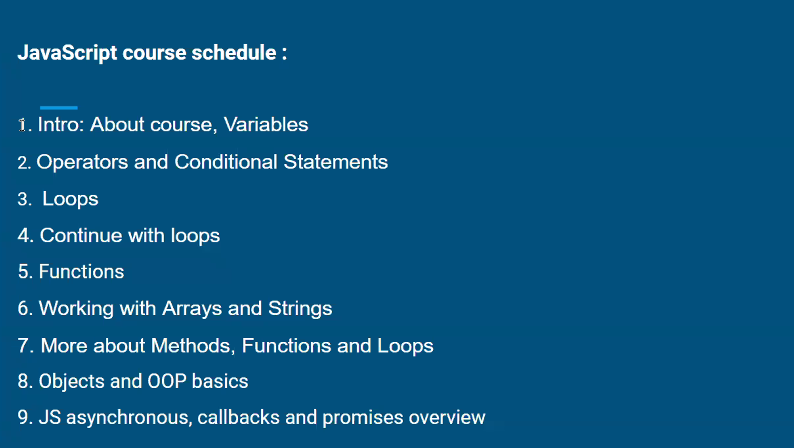
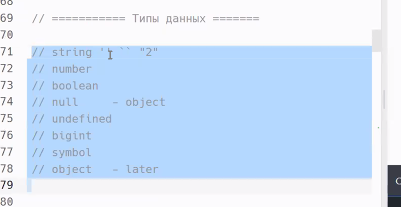
# Занятие 1 - Переменные и их типы

