**Bolean** :

Bolean de 0 = false

Bolean de ‘’ = false

Bolean de -0 = false

Bolean de undefined = false

Bolean de null = false

Bolean de false = false

Bolean de NaN = false

**Condition** :

a>b ? x = ‘vrai’ : x = ‘faux’

* if (condition){}

Else if (condition){}

else{}

* switch () {

case \_ : résultat ;

break ;

case \_ : résultat ;

Break ;

Default : résultat}

**Compare fonction** :

(a,b){return a-b}

**Boucle ou loop**:

For (initialisation ; condition ; incrémentation) {traitement}

Let i ;

Let text = ’’ (important)

For (i = 0 ; i <10 ; i++) {text+=i} <<< syntaxe

On peut attribuer deux variable :

For (i = 0 ; i <10 ; i++) {text+=i ; text2+= text + i}

For in

Utile pour parcourir tout le tableau d’un objet. Donc la condition est for

Let x ;

(x in objet){ text+= object[x]} <<<< syntaxe

Exemple :

let lesgens = ["Youssefwarzone", "yassineraleur", "farouk", "leprof"];

let lesgens2 = ["marvin","mayan","youssefleboss"];

function test2(essai2) {

  let z;

  let test2 = "";

  for (z in essai2) {

    test2 += essai2[z] + "\n";

  }

  return test2;

}

console.log(test2(lesgens2));

J’ai mis deux variables (lesgens, lesgens2) pour que, peu importe la formule que j’ai mis, elle s’applique au deux.

Ensuite on commence la fonction avec deux variante pour quelle s’applique dans ma formule.

For (z in essai2) : ça veut dire dans le tableau z (les crochets veulent dire tableau), dans la fonction essai2.

On ouvre les accolades, on prend notre variable(vide) += (à regarder dans la doc <https://www.w3schools.com/js/js_assignment.asp>), la formule de la fonction’’essai2’’, dans le tableau + à la ligne.

Retourne-nous la valeur de la fonction test2

Console.log(test2(lesgens2)), Qu’il m’affiche le résultat de la fonction test2 dans la variante ‘’lesgens2’’.

forEach()

Applique une fonction donnée à toutes les valeurs d’un objet :

objet.forEach(function)

for of

Comme for in : for (let x of object) {text+= x}

break

Fait quitter une boucle :

For (i = 0; i < 10; i++) {

if (i === 3) { break; }

text += "The number is " + i + "<br>";

}

Continue

Fait sauter juste l’itération cibler :

for (i = 0; i < 10; i++) {

if (i === 3) { continue; }

text += "The number is " + i + "<br>";

}

While

A la différence du for l’incrémentation se fait manuellement :

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

Tableau indexé JS

Var cars = ["Saab", "Volvo", "BMW"] ;

Cars [0] = "Opel’’ ;

Dans cette exemple on remplace la valeur Opel par Saab

Pour qu’il s’affiche on met console.log et on met le nom du tableau avec le numéro de la case choisi.

Les tableaux sont des objets, par ce fait, ils ont des attributs et des méthodes.

.sort() = Sa affiche par ordre alphabétique ou par ordre de chiffre. <https://www.w3schools.com/js/js_array_sort.asp>

Pour faire le décroissant on laisse short mais dans la formule,

Au lieu de mettre a-b, on met b-a.

Si on met console.log(le nom du tableau) y va afficher tout le tableau dans toute sa conception

fruits = ["Banana", "Orange", "Apple", "Mango"];

var last = fruits[fruits.length - 1];

fruits.length = tout le contenu du tableau (de 0 à 3)

4 valeurs, donc on met la valeur -1 pour qu’il commence par 0

.push = rajouter un élément au tableau

Exemple

var fruits = ["Banana", "Orange", "Apple", "Mango"];

Fruits.push(‘’Lemon’’) ;

.pop c’est de supprimé le dernier élément et si on l’affecte à une variable, y la prend en compte

.shift supprime le premier élément.

.unshift ajoute un élément en premier et bouscule les autres.

