**Что такое конкатенация JS?**

Конкатенация – это процесс соединения вещей друг с другом. В JavaScript конкатенация чаще всего используется для соединения значений переменных друг с другом, или строк со строками (для образования более длинных строк).

**var title = “We wish” + “to be friends”;**

**console.log(title);**

*We wish to be friends*

**Какими бывают комментарии и как они записываются.**

Как мы знаем из главы [Структура кода](https://learn.javascript.ru/structure), комментарии могут быть однострочными, начинающимися с //, и многострочными: /\* ... \*/.

Обычно мы их используем, чтобы описать, как и почему работает код.

**Различия записи переменных в ES5 и ES6.**

**Как допустимо записывать имена переменных в JS?**

**Правило 1.**

**Никакого транслита. Только английский.**

**Правило 2.**

**Использовать короткие имена только для переменных «местного значения».**

Называть переменные именами, не несущими смысловой нагрузки, например a, e, p, mg – можно только в том случае, если они используются в небольшом фрагменте кода и их применение очевидно.

Вообще же, название переменной должно быть понятным. Иногда для этого нужно использовать несколько слов.

**Правило 3.**

**Переменные из нескольких слов пишутся вместеВотТак.**

Camel

**Правило последнее, главное.**

**Имя переменной должно максимально чётко соответствовать хранимым в ней данным.**

**Что такое переменная?**

[Переменная](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) – это «именованное хранилище» для данных. Мы можем использовать переменные для хранения товаров, посетителей и других данных.

Для создания переменной в JavaScript используйте ключевое слово let.

**Отличия переменных var, let, const их область видимости.**

const – похоже на let, но значение переменной не может изменяться.

**объявление 2х let невозможно**

**переназначение const невозможно**

**объявление 2х var возможно**

**var хойстится let не хойстится**

**Перечислите основные типы данных JS.**

**// var str = 'Jack';**

**// var num = 10;**

**// var bool = true;**

**// var und = undefined;**

**// var empty = null;**

**// var sm = Symbol();**

**// var obj = {**

**// str: 'Jack',**

**// num: 10**

**// };**

**Что делает оператор TypeOf?**

Оператор typeof x позволяет выяснить, какой тип находится в x, возвращая его в виде строки.

typeof true // "boolean"

**Напишите инкремент и декримент.**

**Инкремент** ++ увеличивает на 1:

**Декремент** -- уменьшает на 1:

Инкремент/декремент можно применить только к переменной. Попытка использовать его на значении, типа 5++, приведёт к ошибке.

<https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide/Expressions_and_Operators>

**Перечислите логические операторы JS?**

expr1 && expr2 оператор И

expr1 || expr2 оператор ИЛИ

!expr оператор НЕ

**В чем разница между "==" и "===" ?**

Дело в том, что, сравнивая значения через двойное равно, JavaScript пер-вым делом пытается преобразовать их к одному типу. В данном случае булево

значение он преобразует в числовое — при этом falseстановится нулем,

а true— единицей. Поэтому, сравнивая 0 == false, вы получите true

Чтобы проверить два числа на точное равенство, используйте тройной знак

равенства (===)— это оператор «равно». Будьте осторожны, не путайте ===

с одиночным знаком равенства (=), поскольку ===означает «равны ли эти

два числа?», а =означает «положить значение справа в переменную слева».

Иначе говоря, ===задает вопрос, а =присваивает переменной значение.