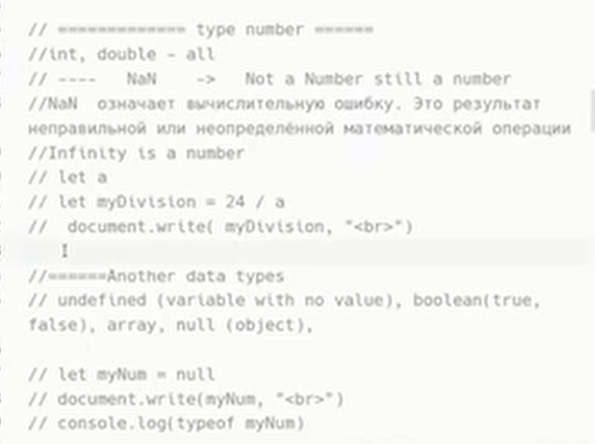


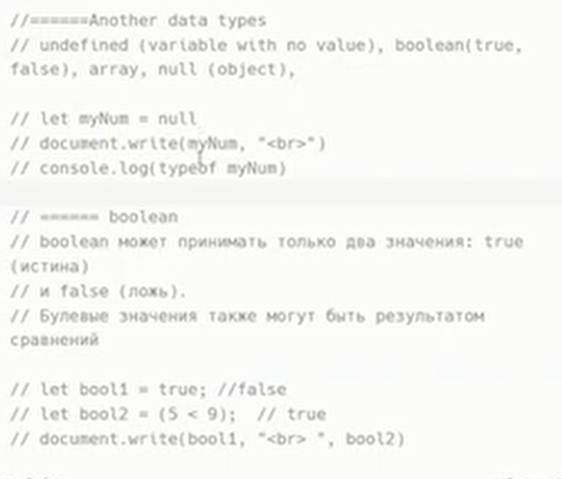


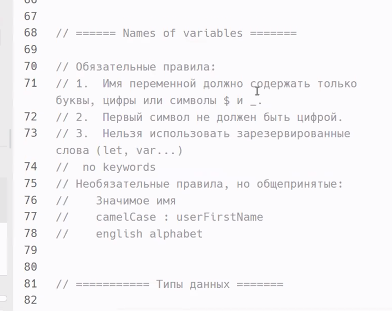


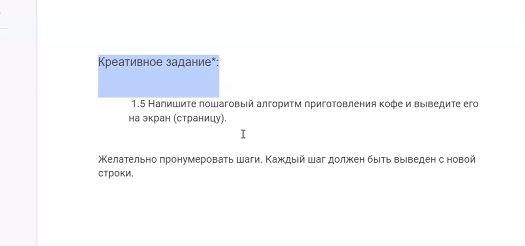
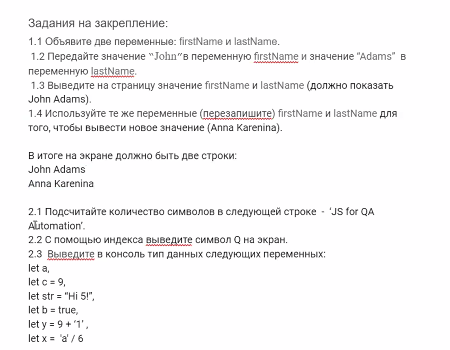
## Типы переменных

* Строка
* Число Integer(целые), float (число с точкой)
* Булин значения true/false
* Null – когда говорим что там что то есть но еше не решили что
* Не присвоили нечего
* Большие данные
* Символ
* Объект



