



JavaScript

Урок №8: Основы. Выражения. Конструкция if else

Автор Бахшиллов Бехзод

Ментор по Frontend в **PRO** UNITY



Выражения.

Выражением является любой валидный блок кода, который имеет ценность.

Оператор - это символ или ключевое слово, которое производит вычисление над некоторыми значениями. Эти значения называют *операндами*.

Оператор с одним операндом называется *унарным*, с двумя - *бинарным*, с тремя - *тернарным*.

Унарный: `-10, +'10'`

Бинарный: `1 - 4, year = 2018`

Тернарный: `10 > 9 ? true : false`




Приоритеты.

Арифметика: $*/\%$ > $+ -$

Логика: $\&\&$ > $\|\|$

Оператор запятая: нулевой приоритет

Арифметики > логика > присваивание > оператор запятая



Присвоение. Сокращения. Арифметические операторы.

Присвоение:

```
let x = 5;
```

```
let y = 5 + 5;
```

```
let z = x + y;
```

```
x = x + 10;
```

```
x += 10;
```

```
y = y * 15;
```

```
y *= 15;
```

```
let a = b = 5;
```

Арифметические операторы:

1. Инкремент: **x++**, **++x**

2. Декремент: **--x**, **x--**

3. Взятие остатка:

```
4 % 2 // 0
```

```
100 % 60 // 40
```

```
10 % 3 // 1
```



Операторы сравнения.

```
10 < 20 // true
10 > 20 // false
5 <= 10 // true
10 >= 8 // true
"text" == "text" // true
5 != 5 // false
12 === 12 // true
12 !== 12 // false
```

Нестрогое (проверяется только значение)

```
12 == '12' // true
0 == false // true
undefined == null // true
12 != '12' // false
```

Строгое (проверяется тип данных)

```
12 === '12' // false
0 === false // false
undefined === null // false
12 !== '12' // true
```



Логические операторы. ||, &&.

Оператор “Или” ||

```
let x = true || false; // true
let x = '' || 'string'; // "string"
let x = '' || 0 || 'string'; // "string"
```

```
let user = { name: "Denis", age: 29 };
let userName = user.name || "User";
```

Оператор “И” &&

```
let x = true && false; // false
let x = '' && 'string'; // ""
let x = 'string' && 0 && ''; // 0
```

На заметку, логические операторы такие как || и && производят булево преобразование под капотом, но при этом всегда возвращают оригинальное значение операндов, даже если они не являются булевыми.



Логические операторы. Итого.

1. В JavaScript есть логические значения **true** (истина) и **false** (ложь). Операторы сравнения возвращают их.
2. Строки сравниваются **побуквенно**.
3. Значения разных типов приводятся к **числу** при **сравнении**, за исключением **строгого** равенства **===** (**!==**).
4. Значения **null** и **undefined** равны **==** друг другу и не равны ничему другому.
5. Будьте осторожны при использовании сравнений типа **>** или **<** с переменными, которые иногда могут быть **null/undefined**. Сделать отдельную проверку для **null/undefined** – хорошая идея.



Задачи часть 1:

Чему равно а, почему?

```
let a = 0 || 'string';  
let a = 1 && 'string';  
let a = null || 25;  
let a = null && 25;  
let a = null || 0 || 35;  
let a = null && 0 && 35;
```

Что отобразится в консоли. Почему?

```
12 + 14 + '12'  
3 + 2 - '1'  
'3' + 2 - 1  
true + 2  
+'10' + 1  
undefined + 2  
null + 5  
true + undefined
```


Условные операторы. if else. Ветвление кода.





Условные операторы. if else. Синтаксис.

```
if (условие) {  
    инструкции которые выполняются если условие правдиво  
}
```

```
if (условие) {  
    инструкции которые выполняются если условие правдиво  
} else {  
    инструкции которые выполняются если условие не правдиво  
}
```

```
if (условие) {  
    инструкции которые выполняются если условие правдиво  
} else if (условие) {  
    инструкции которые выполняются если условие правдиво  
} else {  
    инструкции которые выполняются если условие не правдиво  
}
```



Условные операторы. if else. Примеры.

```
let age = 29;
```

```
if (age >= 18) {  
    console.log("Вход разрешен");  
} else {  
    console.log("Вход запрещен");  
}
```

```
let user = { name: "Denis", type: "admin" };
```

```
if (user.type === "user") {  
    console.log("Добро пожаловать пользователь");  
} else if (user.type === "admin" ) {  
    console.log("Добро пожаловать администратор");  
} else {  
    console.log("Вход воспрещен");  
}
```

Задачи часть 2:

1. Если переменная равна "hidden", присвоить ей значение "visible", иначе - "hidden".
2. Используя if, записать условие:
 - а. если переменная равна нулю, присвоить ей 1;
 - б. если меньше нуля - строку "less then zero";
 - с. если больше нуля - используя оператор "присвоение", переменную умножить на 10 (использовать краткую запись).

3. Дан объект `let car = { name: 'Lexus', age: 10, create: 2008, needRepair: false }`. Написать условие если возраст машины **больше 5 лет** то нужно вывести в консоль сообщение 'Need Repair' и свойство **needRepair** в объекте **car** изменить на **true**; иначе изменить на **false**.

4. Дан объект `let item = { name: 'Intel core i7', price: '100$', discount: '15%' }`. Написать условие если у **item** есть поле **discount** и там есть значение которое **не null** а также есть поле **price** значение которого также **не null** то в объекте **item** создать поле **priceWithDiscount** и записать туда цену с учетом скидки и вывести ее в консоль, обратите внимание что поля **discount** и **price** это строки и вам из них нужно получить числа чтобы выполнить расчет. иначе если поля **discount** нет то вывести просто поле **price** в консоль.



Задачи часть 3:

5. Дан следующий код:

```
let product = {  
  name: "Яблоко",  
  price: "10$"  
};
```

```
let min = 10; // минимальная цена  
let max = 20; // максимальная цена
```

Написать условие если **цена товара больше или равна минимальной цене и меньше или равна максимальной цене** то вывести в консоль название этого товара, иначе вывести в консоль что товаров не найдено.