|  |  |  |
| --- | --- | --- |
| attribut | Id, href, src, width, name | monImage.width  monImage.src =   image.jpg |
| parseInt(nombre1) | Converti en nombre entier | var a = parseInt("10.33") -> 10 |
| parseFloat(nombre1) | Converti en nombre décimal | Var a = parseFloat("40 ans); -> 40 |
| Math.random() | nombre aléatoire entre 0 et 1 | -> 0.879789797 |
| Math.floor() | Arrondi |  |
| 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 ") | Sélécteur 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.classList.contains("nomClasse"); contains ->vérifie si la classe existe |
| .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]; |
| foreach() | Parcourir un tableau avec une fonction | array1.forEach(function(item, index, array) {,console.log(item, index);,}); |
| 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 onDblClick | Clic / Doucble 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/relachée |  |
| 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 | correction |  |

Un formulaire : accède aux éléments grâce à leur attribut name

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

**Debugger**

indefined : 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;

}

while(condition)

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

var msg;

do

    msg = prompt("Entrez un message ou cliquez sur annuler");

while(msg);

**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;

}

**for in** : pour un tableau associatif

for(var element in tableau){

conslole.log(tableau[element]);

}

**foreach** : permet d’exécuter une fonction donnée sur chaque élément du tableau

var tableau = ['a', 'b'];

tableau.foreach(function(element){

console.log(element);

});

**api**

XMLHttpRequest:

est un objet javascript qui permet de récupérer des données via HTTP. Pour l’utiliser il suffit de créer une instance de cet objet, d’ouvrir une URL et d’envoyer la requête

const req = new XMLHttpRequest();