

ЛОГИЧЕСКИЙ ТИП И ВЕТВЛЕНИЕ



КОНСТАНТИН ПОЛЯНСКИЙ



КОНСТАНТИН ПОЛЯНСКИЙ



kv.polyanskiy@gmail.com



[@kvpolyanskiy](https://t.me/kvpolyanskiy)



ПЛАН ЗАНЯТИЯ

1. Повторение – мать учения
2. Зачем нам условные конструкции?
3. Логический тип данных и с чем его едят
4. Простые и сложные условия. Булева алгебра
5. Логическое преобразование




ПОВТОРЕНИЕ – МАТЬ УЧЕНИЯ

ВСПОМНИМ ПРОШЛЫЙ МАТЕРИАЛ

```
1  let message;  
2  
3  message = 'Hello!';  
4  
5  message = 'World!';  
6  
7  alert(message); // ?
```

ЧТО МЫ УВИДИМ В РЕЗУЛЬТАТЕ?

Мы увидим на экране окно с сообщением `'World! '`, потому что в изначально объявленную переменную `message` мы сначала запишем значение `Hello!`, а после перезапишем его строкой `World!` и выведем на экран.



ЗАЧЕМ НАМ УСЛОВНЫЕ КОНСТРУКЦИИ?



ЗАЧЕМ НАМ УСЛОВНЫЕ КОНСТРУКЦИИ?

Код обычно выполняется линейно сверху вниз, но иногда мы хотим выполнять (или не выполнять) какой-то блок кода только если некоторое условие верно.

Допустим, нам нужно в интернет-магазине вывести количество товара, либо «товар закончился» в зависимости от наличия этого товара на складе.

Давайте посмотрим как это реализовать.

ИНТЕРНЕТ-МАГАЗИН

```
1  let itemsInStock = 18;
2
3  if (itemsInStock > 0) {
4      console.log(itemsInStock);
5  } else {
6      console.log('Товар закончился');
7  }
```

А ЧТО ЕСЛИ Я ХОЧУ БОЛЬШЕ УСЛОВИЙ?

Все просто — вы можете использовать не только `if-else`, но и `else if` (важно: эта конструкция работает только в связке с `if` и имеет смысл, если нужно учесть несколько разных вариантов с конкретными условиями).

```
1  let age = 18;
2
3  if (age >= 18) {
4      console.log('Доступ разрешен');
5  } else if (age > 21) {
6      console.log('Вам даже в штатах можно все!');
7  } else if (age > 60) {
8      console.log('Пенсия... хотя может быть и нет :)');
9  } else {
10     console.log('Маловат еще');
11 }
```

А ЕСЛИ ТОЛЬКО ОДНО?

Так тоже можно :) Конструкция `if` может легко существовать без `else` (и довольно часто это, кстати, делает).

```
1  let username = "Yoda";  
2  
3  if (username == "Yoda") {  
4      console.log('Приветствую вас, учитель');  
5  }  
6  
7  // в противном случае просто ничего не делаем
```

`if` – ЭТО КИРПИЧИК, ИЗ НИХ МОЖНО СКЛАДЫВАТЬ БАШНИ

Один `if` может идти за другим. Они никак друг с другом не связаны.

Задача: если у Буратино есть больше 5 монет, то он покупает билет в театр. Если у него остаются деньги, то он покупает бутерброд за 1 монету.

```
1  let balance = 7;
2
3  if (balance >= 5) {
4      balance = balance - 5; // баланс после покупки
5      console.log('Куплен билет');
6  } else {
7      console.log('Без театра');
8  }
9
10 if (balance > 0) {
11     balance = balance - 1;
12     console.log('Куплен бутерброд');
13 }
14
15 console.log("Осталось монет " + balance);
```

КАКОВ РЕЗУЛЬТАТ ЛОГИЧЕСКИХ ОПЕРАЦИЙ?

В качестве сравнения мы используем, например, `age balance >= 5`. А каков собственно результат этого сравнения?

ЛОГИЧЕСКИЙ ТИП ДАННЫХ И С ЧЕМ ЕГО ЕДЯТ

БУЛЕВ/БУЛЕВЫЙ (ОН ЖЕ ЛОГИЧЕСКИЙ) ТИП «BOOLEAN»

У него всего два значения: `true` (истина) и `false` (ложь).

Как правило, такой тип используется для хранения значения типа да/нет, например:

```
1 | 18 > 0; // true
2 | 18 < 0; // false
```

```
1 | let checked = true; // галочка поставлена
2 | checked = false;    // галочка снята
```

ОПЕРАТОРЫ СРАВНЕНИЯ

Большинство операторов сравнения знакомы нам еще из школьной программы:

- Больше/меньше: `a > b`, `a < b`.
- Больше/меньше или равно: `a >= b`, `a <= b`.
- Равно `a == b`. Для сравнения используется два символа равенства '='. Один символ `a = b` означал бы присваивание.
- «Не равно». В математике он пишется как `≠`, в JavaScript – знак равенства с восклицательным знаком перед ним `!=`.



ПРОСТЫЕ И СЛОЖНЫЕ УСЛОВИЯ. БУЛЕВА АЛГЕБРА

ПРОСТЫЕ И СЛОЖНЫЕ УСЛОВИЯ

То, как работают простые условия мы с вами разобрали. А вот сложные условия включают в себя еще и логические операторы.

Например, если человеку больше 18 лет и у него есть кот, то он совершеннолетний любитель котов. В коде это будет выглядеть примерно так:

```
1  let age = 20;  
2  let hasCat = true;  
3  
4  if (age > 18 && hasCat === true) {  
5      console.log('Мяу');  
6  }
```

А КАКИЕ ЕЩЕ БЫВАЮТ ЛОГИЧЕСКИЕ ОПЕРАТОРЫ?

