



Лекция 2

Циклы и программная логика

Цели на сегодня

- Поговорить об алгоритмах;
- Узнать о базовых способах взаимодействия с пользователем;
- Глянуть на циклы, понятие инкремента / декремента;
- Научиться использовать программную логику.

Алгоритм --

набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результата.

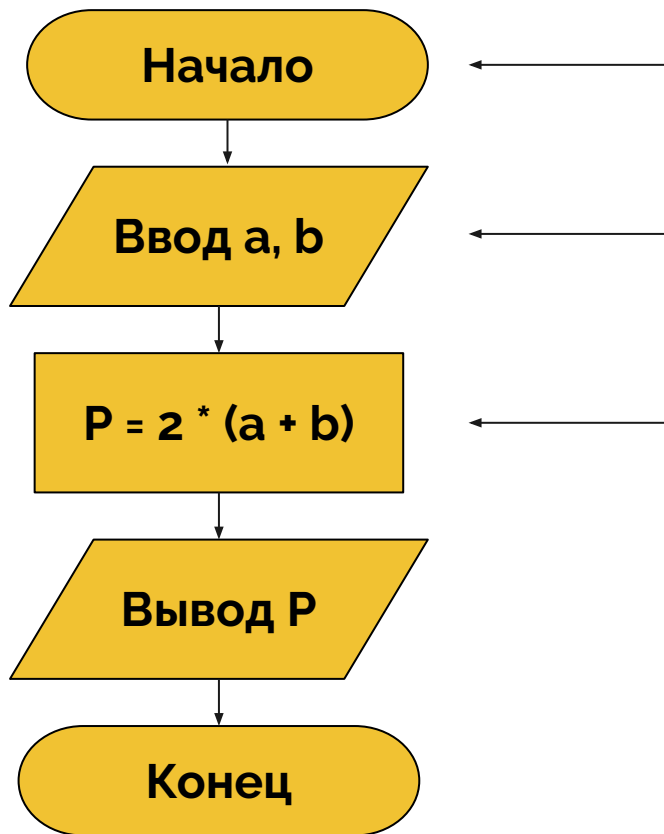
Примеры алгоритмов:

- синтез вещества;
- готовка по рецепту;
- сортировка предметов;
- перемещение по навигатору.

Основные виды описания алгоритмов:

- **блок-схемы** -- представление алгоритма в виде геометрических фигур;
- **псевдокод** -- использование основных конструкций JS для описания алгоритма, однако без подробностей.

