**Tableau :**

<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects/Array>

// Déclaration :  
var tab1 = [];  
var fruits = ['Apple', 'Banana'];

// Les clés d’un tableau sont en réalité des String :  
tab[2] == tab[‘2’] mais tab[2] != tab[‘02’]

// Accès aux éléments :  
tab1[0] équivalent à tab1.0

// Taille : fruits.length

// Boucler sur un tableau :  
tab.forEach(function(item, index, array) {  
 console.log(item, index);  
});

//Ajouter à la fin du tableau :  
let newLength = fruits.push('Orange');

// Supprimer le dernier élément du tableau :  
let last = fruits.pop();

// Supprimer le premier élément du tableau :  
let first = fruits.shift();

// Ajouter au début du tableau :  
let newLength = fruits.unshift('Strawberry')

// Trouver l'index d'un élément dans le tableau :  
let pos = fruits.indexOf('Banana');

// Supprimer un élément par son index :  
let removedItem = fruits.splice(pos, n);  
// supprime n éléments à la position pos

**switch :**

switch (variable) {  
 case 'Oranges':  
 console.log('Oranges');  
 break;  
 case 'Mangoes':  
 case 'Papayas':  
 console.log('Mangoes and ');  
 break;  
 default:  
 console.log(`Sorry`);  
}

**Fonctions :**

function calcRectArea(width, height) {  
 return width \* height;  
}

calcRectArea = (width, height) => {  
 return width \* height;  
}

// Une fonction sans paramètre peut s'écrire avec un couple de parenthèses  
() => {  
 instructions  
}

**Boucles :**

for ([init]; [condition]; [Incrément])  
 instruction

while (condition)  
 instruction

do  
 instruction  
while (condition);

// Boucle sur les valeurs d’un tableau :  
for (item of tab) {  
 console.log(item);  
}

// Boucle sur les propriétés d’un objet :  
for (i in objet) {  
 console.log(objet[i]) ;  
}

**String et characters :**

// Boucle sur les caractères :  
for (i in chaine) { …

// test si la chaine contient une autre chaine  
if (chaine.includes(‘xxx’) { …

// Concatenation tableau dans chaine:  
var tab = ['Bonjour', ' ', 'Alfred'];  
"".concat(...tab);

// Parse hex value :  
parseInt(c,16)

// Caractère à partir de la valeur :  
String.fromCharCode(65)  
String.fromCharCode(72, 69, 76, 76, 79);

// Integer to hex :  
yourNumber.toString(16);

**Lambas :**

(param1, param2, …, param2) => expression

// équivalent à  
(param1, param2, …, param2) => {  
 return expression;  
}

// Parenthèses non nécessaires quand il n'y a qu'un seul argument  
param => expression

**Code Golf :**

<https://codegolf.stackexchange.com/questions/37624/tips-for-golfing-in-ecmascript-6-and-above>

Général : suppression var/const/let, suppression « ; » (mais pas le retour chariot)

// Déclarer plusieurs variables x, y et un tableau s :  
x,y,s=[]

// Affectation tableau dans variables :  
[a,b]=y()

// Reduction taille d’une fonction std :  
z=parseInt

// Ecriture console :  
print(s.join(' '))

// Fonction sous forme lambda :  
y=(v)=>readline().split(' ')

// Boucle sans fin :   
for (;;) {

Précédence des opérateurs : <https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Operators/Operator_Precedence>

++ -- ! \*\* \* / % + - < <= > >= in == != && || (affectation =)