

Pratique

```
alert(message)
var nomvar123 = 4
console.log( "la variable vaut" + nomvar123 )
/ commentaire /
if( typeof variable == "undefined" ) { }
<script src="chemin/fichier.js"> </script>
```

Maths

```
division = Math.floor(y/x);
modulo = y % x;
Math.pow(2, 3)
num="123"
parseInt( num);
var num = 5.56789;
console.log( num.toFixed(2) ); // 5.57
Attention + est aussi opérateur de concaténation de chaînes
```

Variables et tableaux

```
var tab = [ ]
tab.push( 3 );
tab.length
string :
var info = "Bonjour à tous" ;
info.length
info[1] == 'o'
info.indexOf( 'o' ) = 1 ( -1 si pas trouvé)
tab = Array( 1, 2, 3, true, "Bonjour" )
for( var= i=0 ; i < 5 ; i++ ) { }
for( var key in tab )
{ console.log( "tab[" + key, "] = ", tab( key) ); }
/ le tableau a un index forcément numerique entier. /
tableau : indices 0 à n-1
Il peut avoir des trous : tab[1000] = value
```

Manipulation du DOM

```
document.getElementById( 'testid' );
/ ajout suppression de classe css /
document.getElementById("monElement").classList.add("maClasse", "other");
document.getElementById("monElement").classList.remove("maClasse", "other");
if (
document.getElementById("monElement").classList.contains("maClasse") )
document.getElementById("monElement").classList.toggle("maClasse");
function checkInput( self) {
    self.style.boxShadow = " 0 0 3px blue" ; /* hallo de l'élément ayant le focus du clavier souris"
    self.style.borderColor = "1px solid blue" ;
}
element.innerHTML = "duCode html"
var duCodeHtml = element.innerHTML;
var elem = document.createElement('div');
elem.style.cssText =
'position:absolute;width:100%;height:100%;opacity:0.3;
z-index:100;background:#000';
document.body.appendChild(elem);
// Get the element with id="myDIV" (a div), then get all <h2>, <div> and <span> elements inside myDIV
var tabElts =
document.getElementById("myDIV").querySelectorAll("h2, div, span");
for ( var i in tabElts ) // mauvais choix : ça ne marche pas sur un objet
for ( var i = 0 ; i < tabElts.length ; i++ )
{
    tabElts[i].style.backgroundColor = "red";
}
}
```

json et tableaux associatifs

```
var element = {} ;
element.toto.titi
element[ "toto" ]["titi" ]
element[ "toto" ].titi
```

Evénements

```
element.addEventListener( "click", unefonction );  
onfocus  
onblur (je perds le focus)  
onChange après modification  
onError   
onDbClick (en double-cliquant)  
onMouseDown  
onMouseMove  
onMouseOver  
onMouseOut  
onLoad  
onUnload  
onReset
```

possible aussi directement dans le code html :

```
<button id="id1" onclick="javascript:code js, mais de préférence une  
fonction js" >
```

fonctions

```
function nomdelafonction( param1, param2 ) { code };  
var nomdelafonction = function( param1, param2 ) {  
code };
```

paramètres d'entrée non typés, nom comptés...

Structures de contrôle

```
if ( ) { } ;  
if ( ) { } else { } ;
```