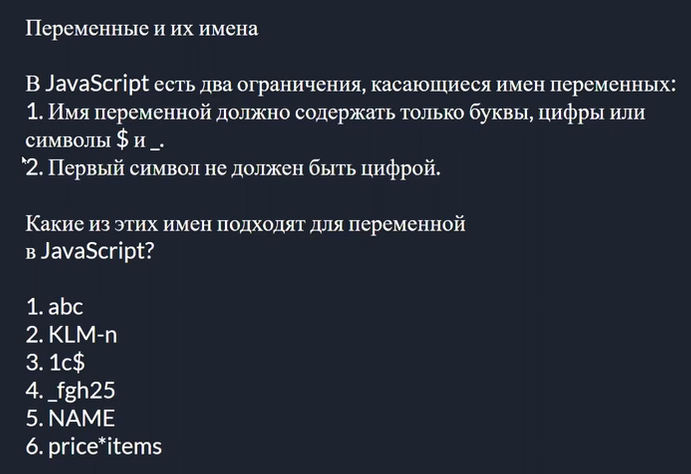


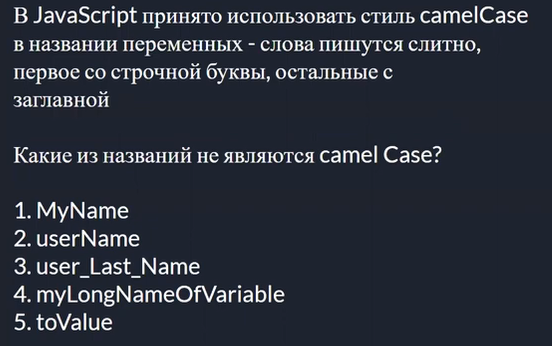


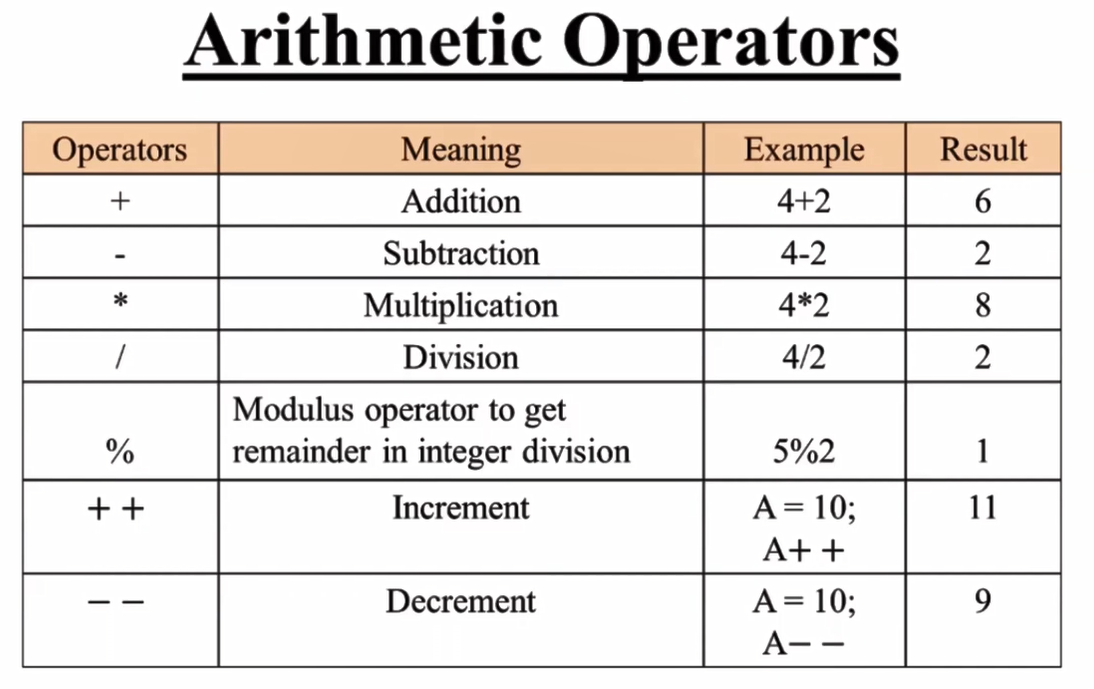


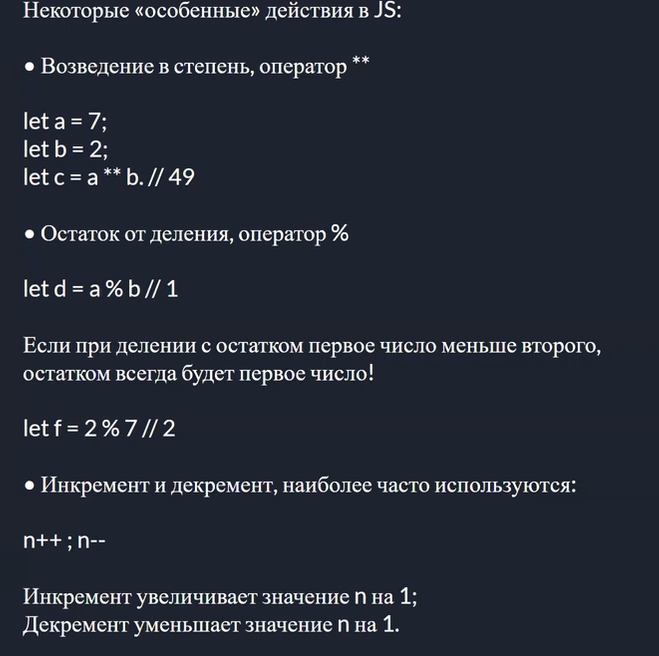
## Review 2 – 1 занятия

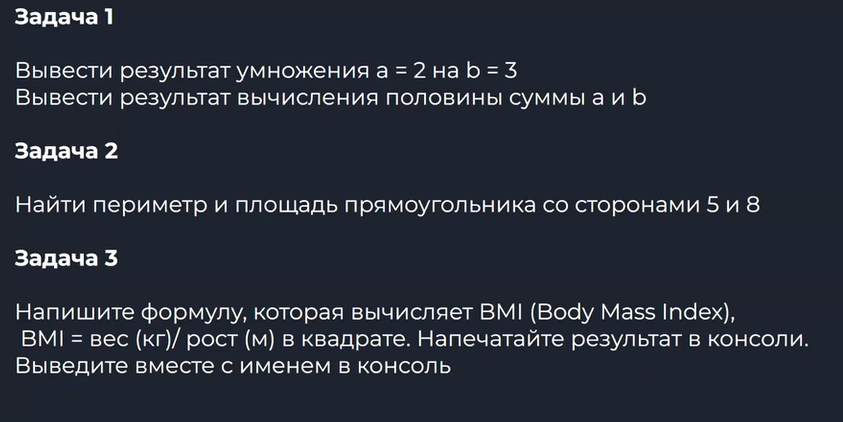
<https://www.youtube.com/watch?v=AYfPa-DvgDQ&list=PL6Gtav8N4O7iYmUm6wBjJd8EJ2KFXMPIH&index=3>









