

# Булевый (логический) тип

Булевый тип (`boolean`) может принимать только два значения: `true` (истина) и `false` (ложь).

Такой тип, как правило, используется для хранения значений да/нет: `true` значит «да, правильно», а `false` значит «нет, не правильно».

```
let nameFieldChecked = true; // да, поле отмечено
```

```
let ageFieldChecked = false; // нет, поле не отмечено
```

Булевы значения также могут быть результатом сравнений:

```
let isGreater = 4 > 1;
```

```
alert( isGreater ); // true (результатом сравнения будет "да")
```

# Логические операторы и операторы сравнения

**Логические операторы** и **операторы сравнения** используются для проверки выражений и переменных на соответствие какому-либо условию.

Оператор	Описание	Выражение	Возвращает	Оператор
>	больше чем	5 > 8	false	>
<	меньше чем	3 < 8	true	<
>=	больше или равно	4 >= 8	false	>=
<=	меньше или равно	8 <= 8	true	<=

Оператор	Описание	Выражение	Возвращает	Оператор
==	равно	5 == 8	false	==
		"5" == 5	true	
		“5” == "5"	true	
===	равно по значению и типу	"5" === "5"	true	===
		5 === "5"	false	
!=	не равно	7 != 8	true	!=
!==	не равно ни по значению, ни по типу	7!==8	true	!==
		5 !== 5	false	
		5 !== "5"	true	

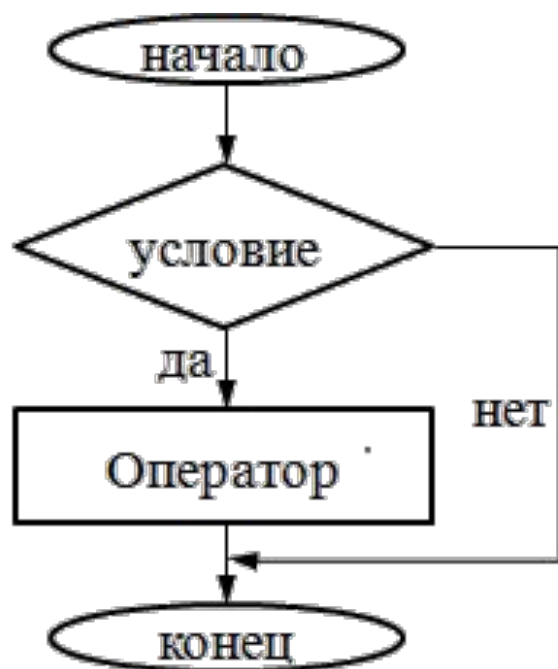
# оператор if

```
if (условное выражение)    действие
```

Примечание. Ключевое слово if должно быть написано строчными буквами. Использование заглавных букв (If или IF) приведет к возникновению ошибки . Заявление **if** состоит из:

- ключевое слово **if**;
- условие (выражения в скобках), которое должно быть истинным или ложным (или приводиться к одному из этих значений);
- инструкция, которая должна быть выполнена, если условие истинно или приведено к нему.

```
if (условное выражение)    {  
    действие1;  
    действие2;  
}
```



# Примеры

```
1)  if (x>0) y=sqrt(x);
```

```
2)  if (a > b) y= a-b;
```

```
3)  if (a !== b) {  
    alert( " значение a не равно b " );  
    a = b = 0;  
}
```

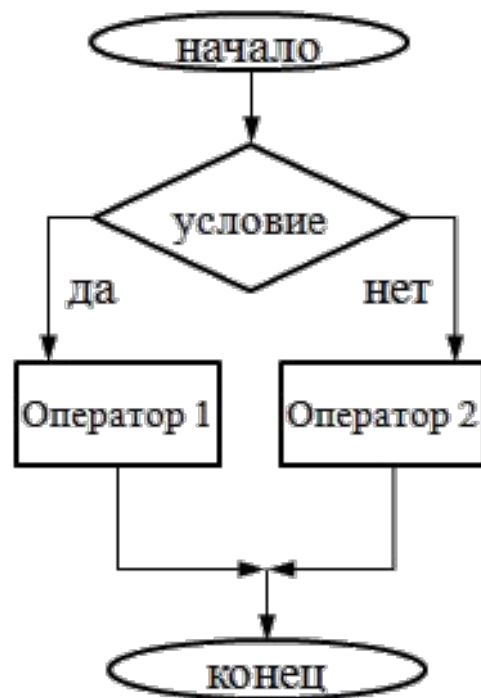
## Конструкция if...else (если...иначе)

Оператор **else** является частью и продолжением оператора **if**, используемого для определения блока кода JavaScript, который будет выполняться, если данное условие возвращает значение **false**.

```
if (условное выражение) {  
    // блок кода, выполняемый, если условие возвращает true  
} else {  
    // блок кода, выполняемый, если условие возвращает false  
}
```

Таким образом:

- Ключевое слово **if** ставится перед круглыми скобками.
- Условие находится внутри круглых скобок (например, «является ли это значение больше, чем другое значение?» или «существует ли это значение?»).
- Внутри фигурных скобок **{ }** находится код, который будет выполняться только в том случае, если условие истинно.
- Ключевое слово **else** означает «иначе».







