

attribut	Id, href, src, width, name	monImage.width monImage.src = image.jpg
	OBJET STRING ET SES METHODES	
toLowerCase() toUpperCase()	Converti en minuscule Converti en majuscule	var a = "J'apprends JAVASCRIPT"; a.toLowerCase(); -> j apprend javascript
charAt()	Retourne un caractère situé à une position donnée. (l'index)	var position0 = a.charAt(0)
indexOf()	Va retourner la 1ère position à laquelle un caractère donné a été retrouvé dans une chaîne	var prems = a.indexOf('p') -> 3 (voir ligne 2)
lastIndexOf()	Retourne la dernière position où se trouve ce caractère	var fin = a.lastIndexOf('p') -> 4
replace()	rechercher un caractère et le remplacer	a.replace("JAVASCRIPT", "PHP"); -> J'apprends PHP
slice()	Va extraire une partie de chaîne et retourner la partie extraite comme nouvelles chaîne.	var coupe = a.slice(0, 10); -> J'apprends
trim()	Supprime les espaces superflus en début et en fin de chaîne	var a = " aaaaaa. "; a.trim(); -> "aaaa"
	L'OBJET MATH ET SES METHODES	
Math.parseInt(nombre1)	Converti en nombre entier	var a = Math.parseInt("10.33") -> 10
Math.parseFloat(nombre1)	Converti en nombre décimal	Var a = Math.parseFloat("40 ans"); -> 40
Math.random()	nombre aléatoire entre 0 et 1	-> 0.879789797
Math.round()	Arrondi vers l'entier le plus proche	var a = Math.round(12,7); -> 13
Math.floor()	Arrondi vers le bas	var a = Math.floor(12,7); -> 12
Math.ceil()	Arrondi vers le haut	var a = Math.ceil(12,7); -> 13
Math.min()	Retourne le plus petit nombre parmi une série de nombres	var a = Math.min(0, -10, 200); -> -10
Math.max()	Retourne le plus petit nombre	
Math.abs()	Retourne la valeur absolue d'un nombre (sans le signe -)	var a = Math.abs(-10); -> 10
	L'OBJET DATE ET SES METHODES	
Date()	Sous forme littérale	var a = Date(); -> Tue Aug 18 2017 15:25:43 GMT + 0200
Date.now()	nombre de ms depuis le 1/1/1970	var a = Date.now(); -> 1439904463163
DOM	DOM	DOM
document.ELEMENT_NODE	les balises html	
document.TEXT_NODE	Le texte (contenu de la balise)	
.innerHTML	Ecrire dans l'html/vider un contenu html	a.innerHTML+= "ajout texte";
.textContent	Le texte sans les balises. Permet de modifier le texte	a.textContent = "un li de plus"; titre.textContent += "ajout de mot dans le titre";

querySelector("li a ")	Sélecteur css	
getElementsByTagName("p")	Renvoie toutes les balises (éléments html) sous forme de tableau	Document.getElementsByTagName('p')[5]
.appendChild()	Ajouter un enfant en dernier (par rapport au papa)	getElementById('papa').appendChild(a);
.childNodes	Accéder aux enfants d'un noeud choisi	document.body.childNodes[1]
.parentNodes	Retourne un Node qui est le parent de ce nœud.	
.lastChild	Retourne un Node représentant le dernier nœud enfant direct de ce nœud	
.nextSibling	Retourne un Node représentant le nœud suivant dans l'arbre.	
.createElement("li")	Créer un élément	Var a = document.createElement("li");
.setAttribute()	définir un attribut (id, class, name, etc)	a.setAttribute("class", "nomClasse");
.contains()	Vérifie si la classe existe	a.classList.contains("nomClasse");
.classList()	Pour accéder, ajouter ou supprimer des classes	a.classList.remove("nomClasse"); a.classList.add("nomClasse");
.createTextNode()	Crée un nouveau noeud de type node	a.appendChild(document.createTextNode("Ruby"));
.insertBefore()	Insérer un noeud avant un autre noeud	document.getElementById("papa").insertBefore(a, document.getElementById("liSuivant"));
.insertAdgacentHTML()	Choix de la position exacte d'un nouveau noeud	document.getElementById('langages').insertAdjacentHTML("afterBegin", 'JavaScript');
.removeChild()	Supprimer un noeud existant	document.getElementById("langages").removeChild(document.getElementById("bash"));
TABLEAU	TABLEAU	TABLEAU
	dernier élément du tableau	var dernier = tableau[tableau.length - 1];
push()	Ajouter à la fin du tableau	array.push('quelque chose');
pop()	Supprimer le dernier élément du tableau	array.pop();
shift()	Supprimer le premier élément du tableau	
unshift()	Ajouter au début du tableau	array.unshift("quelque chose");
indexOf()	Trouver l'index d'un élément dans le tableau	array.indexOf('banane'); -> 2

splice()	Supprimer un élément par son index	fruits.splice(pos, 1); ->supprime 1 élément à la position pos
Array.sort()	Trie par ordre croissant(alphabétique)	
For(var indice in array)	Parcourir un tableau associatif	
EVENEMENTS	EVENEMENTS	EVENEMENTS
type	type d'évènement(click, scroll, keydown...)	e.type -> keydown
target	L'objet qui a envoyé l'évènement	e.target.textContent -> text du bouton
click onDbClick	Clic / Double clic	.addEventListener("click", function(e);
keypress / onkeypress	Lorsqu'on appuie sur une touche du clavier	.addEventListener("keypress", function() {}); <button onkeypress="myFunctions();">
keydown/ onkeydown keyup / onkeyup	Lorsqu'une touche est enfoncée/ relâchée	
mouseover/ onmouseover mouseout	Lorsque le pointeur de la souris arrive/sort de l'élément.	
mousemove	Lors du déplacement de la souris au dessus d'un élément	
keypress / keydown	Quand on appuie/lache sur une touche	
mousedown/ mouseup	Quand on appuie/lache la souris	
e.keyCode	Le numéro de la touche pressée	e.keyCode -> 13 (touche enter)
e.stopPropagation();	Stopper la propagation d'un e	
e.preventDefault();	Bloquer le comportement par défaut. (Envoi formulaire, a, etc..)	
.addEventListener	Ajouter un évènement sur un id(ex)	.addEventListener("click", function (){});
.removeEventListener	Ne regarde plus l'évènement	