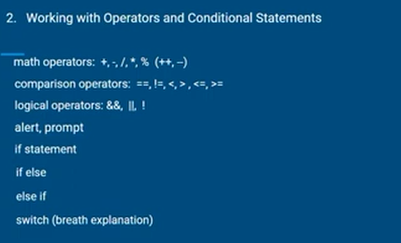


Задача 4

Поменять значения переменных X и Y

# Занятие 2 - Операторы и условные выражения(if, else, switch)

## Содержание



## Термины: «унарный», «бинарный», «операнд»

### Операнд

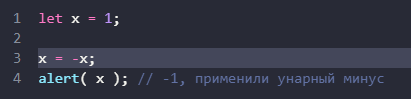
**Операнд** – то, к чему применяется оператор.

Например, в умножении 5 \* 2 есть два операнда: левый операнд равен 5, а правый операнд равен 2. Иногда их называют «аргументами» вместо «операндов».

### Унарный

Унарным называется оператор, который применяется к одному операнду.

Например, оператор унарный минус "-" меняет знак числа на противоположный:



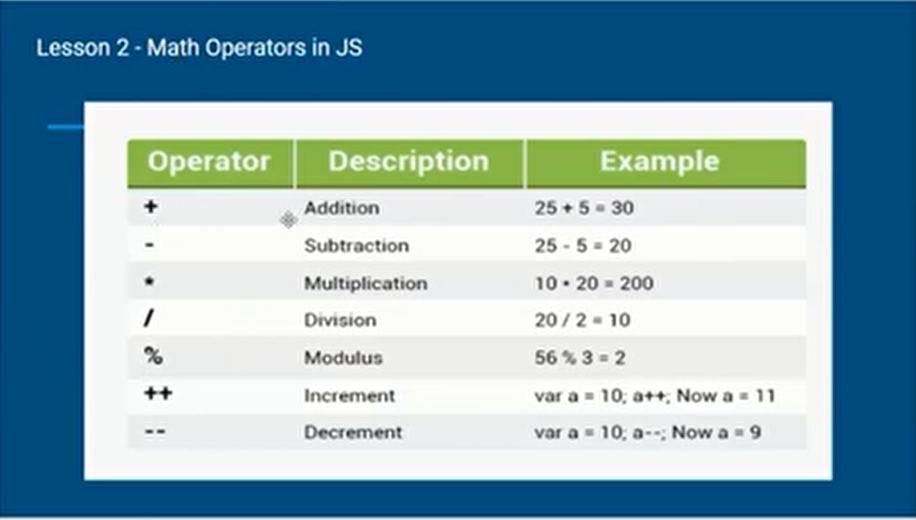
### Бинарный

**Бинарным** называется оператор, который применяется к двум операндам.

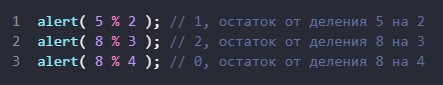
Тот же минус существует и в бинарной форме:



## Математические операторы

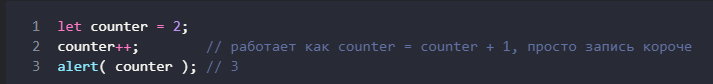


### %



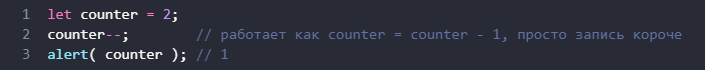
### ++ Инкремент

++ увеличивает переменную на 1:



### -- Декремент

-- уменьшает переменную на 1:



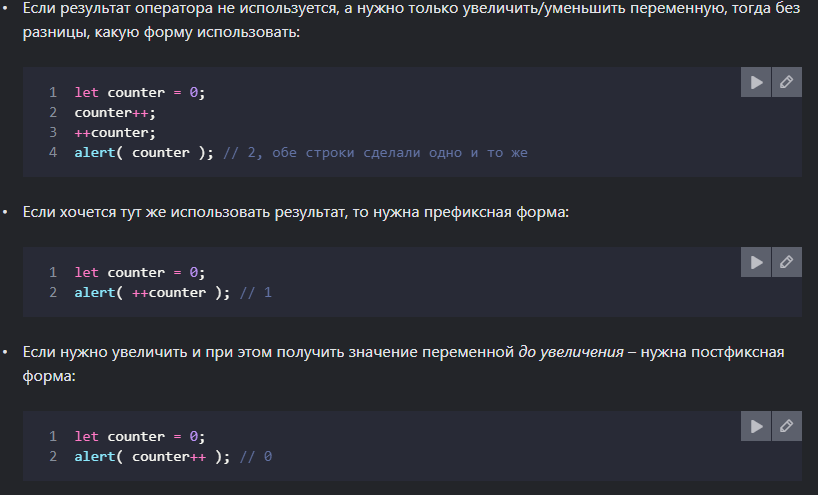
Операторы ++ и -- могут быть расположены не только после, но и **до переменной.**

Когда оператор идёт после переменной — это «**постфиксная форма**»: counter++.

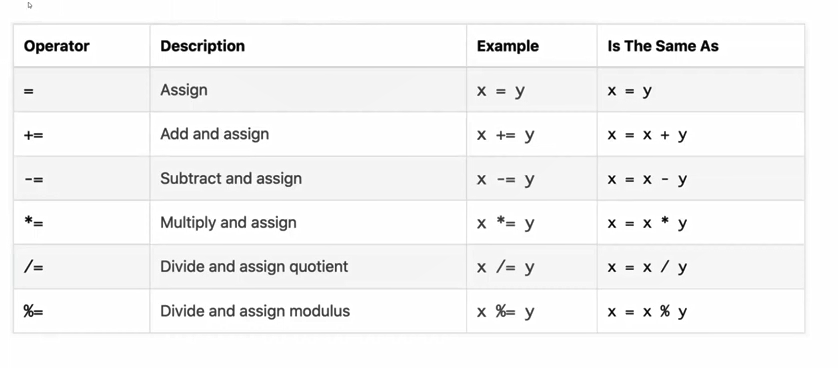
Префиксная форма возвращает новое значение.

«**Префиксная форма**» — это когда оператор идёт перед переменной: ++counter.

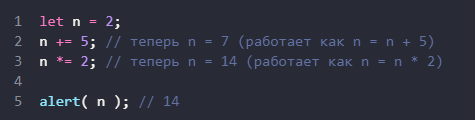
Постфиксная форма возвращает старое (до увеличения/уменьшения числа).