# Проверка возраста

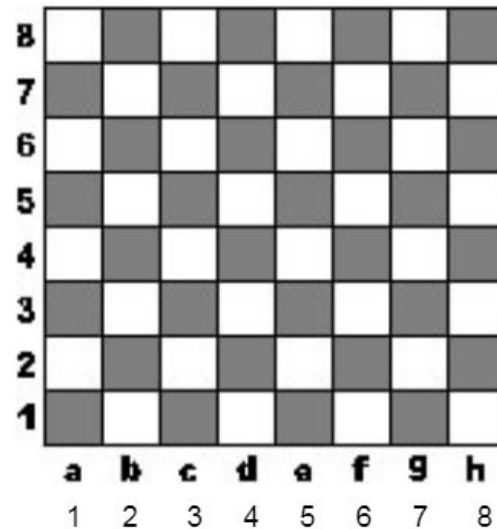
```
let age = +prompt("Введите ваш возраст");  
if (age < 18) alert("Доступ запрещен!")  
    else alert("Добро пожаловать!");
```

## Задача 2

Заданы две клетки шахматной доски.

- Если они покрашены в один цвет, то выведите слово YES,
- а если в разные цвета — то NO.

Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки.




11 и 13 - одинаковые цвета  
Сумма=1+1+1+3=6 - четное

# Задачи

- 1) Пользователь вводит 2 числа. Вывести на экран наибольшее из них.
- 2) Проверить что пользователь ввел положительное число,
- 3) Проверить что пользователь ввел четное число
- 4) Пользователь вводит 3 числа. Вывести на экран наибольшее из них.
- 5) Пользователь вводит 2 числа. Определить, эти числа с одинаковым знаком или нет.
- 6) Создать программу, которая определяет обычный год или високосный
- 7)
- 8) \*Проверка кратности числа. Вводятся два числа. Определить, кратно ли первое число второму, то есть делится ли первое число нацело на второе. Вывести на экран сообщение об этом, а также остаток от деления, если первое число не кратно второму.
- 9) \*Определить возможность существования треугольника по сторонам. Вводятся длины трех сторон предполагаемого треугольника. Определить, может ли существовать треугольник с такими сторонами при условии что, треугольник существует только тогда, когда ни одна его сторона не превышает сумму двух других.

## Преобразование условий в истинное или ложное

Если выражение в **if** операторе не является истинным или ложным, JavaScript изменит его на одно из этих значений. Он выполняет это действие, используя так называемое «правило лжи», т.е. любое выражение является истинным, кроме следующих значений:

- **false**;
  - **""** или **' '** - пустая строка;
  - **NaN** - специальный тип данных, "не число";
  - **0** - число «ноль»;
  - **null** - "пустое" значение;
  - **undefined** - "неопределенное" значение.
- 
- false**

Их иногда называют «ложными» значениями, поскольку код при этом условии никогда не будет выполнен.

```
if (0) { // ?
```

```
...
```

```
}
```

```
if ( ' ' ) { //?
```

```
...
```

```
}
```

```
if (null) { // ?
```

```
...
```

```
}
```

```
if (true) {  
    alert('Привет, мир!');  
}else{  
    alert('Пока, мир!');  
}
```

```
let num = 8;  
  
if (num<8) {  
    num += 5;  
}  
  
console.log(num) ; //?
```

```
let num = 8;  
  
if (num) {  
    num += 5;  
}  
  
console.log(num) ; //?
```



```
if(0) {
```

```
    num=0
```

```
    } else {
```

```
        num=1
```

```
    }
```

```
    alert(num)  //?
```

```
let num = 5;
```

```
if (num % 2) {
```

```
    console.log('Число _____! ');
```

```
} else {
```

```
    console.log('Число _____! ');
```

```
}
```

# Логические операторы

Основными логическими или булевыми операциями, названными в честь одного из математиков — Джорджа Буля (1815-1864), являются:

- ИЛИ || — логическое сложение (дизъюнкция) — OR;
- И && — логическое умножение (конъюнкция) — AND;
- НЕ ! — логическое отрицание (инверсия) — NOT.