Линейный алгоритм

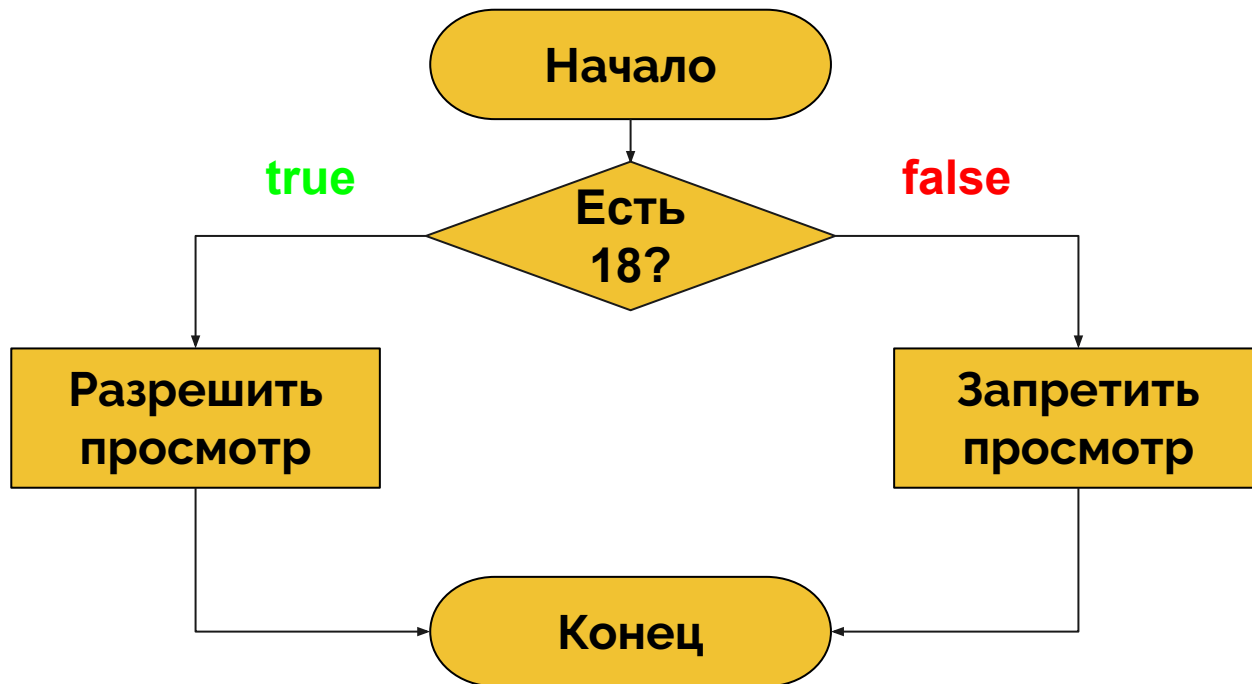


Обычно принято указывать
начало и конец программы

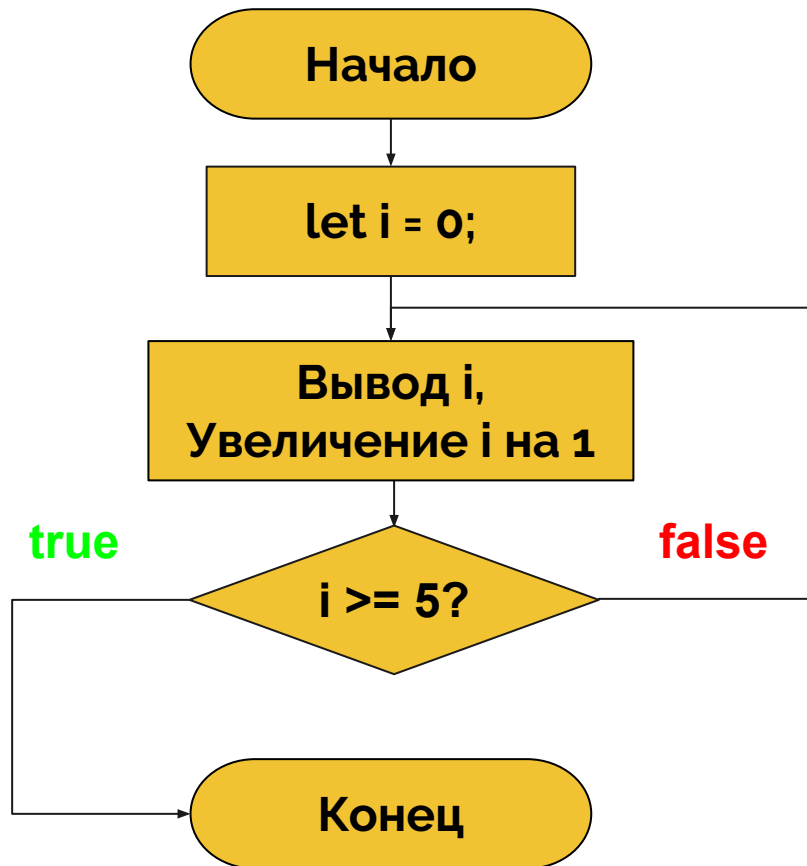
Параллелограмм указывает
на ввод и вывод данных