## Оператор присваивания

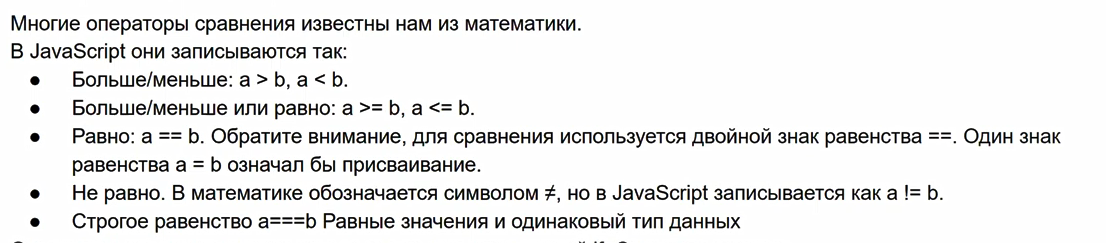


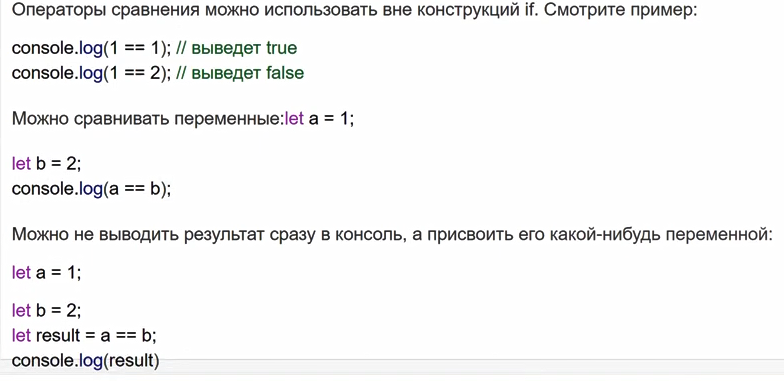
### += и -=



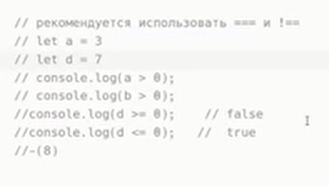
Подобные краткие формы записи существуют для всех арифметических и побитовых операторов: /=, -= и так далее.

## Операторы сравнения

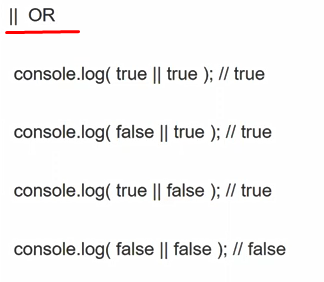
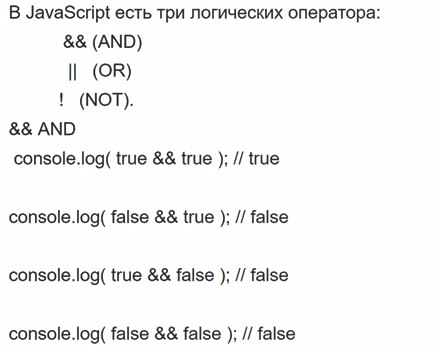