# Логическое И (&& - - and) (конъюнкция - логическое умножение)

Он означает «и то, и другое». **Конъюнкция** – это сложное логическое выражение, которое считается истинным `true` только в том случае, когда оба его операнда (выражения) являются истинными `true`, во всех остальных случаях данное сложное выражение ложно `false`.

- Снег пойдет утром и днем.
- Одеть кофту и пальто
- Выбери яблоки и груши

x	y	$x \wedge y$ (x*y)
0	0	0
0	1	0
1	0	0
1	1	1

Давайте определим, в какую четверть часа попадает указанное количество минут:

let min = 13;



# Логическое ИЛИ (|| - - or ) (дизъюнкция - логическое сложение)

Он означает «или то, или другое, или оба». Если хотя бы один или оба операнда преобразуются в `true`, он возвращает `true` или непреобразованное значение. Если же оба операнда преобразуются в `false`, он возвращает `false` или непреобразованное значение.

- Снег пойдет утром или днем.
- Одеть кофту или пальто
- Выбери яблоки или груши

x	y	$x \vee y$ (x+y)
0	0	0
0	1	1
1	0	1
1	1	1

Написать вывод сообщения, работает офис или нет. Пользователь вводит время (в часах)

# Логическое отрицание НЕ (! - - not) (инверсия)

**Оператор логического НЕ** ! является унарным оператором, помещаемым перед одиночным операндом.

Он преобразует значение своего операнда в булево значение, затем инвертирует это значение в противоположное и возвращает, полученное в результате инвертирования, булево значение.

- Снег пойдет не утром
- Одеть не кофту
- Выбери не яблоки
- Неверно, что у меня есть яблоко
- Верно, что у меня нет яблоко
- Я не знаю китайского языка

x	$\overline{x}$
0	1
1	0



проверить  $x$  на ноль, используя инверсию

проверить наличие значения при заполнении окна `ptompt`

# if ..else if ...else

```
if (условное выражение) {  
    действие 1;  
} else if (условное выражение) {  
    действие 2;  
} else {  
    действие 3;  
}
```

# «Тернарный» оператор

**Тернарный оператор** – это оператор JavaScript, который возвращает результат первого или второго выражения в зависимости от истинности условия.

```
let result = условие ? значение1 : значение2;
```

*пример #1:*

```
let x=4;
```

```
let y = (x > 0) ? sqrt(x) : x**2;    // 2
```

*пример #2:*

```
let result = (1 + 5 < 4) ? 'Мало' : 'Много';    // много
```

Задания, в которых необходимо использовать тернарный оператор.

1. Запросить 2 числа и вывести большее из них.
2. Запросить 1 число и проверить, оно кратно 5 или нет.
3. Запросить у пользователя название планеты. Если пользователь ввел «Земля» или «земля», то вывести «Привет, землянин!», в остальных случаях вывести «Привет, инопланетянин!».

# Оператор switch

switch – это условный оператор, который сравнивает выражение сразу с несколькими возможными вариантами и выполнять один или несколько блоков кода на основе обнаруженных соответствий. Оператор switch по сути может заменить несколько блоков else if.

СИНТАКСИС

```
switch (expression) {  
  case x:  
    // execute case x code block  
    [break]  
  
  case y:  
    // execute case y code block  
    [break]  
  default:  
    // execute default code block  
}
```

В этом коде события будут происходить в такой последовательности:

- Сначала оценивается выражение.
- Оно сравнивается с первым условием (case x). В случае если выражение отвечает условию, оператор выполнит соответствующий код и оператор break прекращает выполнение блока switch.
- Если выражение не отвечает первому условию, оператор сравнит его с условием y. В случае если выражение отвечает условию y, оператор выполнит соответствующий код, а break прекращает выполнение блока switch.
- Если выражение не отвечает ни одному из условий, перечисленных в блоке switch, выполняется блок default.

пользователь вводит протоко, а выводится  
определение (HTTP,HTTPS,FTP)



Введите число от 0 - 2 и выведите его символьное написание

```
let arg = prompt("Введите число от 0 - 2?");
```

Запросить у пользователя номер месяца и вывести на экран его название.

приложение обрабатывает количество баллов, полученных за тест, и выставляет финальную оценку, где:

- 90 и выше – A.
- 80-89 – B.
- 70-79 – C.
- 60-69 – D.
- 59 и ниже – F.

Создать калькулятор для 4х математических операций. Пользователь вводит числа и знак операции +, -, \*  
Программа выводит результат, например:  $4*5=20$

```
number1 = +prompt("Введите 1 число:")  
number2 = +prompt("Введите 2 число:")  
operator = prompt("+, -, *, / \n Выберите математическую  
операцию: ")
```