Pour un compteur : penser à incrémenter. compteur++;

Debugger

undefined : la variable est déclarée mais on ne lui a jamais donné de valeur.

null : objet vide

Boucles

switch

```
switch(choix){  
  case "oui" : console.log('kjhkjh');  
  break;  
  
  default : console.log('kjhkjh');  
  break;  
}
```

```
switch ( new Date().getDay() ) {  
  case 6:  
    text = "Saturday"; // if (day == 6)  
    break;  
}
```

while(condition)

do ... while : les instructions seront toujours exécutées au moins une fois. Ex: on demande à l'utilisateur une chaîne de caractère tant qu'il ne clique pas sur annuler.

```
var msg;  
do  
  msg = prompt("Entrez un message ou cliquez sur annuler");  
while(msg);
```

Boucles : parcourir un tableau

for : pour un tableau numéroté (simple)

```
var div = document.getElementsByTagName('div');  
for(var i=0; i< div.length; i++){  
  div[i].style.backgroundColor = couleur;  
}
```

forEach : Pour un tableau associatif. Permet d'exécuter une fonction donnée sur chaque élément du tableau.

```
var tableau = { "un" : 1, "deux" : 2}  
tableau.forEach(function(element){  
  console.log(element);  
});
```

for in : pour les objets

```
for(var element in tableau){  
    console.log(tableau[element]);  
}
```

Les Objets

```
var stylo = {  
    type: "bille",  
    couleur: "bleu",  
    marque: "Bic",  
  
    id : function(){  
        return 'type : ' + this.type + ' couleur : ' + this.couleur + ' marque : ' + this.marque;  
    }  
};  
  
console.log("Mon stylo à " + stylo.type + " " + stylo.marque + " écrit en " + stylo.couleur);  
alert(stylo.id());
```

Formulaire : accéder aux éléments grâce à leur attribut name.

```
var form = document.querySelector("form");  
form.addEventListener("submit", function (e){  
    var mdp1 = form.elements.mdp1.value;  
    console.log(mdp1);  
    e.preventDefault();  
});
```

elements : la balise form possède une propriété elements qui regarde les inputs.

regex : Expression régulière. Vérifie si existe dans une chaîne de caractère.

`/.+@.+\.+\/` -> `xxx@yyy.zzz` (pour vérification des mails)

`/\d+\/` (cherche si il y a un nombre)

change (pour les checkbox, liste déroulante...). On accède à l'événement grâce à change.

```
document.getElementById("nationalite").addEventListener("change", function (e) {  
    console.log("Code de nationalité : " + e.target.value);  
});
```

Avoir la date du jour

```
var today = new Date();
var dd = today.getDate();
var mm = today.getMonth()+1; //January is 0!
var yyyy = today.getFullYear();

if(dd<10) {
    dd = '0'+ dd
}

if(mm<10) {
    mm = '0'+ mm
}

today = dd + '/' + mm + '/' + yyyy;
document.write(today);
```

-> 01/06/2018

getDate() Les jours du mois (1-31)

getDay() Les jours de la semaine entant que nombre(0-6)

getFullYear() (yyyy)

AJAX

	Description	Exemple
METHODE		
new XMLHttpRequest()	Objet qui permet de faire des requêtes	var ajax = new XMLHttpRequest();
abort()	Stopper la requête	
open(method,url,async)	Prend 3 paramètre : La méthode d'envoi, l'url/ chemin vers le fichier/url qui reçoit la requête, true = asynchrone (ne bloque pas le script si pas de réponse)	ajax.open("POST", 'core/request.php', true);
getResponseHeader()	Retourne les infos de l'entête http	
send() send("string")	Envoyer au serveur	ajax.send(); ajax.send("functionMontre");
setRequestHeader(header, value)	Pour définir un en-tête de requête HTTP, cet en-tête spécifie les données envoyé avec send. header=nom de l'en-tête, la valeur associée	xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
PROPRIETE		
onreadystatechange	Regarde le status de la requête (finie, erreur?)	ajax.addEventListener('readystatechange', function() {} ajax.onreadystatechange = function() {
readyState	Le statut de la requête. 1 = connecté au serveur. 4 = finie et prête	(if ajax.readyState === 4)
responseText	renvoie la réponse du serveur en tant que chaîne JavaScript	a.innerHTML = xhttp.responseText;

	Description	Exemple
METHODE		
status	Retourne un nombre sur le statut de la requête. 200 = OK, 404 not Found	(if ajax.status === 200)

GET

```
function loadDoc() {
  var xhttp = new XMLHttpRequest(); // crée l'objet qui échange des données avec un serveur
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
```

POST

Lorsqu'on utilise la méthode POST il faut absolument changer le type MIME de la requête avec la méthode `setRequestHeader`.