В прямоугольниках
записывают операции

Разветвленный (условный) алгоритм

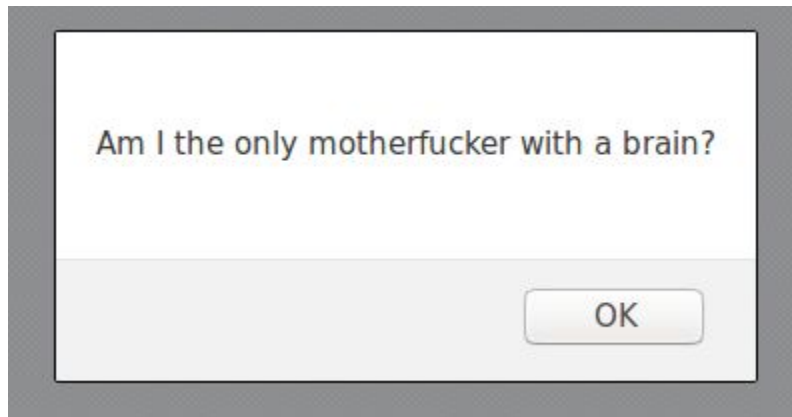


Циклический алгоритм

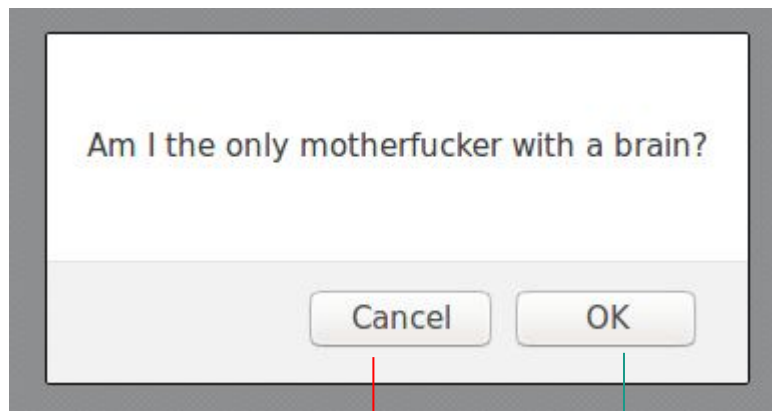


Взаимодействие с пользователем

```
alert('Am I the only motherfucker with a brain?');
```

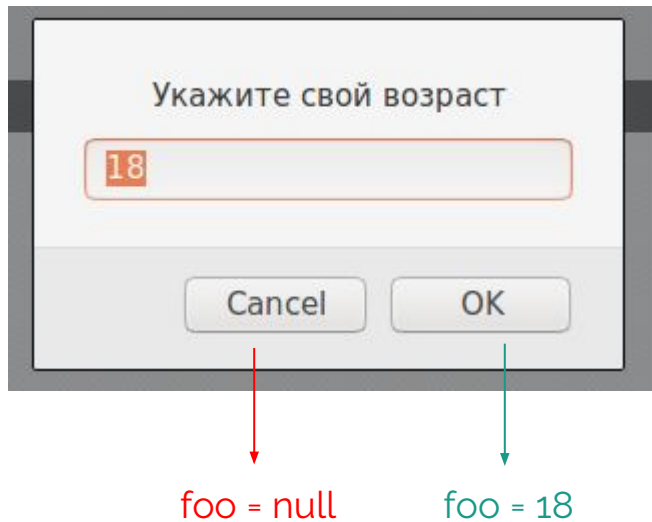


```
const question = confirm('Am I the only motherfucker  
with a brain?');
```



question = false question = true

```
const foo = prompt('Укажите свой возраст', '18');
```



Циклы (Loops)

Зачем нужны циклы?

При написании скриптов зачастую встает задача сделать однотипное действие много раз.

Например, вывести товары из списка один за другим, вывести записи блога. Или просто перебрать все числа от 1 до 10 и для каждого выполнить одинаковый код.

Для многократного повторения одного участка кода – предусмотрены циклы.

