**Hoisting** всплытие. Где бы ни была прописана функция, она всегда по умолчанию подымается в самый верх скрипта. Т.е. вызывать аргументы функции можно выше, чем записана сама функция.

asd()

function (asd) {

console.log(‘Hello’) } // Будет выведено Hello

Если переменной назначить функцию, то аргумент выше нельзя прописать:

Let greeting = function () {

console.log(‘Hello’) // Это **function expression**

}

greeting()

function calc(a, b) {

return a+b; // Это **function declaration**

}

**Стрелочные функции.**

let calcul = (c,d) => c+d  
***console***.log(calcul(22,33))

let calc = (a, b, action) => {  
 let rezult  
 if(action === '+') {  
 rezult = a + b;  
 } else if(action === '-') {  
 rezult = a-b;  
 }  
 return rezult;  
}  
***console***.log(calc (10,5, '-'))

**Рекурсия .** Если есть массив, который постоянно меняется, дополняется. Этот массив нужно перебрать, но не циклом, а функциями. Тогда используется рекурсия. Рекурсия перебирает массив и выводит все елементы в ряд.

let ***arr*** = [1,2,3,[4,5], [6,7,8]];  
let ***flatArray*** = [];  
function flatter (array) {  
 for (let item of array) {  
 if(***Array***.isArray(item)) {  
 flatter(item);  
 }  
 else {  
 ***flatArray***.push(item) // push вставляет элемент в конец массива  
 }  
 }  
}  
flatter (***arr***)  
***console***.log(***flatArray***) // Будет выведено [1, 2, 3, 4, 5, 6, 7, 8]

**Функция callback** Это протокол как должна работать другая функция. Например, музыкальный проигрыватель — это функция, а кассета для него – это колбек.

function foobar (rrr) {  
 rrr();  
}  
let xxx = function () {  
 ***console***.log('hello');  
}  
foobar(xxx) // rrr - это колбек. Он выводит функцию ххх. Вызывается rrr, а показывает xxx

let users = [  
 {name: 'vasua', age : 29, status: false},  
 {name: 'kolya', age : 30, status: true},  
 {name: 'petya', age : 31, status: false},  
 {name: 'olya', age : 32, status: true}  
]  
let filter = users.filter(**function (user) {  
 return user.age > 30;** // Это колбек.  
})  
console.log(filter);

let filter = users.filter((allname) => allname.name === 'olya');  
console.log(filter) // Пример колбека с помощью стрелочной функции

let filter = users.filter(function(allname) {  
 return allname.name === 'olya'  
});  
console.log(filter) // Тоже самое НЕ стрелочным методом

**Метод filter()** создаёт новый массив, отфильтровывая все элементы, которые не проходят по условиям при выполнении callback() функции. Под капотом, filter() итерирует по каждому элементу в массиве и передаёт каждый из них самой callback() функции. Если callback() выдаёт true, то этот элемент будет включен в финальный массив.

Array.filter(callback, contextObject)