (en valeur) ? : ' + test3 +
  '\nValeur dans x égale à "4" (valeur & type) ? : ' + test4 +
  '\nValeur dans x différente de "4" (en valeur) ? : ' + test5 +
  '\nValeur dans x différente de "4" (valeur & type) ? : ' + test6 +
  '\nValeur dans x strictement supérieure à 4 ? : ' + test7 +
  '\nValeur dans x supérieure ou égale à 4 ? : ' + test8 +
  '\nValeur dans x strictement inférieure à 4 ? : ' + test9);
```

Les conditions `if`, `if...else` et `if...else if...else` en JavaScript

La structure de contrôle conditionnelle `if` est présente dans l'ensemble des langages de programmation utilisant les structures de contrôle et notamment en JavaScript.

La condition `if` est l'une des conditions les plus utilisées et est également la plus simple à appréhender puisqu'elle va juste nous permettre d'exécuter un bloc de code si et seulement si le résultat d'un test vaut `true`.

Créons immédiatement nos premières conditions `if` :

```
let x = 4;
let y = 0;

if(x > 1){
    alert('x contient une valeur supérieure à 1');
}

if(x == y){
    alert('x et y contiennent la même valeur');
}

if(y){
    alert('La valeur de y est évaluée à true');
}
```

vous devez savoir que toute valeur évaluée par le JavaScript dans un contexte booléen va être évaluée à `true` à l'exception des valeurs suivantes qui vont être évaluées à `false` :

- Le booléen `false` ;
- La valeur `0` ;
- Une chaîne de caractères vide ;
- La valeur `null` ;
- La valeur `undefined` ;
- La valeur `NaN` (« Not a Number » = « n'est pas un nombre »).

La condition `if...else` en JavaScript

La structure conditionnelle `if...else` (« si... sinon » en français) va être plus complète que la condition `if` puisqu'elle va nous permettre d'exécuter un premier bloc de code si un test renvoie `true` ou un autre bloc de code dans le cas contraire.

```
let x = 0.5;

if(x > 1){
    alert('x contient une valeur strictement supérieure à 1');
}else{
    alert('x contient une valeur inférieure ou égale à 1');
}
```

La condition if...else if...else en JavaScript

La condition `if...else if...else` (« si...sinon si...sinon ») est une structure conditionnelle encore plus complète que la condition `if...else` puisqu'elle va nous permettre cette fois-ci de générer et de prendre en charge autant de cas que l'on souhaite.

```
let x = 0.5;

if(x > 1){
    alert('x contient une valeur strictement supérieure à 1');
}else if(x == 1){
    alert('x contient la valeur 1');
}else{
    alert('x contient une valeur strictement inférieure à 1');
}
```

Exercice n°1 :

Ecrire le code d'un fichier HTML qui permet de saisir trois notes (DC1, DC2, DS) puis calcule et affiche la moyenne.

Exercice n°2 :

Ecrire le code d'un fichier HTML qui permet de saisir **le numéro du mois et l'année** puis **affiche le nombre** de jours.

Exercice n°3 :

Ecrire le code d'un fichier HTML **permettant d'afficher** la table de multiplication d'un entier compris entre 1 et 10.

Exercice n°4 :

Refaire l'exercice n°1 en utilisant une fonction **Moyenne**, **traiter toute les possibilités d'emplacement d'une** fonction :

- Entre <BODY> et </BODY>
- Entre <HEAD> et </HEAD>

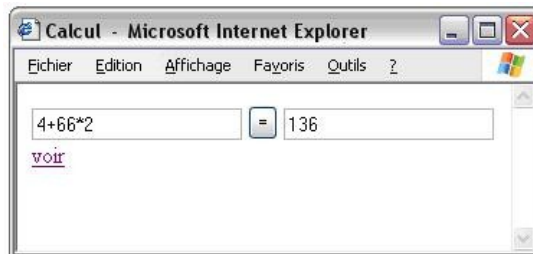
Exercice n°5 :

Ecrire le code d'un fichier HTML permettant de calculer et d'afficher le carré d'un nombre saisi au clavier.



Exercice n°6 :

Ecrire le code d'un document HTML, intitulé **Calcul**, contenant une zone de saisie qui permet d'entrer une opération à calculer, un bouton (=) pour afficher le résultat et une zone de réponse. Ajouter un lien hypertexte **permettant d'afficher le résultat**.



Exercice n°7 :

Créer un fichier HTML, intitulé **Formation**, contenant le formulaire suivant :

The screenshot shows a web browser window titled "FORMATION - Microsoft Internet Explorer". The browser's menu bar includes "Fichier", "Edition", "Affichage", "Favoris", "Outils", and "?". The main content area has a pink background and is titled "INSCRIPTION" in a large, bold, black font. Below the title, there are several form elements: a text input field for "Nom et Prénom *:", a text input field for "CIN *:", a text input field for "Adresse e-mail *:", a radio button group for "Niveau:" with options "Secondaire" (selected), "Universitaire", and "Autre", and a dropdown menu for "Modules de formation:" with options "Bureautique", "SE et Réseaux info.", and "TIC". At the bottom right, there is a note "(*) Champs obligatoires." and two buttons: "Envoyer" and "Annuler".

- Le clic sur le bouton "**Envoyer**" de type **submit** fait l'appel à une fonction "**Verif**" écrite en JavaScript et qui permet de vérifier les conditions suivantes :
 - Les champs Nom et Prénom, CIN et Adresse e-mail doivent être non vides.
 - Le champ CIN ne contient que 8 chiffres de 0 à 9.
 - L'**adresse** e-mail doit comporter le caractère @.
 - Le niveau Secondaire est sélectionné par défaut.
 - On peut choisir au maximum deux modules de formation.
- Si les conditions précédentes sont satisfaites, le message suivant sera affiché : "**Inscription achevée !!**"
- Le bouton "**Annuler**" permet de remettre à vide tous les champs du formulaire.