

# Условный оператор if

**№ урока:** 6    **Курс:** Условный оператор if

**Средства обучения:** Visual Studio Code  
Web Browser

## Обзор, цель и назначение урока

Научиться использовать условный оператор if, обеспечивая выполнения ветки алгоритма только при условии истинности указанного выражения. Изучить особенности конвертации типа данных Boolean и разобраться с логическими операциями – конъюнкцией, дизъюнкцией, отрицанием.

## Изучив материал данного занятия, учащийся сможет:

- Использовать оператор if/else.
- Конвертировать разные значение в тип Boolean и понимать особенности такой конвертации.
- Работать с логическими операциями – И, ИЛИ, НЕ.

## Содержание урока

1. Использование оператора if else
2. Преобразование к булевому типу
3. Логические операции – конъюнкция, дизъюнкция, отрицание

## Резюме

- **Условные конструкции** или **операторы ветвления** - конструкция языка программирования, обеспечивающая выполнение определённой команды (набора команд) только при условии истинности некоторого логического выражения. В данном уроке рассматривается оператор if.
- Синтаксис:  
  
if(выражение)  
    Строка 1  
  
Если выражение true – выполняется «строка 1», иначе – «строка 1» не выполнится.
- Если при истинности выражения необходимо выполнить несколько строк кода, эти строки кода нужно взять в операторные скобки { и }.
- После оператора if можно использовать необязательный оператор else. Оператор else применяется в том случае, если необходимо задать альтернативный алгоритм, выполняющийся при ложном значении в выражении, заданном для if.
- Синтаксис:

```
if(выражение)
    Строка 1
else
    Строка 2
```

Если выражение true – выполняется «строка 1», иначе – выполняется «строка 2».

- **Логические операции** - операции над выражениями, позволяющие получить сложное выражение на основании более простых.
- **Конъюнкция** - и (&&). Результат логической операции true, если оба аргумента true, иначе – false.
- **Дизъюнкция** - или (||). Результат логической операции true, если хотя бы один из аргументов равен true, иначе – false.
- **Отрицание** - не (!). Принимает один аргумент и возвращает противоположное значение.
- Пример использования логических операций для сокращения количества условий:

```
if(inputLogin == login && inputPassword == password)
{
    Инструкции
}
```

Этот же код, но без конъюнкции:

```
if(inputLogin == login)
{
    if(inputPassword == password)
    {
        Инструкции
    }
}
```

- Если в выражении условия будет находиться значение, которое не является типом Boolean, то конвертация значение в логический тип произойдет автоматически. При этом:  
Как значение false будут рассматриваться значение 0, пустая строка "", null, undefined и NaN.  
Как значение true - все остальные значения.

### Закрепление материала

- Что такое условная конструкция?
- Что нужно сделать, чтобы выполнилось несколько инструкций, в случае если условие для if - истина.
- Что такое дизъюнкция, как она используется?

- Что такое конъюнкция, как она используется?
- Что такое отрицание, как оно используется?
- Если передать в условие значение не логического типа, как поведет себя условная конструкция?

### Дополнительное задание

#### Задание

Создайте сценарий. Запросите у пользователя его возраст. Если возраст пользователя больше 18 – выведите на экран сообщение «доступ разрешен». Если введенное пользователем значение меньше 18 - выведите на экран сообщение «доступ запрещен»

### Самостоятельная деятельность учащегося

Выполните задания в директории Exercises\Tasks\06 If Statement в материалах к этому уроку.

### Рекомендуемые ресурсы

If... else

<https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Statements/if...else>

Операторы сравнения

[https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/Операторы\\_сравнения](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/Операторы_сравнения)

Логические операторы

[https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/Логические\\_операторы](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/Логические_операторы)