

# HTML - Урок 12

Циклы и условные операторы



Геннадий Караченцев

[www.weblaboratory.in.ua](http://www.weblaboratory.in.ua)

# Как устроены программы?

Программа выполняет какое либо действие или комплекс действий и возвращает нам результат выполнения этих действий.

Каждое действие описано программистом, в виде выражений (строчек кода), так каждый шаг программы заведомо предусмотрен программистом.

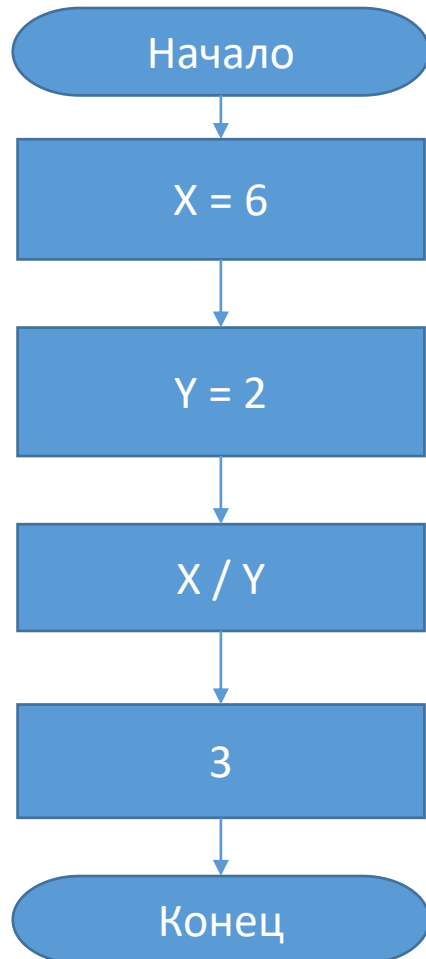
Программа не умеет думать, она умеет только выполнять указания программиста.

Но почему же программы такие умные и знают, что им делать в разных ситуациях?

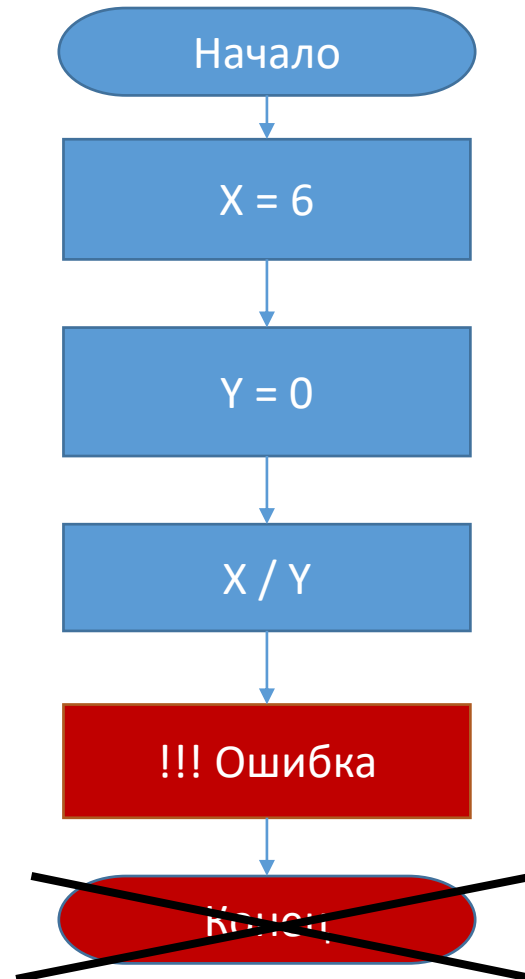
Все потому, что программист прописал им условия выполнения в зависимости от результата выполненного действия.