.splice() Supprimer des éléments sans faire de trou.

Il va afficher : Banana,Orange,Apple,Mango,Lemon

Tableau associatif JS

var person = [];

person["firstName"] = "John";

person["lastName"] = "Doe";

person["age"] = 46;

var x = person.length; // person.length will return 0

var y = person[0];

Sur le tableau ci-dessus, il y a 3 cases (firstname, lastname , Age)

Le nom du tableau c’est ‘’person’’

Pour accéder à la 3cases, on ne fait pas [3] mais plutôt le nom de la case [‘’Age’’].

C’est la même syntaxe que le premier tableau.

Pour changer le prénom on fait la manipulation suivante :

Person [‘’firstName’’] = ‘’Nouveau Blaz’’ ;

Pour les chiffres ou les nombres on met pas de guillemet, rien.

Pour déclarer un tableau, il faut les [] et à l’intérieur on y implante des objets entre {}.

Les objets peuvent avoir une structure différente.

On peut mettre un tableau dans un tableau (tableau multijesaisplus)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Rouge : case 1 (un tableau)

Bleu : case 2 (un tableau)

Vert : case 3 (un tableau)

Le terme ‘’new’’ c’est la déclaration d’un nouvel objet

Pour afficher une date on met :

var d = new Date();

Il existe 4 façons de créer un nouvel objet de date :

new Date()

New Date (year, month, day, hours, minutes, seconds, milliseconds)

New Date(milliseconds)

New Date (date string)

Exemple :

Var d = new Date(2018, 11, 24, 10, 33, 30, 0);

24 décembre 2018 10h33.30.0

Les dates sont indexées de 0 à 11 (Janvier=0)

Les années à un et deux chiffres seront interprétées comme 19xx:

var d = new Date(99, 11, 24);

Sinon la date on peut l’écrire de cette façon :

var d = new Date("October 13, 2014 11:13:00");

Il existe généralement 3 types de formats de saisie de date JavaScript:

Date ISO "2015-03-25" (La Norme internationale)

Date courte "25/03/2015"

Date longue «25 mars 2015» ou «25 mars 2015»

Ne pas oublier de mettre les mois et jours sur 2 caractères

Si on veut afficher juste l’année par exemple on met le ‘’.’’ Ensuite ‘’get’’ et la méthode.

<https://www.w3schools.com/js/js_date_methods.asp>

exemple :

apres notre formule on met ‘’ d.getMonth() + 1;’’ y va nous affiché ‘’2’’ comme le mois de février

var d = new Date();

var months = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"];

document.getElementById("demo").innerHTML = months[d.getMonth()];

New Date, il va nous afficher la date d’aujourd’hui en entier avec les secondes et miliseconde.

Pour nous affiché juste le mois, on lui met ca ‘’months[d.getMonth()];’’

Il nous affichera février parce qu’on est en février à l’heure où j’écris ce texte.

GET = récupérer

SET = Ajuster (exemple : ajouter une nouvelle valeur)

Exemple pour SET :

<script>

var d = new Date();

d.setFullYear(2020);

document.getElementById("demo").innerHTML = d;

Avec la formule ‘’ d.setFullYear’’, il va ajuster la date et il va lui mettre la valeur 2020.

str.indexOf("locate"); permet de localiser le terme précis et il nous affiche en nombre .

Exemple :

var str = "Please locate where 'locate' occurs!";

var pos = str.indexOf("locate");

document.getElementById("demo").innerHTML = pos;

La réponse est 7, car le début de locate, se trouve dans le 7eme caractère

On peut aussi le chercher avec la méthode search().

EVENEMENTS JS

DOM : Document Objet Model

(Tout notre document, avec toutes les balises)

C’est le système d’emboitement.

Exemple : L’objet H1 se trouve dans header et header se trouve dans body et l’objet body se trouve dans html …

Les évènements c’est tout ce qu’on fait dans le DOM

(Survol, click, écrire, …)

Exemple : quand je survole le bouton, cette fonction sera appliquée.

Les balises script on le met sur html sinon on utilise une page js