~ /Courses/ec-develop/loops.js - Sublime Text (UNREGISTERED)

loops.js x

```
1 // Инкремент -- используется для увеличения числа на 1
2
3 // (0) Постфиксная форма
4
5 let i = 0;
6 console.log(i++); // 0
7
8 // (1) Префиксная форма
9
10 let j = 0;
11 console.log(++j); // 1
```

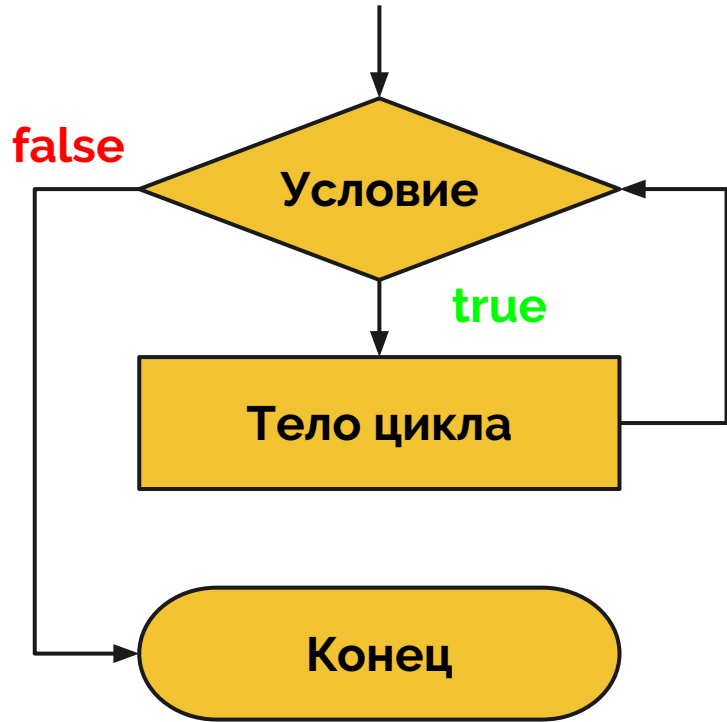
Line 11, Column 23 UTF-8 Tab Size: 4 JavaScript

~/Courses/ec-develop/loops.js - Sublime Text (UNREGISTERED)

loops.js x

```
1 // Декремент -- используется для уменьшения числа на 1
2
3 // (0) Постфиксная форма
4
5 let i = 10;
6 console.log(i--); // 10
7
8 // (1) Префиксная форма
9
10 let j = 10;
11 console.log(--j); // 9
```

Line 11, Column 23 UTF-8 Tab Size: 4 JavaScript



Первый рассматриваемый цикл -- **while**. Он проверяет условие, и если оно приравнивается к **true**, то тело цикла выполняется.

Повторение цикла по-научному называется «итерация». Цикл в примерах ниже совершает 10 итераций.

```
~/Courses/ec-develop/loops.js - Sublime Text (UNREGISTERED)
loops.js x
1 // Цикл while выглядит таким образом
2
3 while (условие) {
4     // код, тело цикла
5 }
6
7 let counter = 0;
8
9 while (counter <= 10) {
10     console.log(`The number is ${counter}`);
11     counter++;
12 }
```

Line 12, Column 2 UTF-8

```
The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9
The number is 10
```

← 10

```
~/Courses/ec-develop/loops.js - Sublime Text (UNREGISTERED)
loops.js x
1 // Итератор можно как увеличивать, так и уменьшать
2
3 let counter = 10;
4
5 while (counter >= 0) {
6     console.log(`The number is ${counter}`);
7     counter--;
8 }
```

Line 1, Column 51 UTF-8 Tab

```
The number is 10
The number is 9
The number is 8
The number is 7
The number is 6
The number is 5
The number is 4
The number is 3
The number is 2
The number is 1
The number is 0
```

