**Date – шаблон**. Необходимо добавить ключевое слово **new**

let now = new Date();  
console.log(now) // Выводит сегодняшнее время

console.log(now.getFullYear()) // Выводит год  
console.log(now.getMonth()) // Выводит месяц. Февраль = 1

let date = new Date(1990,0,30,12,10)  
console.log(date) *// Создаем и выводим конкретную дату*

let user = {  
 id: 1,  
 name: 'kokos',  
 bDay: new Date(124365743838)  
}  
console.log(user) *// Выводим данные user. id, name, день*

**try catch**

console.log('start');  
let obj = null;  
try {  
 console.log(obj.id);  
}catch (error) { *// try catch используем когда считаем что будет* console.log(error) *// ошибка, которая заблокирует дальнейшую работу кода.*}  
console.log('end') // try – Если будет ошибка, то catch ее выведет

**throw new** Выводит новую ошибку

function *divider* (a,b) {  
 if (b===0) {  
 throw new *Error*('b cannot be ZERO');  
 }  
 let result = a/b;   
 return result;  
}  
try {  
 *divider* (10,0)  
} catch (e) {  
 */// Сделать что то. Если будет делить на ноль, то сработает* throw new *Error*}

**Оператор опциональной последовательности ?.**

function *asd*(obj) {  
 console.log(obj?.inner) *// Мы не указали параметр inner у obj. Поэтому будет ошибка*} *// Но если поставить ?. это будет одноглазый Елвис.  
asd*({}) *// Он пропускает возможную ошибку и код проверяется дальше.*

**Short Hand. Сокращения**

let name = 'kokos';  
let age = 123;  
let user = {  
 name, *// Это эквивалентно: name: name,* age, *// Это эквивалентно: age: age* foobar () {  
  
 },  
}  
console.log(user)

**Деструктуризация**

let car = {  
 model: 'subaru',  
 power: 123,  
 drive() {  
 },  
 manufacturer: 'fudji'  
}  
let {model, power, *drive*} = car; *// Деструктуризация*console.log(model); *// Это эквивалентно let model = car.model;*

function *extractor*({model}) {  
 console.log(model)  
}  
*extractor*({model: 'asd', a: 123, b: 345, c: 'fff'}) *// Достает из функции значение model. Сокращаем объект, мы не пишем его имя, а только содержимое.*

**SPRED. Размазывание.** С помощью трех точек {…user} мы обозначаем, что в объекте user есть много ключей-значение, которые нужно все вытянуть. Что бы не прописывать их вручную – есть spred.

let user = {  
 name: 'kokos',  
 age: 31,  
 status: true  
}  
let userCopy = {...user} *// Если мы присваиваем другой переменной все значения user*console.log(userCopy) *// Нам не нужно все копировать, достаточно выбрать СПРЕД {...user}  
 // Важно, что значения тут будут одинаковые, но переменные сами по себе разные.*

**JSON. Копии обьектов**

Функции скопировать нельзя

let user = {name: 'vasya'}  
console.log('{"name":"vasya"}') *// JSON, который прописывают вручную*let s = JSON.stringify(user) *// Переменной s присваиваем JSON без ручной обработки и переводим ее в стрингу*console.log(s)  
  
let parse = JSON.parse(s); *// Возвращаем обратно стрингу в обьект*console.log(parse);

**Метод assign**

Поверхносное клонирование объектов