Нам с вами достаточно знать о трех:

- И, он же AND, он же &&;
- ИЛИ, он же OR, он же ||;
- НЕ, он же NOT, он же !.

То, как работают эти операторы в связке —легко понять по т.н. таблицам истинности. Давайте посмотрим как это выглядит.

ЛОГИЧЕСКИЙ ОПЕРАТОР И (ОН ЖЕ AND, ОН ЖЕ &&)

	F	T
F	F	F
T	F	T

```
1  let hour = 20;  
2  let minute = 45;  
3  
4  if (hour === 20 && minute === 45) {  
5      alert('Спокойной ночи малыши');  
6  }
```

ЛОГИЧЕСКИЙ ОПЕРАТОР ИЛИ (ОН ЖЕ OR, ОН ЖЕ ||)

	F	T
F	F	T
T	T	T

```
1  let hour = 9;
2
3  if (hour < 8 || hour > 21) {
4      alert('Аптека до 8 или после 21 закрыта');
5  }
```

ЛОГИЧЕСКИЙ ОПЕРАТОР НЕ (ОН ЖЕ NOT, ОН ЖЕ !)

a	F	T
¬a	T	F

```
1  let iKnowJS = false;
2
3  if (!iKnowJS) {
4      alert('Жизнь скучна и неинтересна!');
5  }
```

ХОЧУ БОЛЕЕ РЕАЛИСТИЧНЫЙ ПРИМЕР!

Давайте попробуем вместе!

Скажем, вы собрались на вечеринку. Но вечеринка тематическая и только для своих. *Вас пустят только если вы в костюме совы или кота и если охранник на входе вас знает.*

- название костюма мы храним в переменной `costume` ;
- знает человека охранник или нет, храним в переменной `knownPerson` .

Сможете превратить это условие в код? Присылайте ваши варианты прямо в чат.

ВОТ КАК БЫ МЫ МОГЛИ ЭТО ЗАПИСАТЬ

```
1 let costume = 'Cat';
2 let knownPerson = true;
3
4 if ((costume === 'Cat' || costume === 'Owl') && knownPerson) {
5     console.log('Заходи!');
6 }
```




ЛОГИЧЕСКОЕ ПРЕОБРАЗОВАНИЕ

ЛОГИЧЕСКОЕ ПРЕОБРАЗОВАНИЕ

В JS мы можем получить булевый тип не только при операциях сравнения, но и в логическом контексте, таком как `if (value)`.

Все значения, которые интуитивно «пусты», становятся `false`. Их несколько:

- `0`;
- Пустая строка;
- `null`;
- `undefined`;
- `NaN`.

Остальное, в том числе и любые объекты – `true`.

ПРИМЕР

В данном примере мы выведем в консоль имя **Bob**, так как в логическом контексте другие типы данных неявным образом преобразуются в булево значение.

```
1  let name = 'Bob';  
2  if (name) {  
3      console.log(name);  
4  }
```

ПОЛНАЯ ТАБЛИЦА ПРЕОБРАЗОВАНИЙ

Значение	Преобразуется в...
undefined, null	false
Числа	Все true, кроме 0, NaN – false.
Строки	Все true, кроме пустой строки "" – false
Объекты	Всегда true

ДАВАЙТЕ ЕЩЕ НЕМНОГО ПОПРАКТИКУЕМСЯ!

Как правило, у нас у всех есть много друзей, знакомых и просто людей, которые нас окружают. В зависимости от того, знаете вы имя человека и как долго вы знакомы — выведете на экран сообщения, исходя из таких условий:

- если вы знакомы, но не помните имя человека, то выведете в консоль "Знакомый"
- если имя вам известно, но вы знакомы меньше года, то "Приятели"
- если же вы знакомы более пяти лет, то "Друзья"
- во всех остальных случаях пусть будет "Не определились" :)

ИТАК, ПОСМОТРИМ, ЧТО ПОЛУЧИЛОСЬ

```
1  let name, yearsOfRelationship;
2  name = 'Bob';
3  yearsOfRelationship = 10;
4
5  if (name === "") {
6      console.log("Знакомый");
7  } else if (name && yearsOfRelationship < 1) {
8      console.log("Приятели");
9  } else if (name && yearsOfRelationship > 5) {
10     console.log("Друзья");
11 } else {
12     console.log("Не определились");
13 }
```

ЧЕМУ МЫ НАУЧИЛИСЬ?

1. Ветвления — основа основ программирования;
2. Булева алгебра, логические операторы, простые и сложные условия — как предусмотреть все нюансы;
3. Как все это использовать в JS.

РАДУЕМСЯ!



ДОМАШНЕЕ ЗАДАНИЕ

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задаем в группе Facebook!
- Задачи можно сдавать по частям.
- Зачет по домашней работе проставляется после того, как приняты все **3 задачи**.



ЧТО ПОЧИТАТЬ

- https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/Логические_операторы
- <https://learn.javascript.ru/logical-ops>
- <https://learn.javascript.ru/comparison>

А ПОСМОТРЕТЬ?

- <https://www.youtube.com/watch?v=2cGleL5FMZM> — хороший обзор того, на что мы сегодня посмотрели;
- <https://www.youtube.com/watch?v=B4ZCFdrBmbE> — еще один обзор с конкретными примерами, но уже на английском;
- <https://www.youtube.com/watch?v=Gy87ZS5sb1w> — параллельно даются несколько полезных советов для работы в консоле веб-разработчика.



Спасибо за внимание! Время задавать вопросы

КОНСТАНТИН ПОЛЯНСКИЙ



kv.polyanskiy@gmail.com



[@kvpolyanskiy](https://www.instagram.com/kvpolyanskiy)