← 0

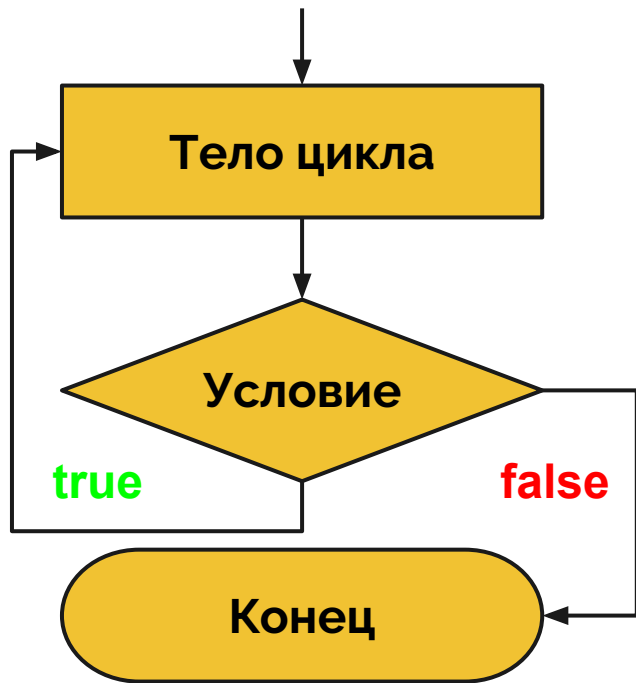
```
~/Courses/ec-develop/loops.js - Sublime Text (UNREGISTERED)
loops.js x
1  /*
2     Можно упростить условие.
3     Чем проще условие, тем быстрее будет выполняться
4     цикл.
5  */
6
7  let counter = 10;
8
9  while (counter--) {
10     console.log(`The number is ${counter}`);
11 }
12
```

Line 11, Column 2 UTF-8 Tab Size: 4

The number is 9
The number is 8
The number is 7
The number is 6
The number is 5
The number is 4
The number is 3
The number is 2
The number is 1
The number is 0

← undefined

Если можно двигаться к нулю, то лучше
двигаться к нулю.



В цикле **do...while** сначала выполняется тело, а уже потом проверяется соответствие условию.

Циклы **do...while** используются достаточно редко, поскольку считается, что они сложнее воспринимаются визуально.

```
loops.js x
1 // Следующим типом циклов является do...while
2
3 // Синтаксис:
4
5 do {
6     выражение
7 } while (условие);
8
9 // Пример:
10
11 let num = 0;
12
13 do {
14     console.log(`There are ${num} students here`);
15     num++;
16 } while (num <= 7);
```

There are 0 students here
There are 1 students here
There are 2 students here
There are 3 students here
There are 4 students here
There are 5 students here
There are 6 students here
There are 7 students here

← 7

```
~/Courses/ec-develop/loops.js - Sublime Text (UNREGISTERED)
loops.js x
1 // Чаще всего используется цикл for
2
3 // Синтаксис:
4
5 for (начало; условие; шаг) {
6     // ... тело цикла ...
7 }
8
9 // Пример:
10
11
12 for (let i = 0; i <= 10; ++i) {
13     console.log(i);
14 }
```

Line 12, Column 32 UTF-8

0

1

2

3

4

5

6

7

8

9

10

← undefined

~/Courses/ec-develop/loops.js - Sublime Text (UNREGISTERED)

loops.js

```
1 // В цикле for можно пропускать все части цикла:
2
3 let i = 0;
4
5 for (;i < 3; i++) {
6     console.log(i); // 0, 1, 2
7 }
```

Line 7, Column 2

UTF-8 Tab Size: 4 JavaScript

Бесконечный цикл (Infinite loop)

loops.js

```
1 // Бесконечный цикл, который что-то делает
2 let i = 0;
3
4 while (true) {
5     console.log(i);
6     i++;
7 }
8
9 // Бесконечный цикл, который ничего не делает
10
11 for(;;);
```

~/Courses/ec-develop/loops.js - Sublime Text (UNREGISTERED)

