

# МОДУЛЬ 11

ПРАКТИКА



# ЗАДАЧА 1

**Есть код:**

```
var str = " 5";  
var num = str;  
console.log(typeof num);
```

**не менее двумя способами сделать так, чтобы num стал типа Number**

# ЗАДАЧА 2

Есть JS код:

```
var a = 20, b = 11;  
var c = a+++b*3-6/3;  
console.log(c);
```

Сейчас он выводит результат: 51

Используя только скобки, сделать так, чтобы c = 30

# ЗАДАЧА 3

Написать функцию, которая создаёт массив указанной длины, заполняет его случайными числами, которые находятся в диапазоне чисел, которые также указывает пользователь и возвращает нам этот массив:

```
function createArray(arrayLength, minNumber,  
maxNumber){  
    //ваш код  
    return arr;  
}
```

# ЗАДАЧА 4

Есть разметка

```
<input/>
```

Будем считать какую пенсию получит наш пользователь по формуле  $\text{Пенсия} = 4000 * \text{Коэффициент}$

Пользователь должен ввести числом свой рабочий стаж, а мы должны считать это значение по событию "blur", то есть при потере фокуса. Далее проверяем введено ли число, если это не число, то проверяем регулярным выражением, есть ли в ведённых данных число и пытаемся извлечь первое найденное число.

Если это точно не число, то выводим alert с предложением ввести корректные данные и очищаем значение `input.value`.

Если стаж больше 30 лет, то коэффициент устанавливаем как 0.8, для стажа от 20 до 30 (20 не включительно, а 30 включительно) коэффициент 0.6, для остального стажа 0.4

В конце выводим размер пенсии

## ЗАДАЧА 5

Создать массив дней недели с воскресенья до субботы (в виде массива строк). С помощью объекта `Date` и его методов проверить, какой сейчас день недели и через `switch` вывести информацию о том, сегодня будний день или выходной

# ЗАДАЧА 6

Есть такой код:

```
function User(name){
    this.name = name;
}
User.prototype.sleepDuration = 8;
User.prototype.present = function(){
    console.log("My name is " + this.name);
    console.log("I am a student of the " + this.university);
}
User.prototype.sleepAtHome = function(){
    console.log("I will sleep " + this.sleepDuration + " hours");
}
function Student(name, university){
    this.name = name;
    this.university = university;
}
Student.prototype.sleepInClass = function(){
    console.log("I will sleep 3 hours in the class");
}
```

Сделать так, чтобы при вызове объекта типа `Student` были доступны все поля и методы из типа `User`.

Дополнительно поместить вызовы функций объекта типа `Student` внутрь блока `try/catch`, чтобы при невозможности вызова каких-то из методов, программа не прерывалась, а обрабатывалась в `catch`