# Условия в программировании

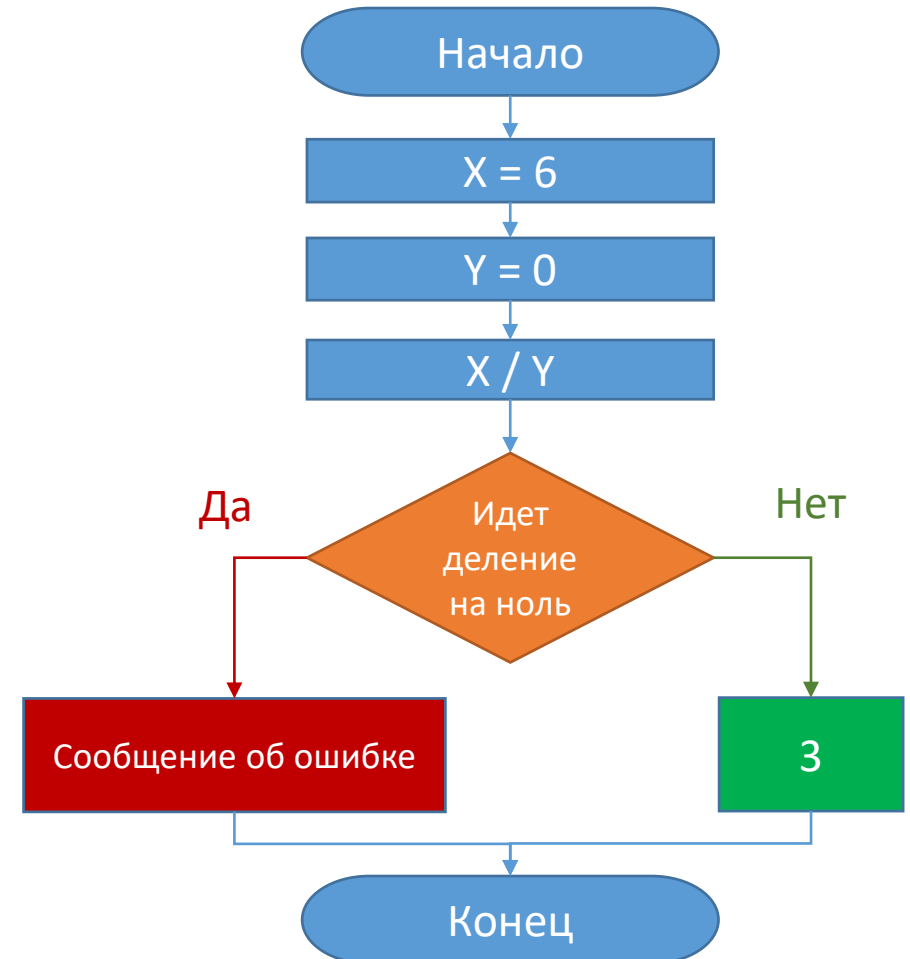
Программа 1



Программа 2



Программа 3



# Логические значения

Нужно понимать так, что операторы сравнения, такие же операторы как и умножения, сложения, вычитания и т.д.

Они также, возвращают значения, значения логического типа, «boolean».

Итак, мы помним, что булевы переменные имеют только 2 значения

- **true** – имеет смысл «да», «верно», «истина», 1.
- **false** – означает «нет», «неверно», «ложь», 0.

```
console.log( 6 > 5 );
```

# Логические операторы

Обычные логические операторы

**>** - Больше

**<** - Меньше

**==** - Равно

**!=** - Не равно

**>=** - Больше или равно

**<=** - Меньше или равно

Строгое равенство

**===** - равенство без преобразования типов

**!==** - не равенство без преобразования типов

# Сравнение строк

Компьютер не понимает алфавита, по этому у каждого символа есть совой номер. Для этого JavaScript использует кодировку [Unicode\(читать подробнее\)](#).

```
console.log( 'a' > 'я' ); // true
```

Если сравнивать строки, то сравниваются сначала правые символы, если код первого символа в строке больше чем во второй, то строка автоматически считается больше.

```
console.log( 'Банан' > 'Абрикос' ); // true
```

**Внимание!** Если сравнивать числа, которые пришли в виде строки, то их нужно сначала перевести в числовой тип данных. Такое бывает, когда получаем ответ от посетителя через **prompt**.

```
console.log( '2' > '14' ); // true
```

```
console.log( +'2' > +'14' ); // false
```

# Сравнение разных типов

Когда происходит сравнение разных типов, то сравниваются их числовые, преобразованные значения.

Так например **true** преобразуется в 1, а **false** в 0.

```
console.log( '2' > 1 ); // true, 2 преобразовалось в number
console.log( '01' == 1 ); // true, 01 преобразовалось в 1 – number
console.log( false == 0 ); // true, false становится числом 0
console.log( true == 1 ); // true, true становится числом 1
```

# Строгое равенство

Бывает так, когда нам необходимо отличить `false` от 0 или `true` от 1.

Сравнить не только логическое значение но и тип данных.

```
console.log( false == 0 ); // true
```

```
console.log( true == 1 ); // true
```

Для этого используется строгое равенство `===` или `!==`.

```
console.log( false === 0 ); // false
```

```
console.log( true === 1 ); // false
```



# Условные операторы: if

При помощи условного оператора **if** можно указать программе варианты выполнения того или иного кода, в зависимости от результата выполнения условия.

```
var age = 18;  
if ( age != 18 ) console.log( 'Вы несовершеннолетний.' );  
  
if ( age != 18 ) {  
    console.log( 'Вы несовершеннолетний.' );  
    console.log( 'Доступ закрыт.' );  
}
```

# Неверное условие, else

Что если мы хотим установить выполнение кода и на неверный результат?

Т.е. если верно выражение, то выполняем один код, а если нет, то выполняем другой.  
Для этого нам понадобится неверное условие **else**

```
var age = 18;
```

```
if ( age != 18 ) {  
    console.log( 'Вы несовершеннолетний.' );  
} else {  
    console.log( 'Все Ок, доступ разрешен!' );  
}
```

**ВАЖНО!!!** else не может существовать без if

```
else {  
    ... Код ...  
}
```

# Несколько условий, else if

А что если мы хотим больше, чем 2 варианта условий, верно и неверно?

В JS можно легко делать несколько сравнений, для этого нам понадобится условие **else if**.

```
var age = 18;
if ( age < 10 ) {
    console.log( 'Вам даже нет 10.' );
} else if ( age > 10 && age < 16 ) {
    console.log( 'Вам бы подрасти до 16.' );
} else if ( age < 18 ) {
    console.log( 'Вы несовершеннолетний!' );
} else {
    console.log( 'Добро пожаловать!' );
}
```

# Логические операторы в условии

Что если в одном условии, мы хотим задать несколько критериев для проверки.

Например, если число больше одного значения, но меньше другого.

Для этого используются логические операторы условия, **||** (или), **&&**(и) и **!** (НЕ).

```
var age = 18;  
if ( age > 10 || age < 18 ) // выполнится  
if ( age > 10 && age < 18 ) // не выполнится  
if ( ! (age < 18) ) // выполнится
```

# Взаимодействие с пользователем

**prompt** - позволяет вывести сообщение со строкой ввода и получить от пользователя ответ в виде строки.

**OK** - возвращает значение из строки, **CANCEL** - возвращает **null**.

```
console.log( prompt() );
```

**!Важно: цифровые значения полученные через prompt являются строками.**

```
console.log( prompt() + prompt() ); // string + string
```

# Программ проверки возраста и авторизации