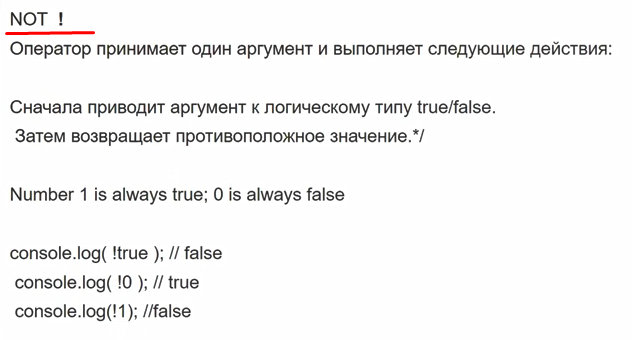


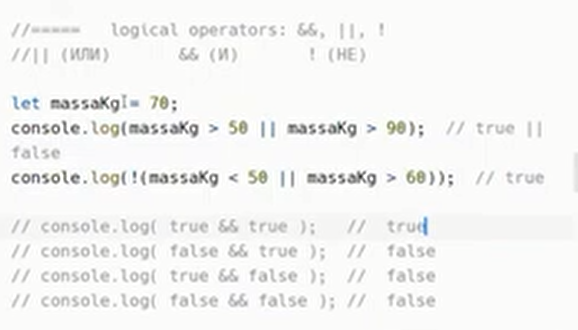




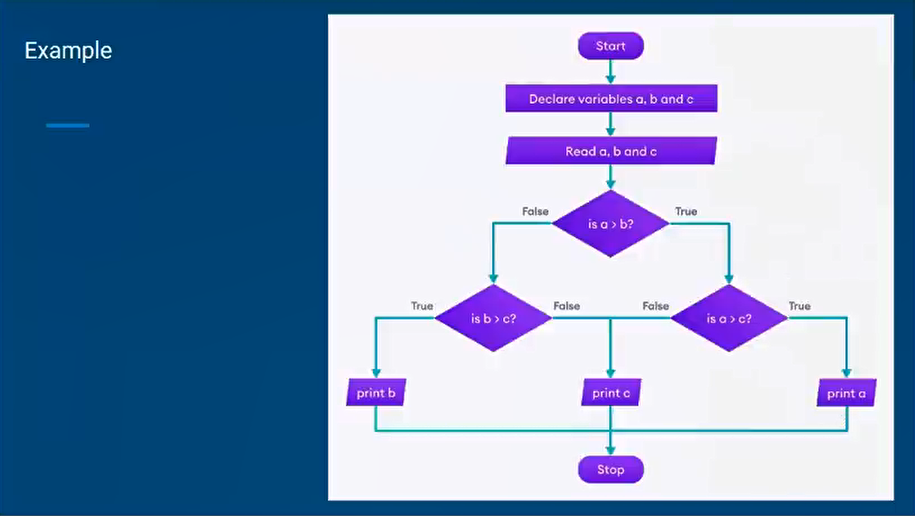
## Логические операторы сравнения

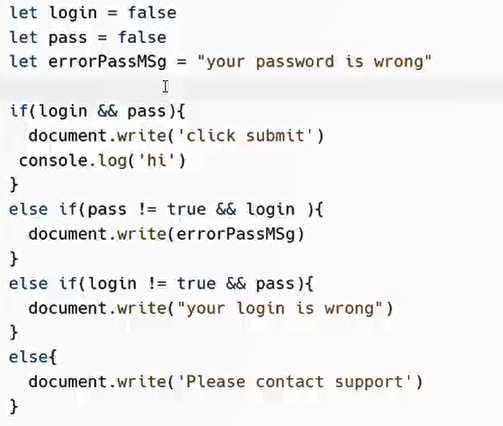




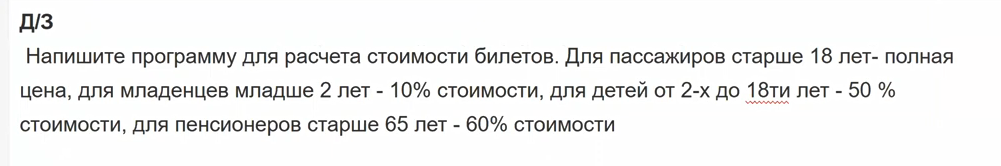






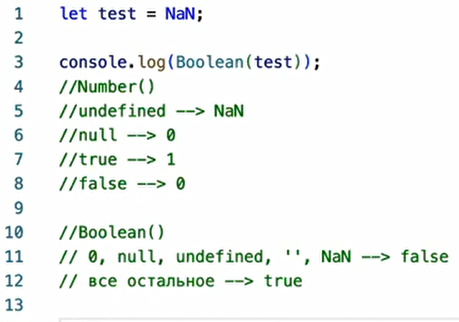






## 2.3 (03/04) Типы данных, взаимодействие с модальными окнами, преобразование типов (Stan P.)

<https://www.youtube.com/watch?v=k81_bhElY8M&list=PL6Gtav8N4O7iYmUm6wBjJd8EJ2KFXMPIH&index=9>



Логика работы логических операторов



1. По приоретету сначала сработает &&
2. Проверяет 3 на true/false 🡪 3
3. Проверяет 4 на true/false🡪 4
4. Останавливается на 4
5. Потом по приоритету || начинаем слева на право
6. null || 4 , null 🡪 false
7. переходим на 4 🡪 true
8. На этом обработка останавливается и результатом будет 4

## Материал к прочтению

|  |  |
| --- | --- |
| <https://learn.javascript.ru/operators>  <https://learn.javascript.ru/comparison>  <https://learn.javascript.ru/ifelse>  <https://learn.javascript.ru/logical-operators> | <https://www.w3schools.com/js/js_operators.asp>  <https://www.w3schools.com/js/js_arithmetic.asp>  <https://www.w3schools.com/js/js_if_else.asp>\_\_ |