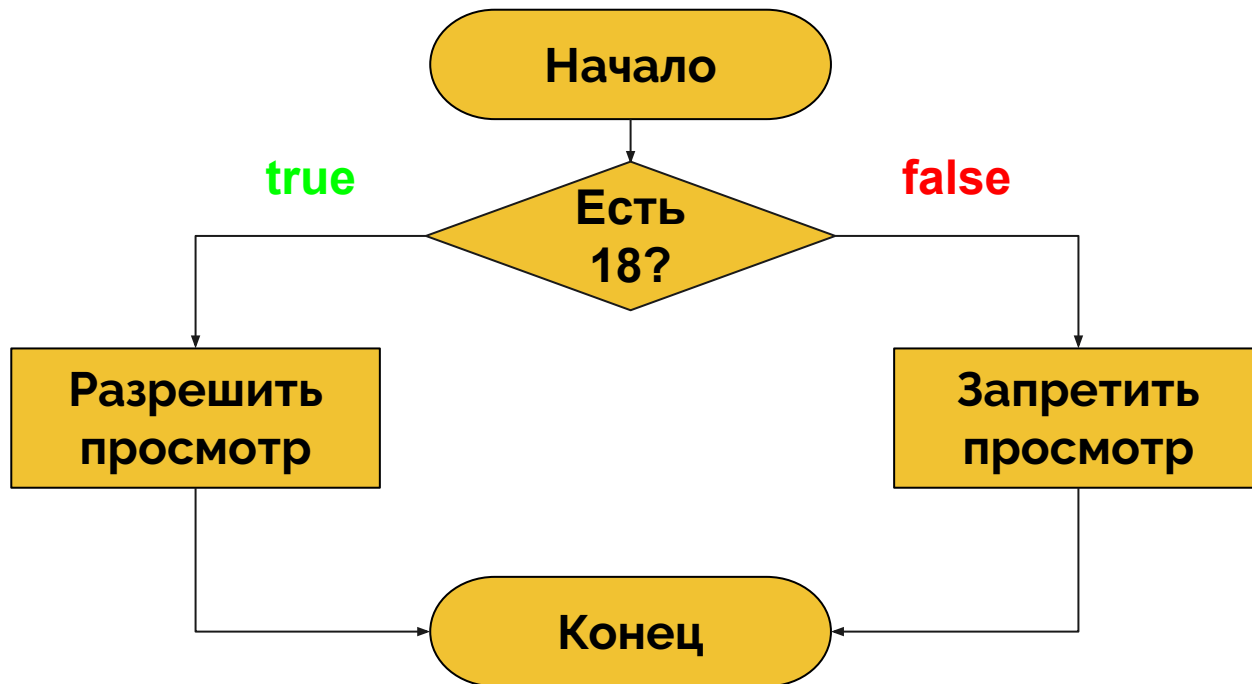
loops.js x

```
1 let sum = 0;
2
3 while (true) {
4
5     let value = +prompt("Введите число", '');
6
7     if (!value) break;
8
9     sum += value;
10
11 }
12
13 alert( 'Сумма: ' + sum );
```

Line 13, Column 26 UTF-8 Tab Size: 4 JavaScript

Условная логика

Разветвленный (условный) алгоритм



Зачем нужны условные операторы?

Если в зависимости от условия, необходимо выполнить различные действия, то используют условные операторы.

Число 0, пустая строка "", null и undefined, а также NaN являются false.

Остальные значения – true.

~/Courses/ec-develop/loops.js - Sublime Text (UNREGISTERED)

loops.js x

```
1  const year = prompt('Сколько Вам лет?', 16);
2
3  if (year === null) {
4      alert('Данные не получены');
5  } else if (year < 18) {
6      alert('Доступ запрещен');
7  } else if (year >= 18) {
8      alert('Доступ разрешен');
9  } else {
10     alert('Системная ошибка');
11 }
```

Line 11, Column 2

UTF-8 Tab Size: 4 JavaScript


```
1 const year = prompt('Сколько Вам лет?', 16);  
2  
3 // условие ? true : false  
4  
5 (year < 18) ? alert('Запрещено') : alert('Разрешено');
```

common.js

```
1  /*
2   Конструкция switch используется для перебора значений и
3   представляет собой более наглядный способ сравнить выражение
4   сразу с несколькими вариантами.
5   */
6
7  const city = 'Львов';
8
9  switch (city) {
10   case 'Киев':
11     console.log('Вы находитесь в Киеве');
12     break;
13   case 'Днепр':
14     console.log('Вы находитесь в Днепре');
15     break;
16   case 'Запорожье':
17     console.log('Вы находитесь в Запорожье');
18     break;
19   default:
20     console.log('Не удалось определить ваше местоположение');
21 }
```