

# Javascript

Базовые операторы  
Условное ветвление  
Циклы  
Конструкция switch

# Урок

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18

# Базовые операторы

Присваивание

Математические

Составное присваивание

Инкремент / Декремент

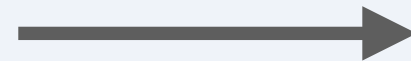
Равенство

Сравнение

Логические

# Присваивание

=



```
let name = 'Brad'
```

# Математические



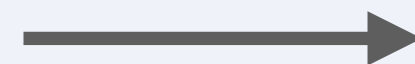
# Составное присваивание

`+=`

`-=`

`*=`

`/=`

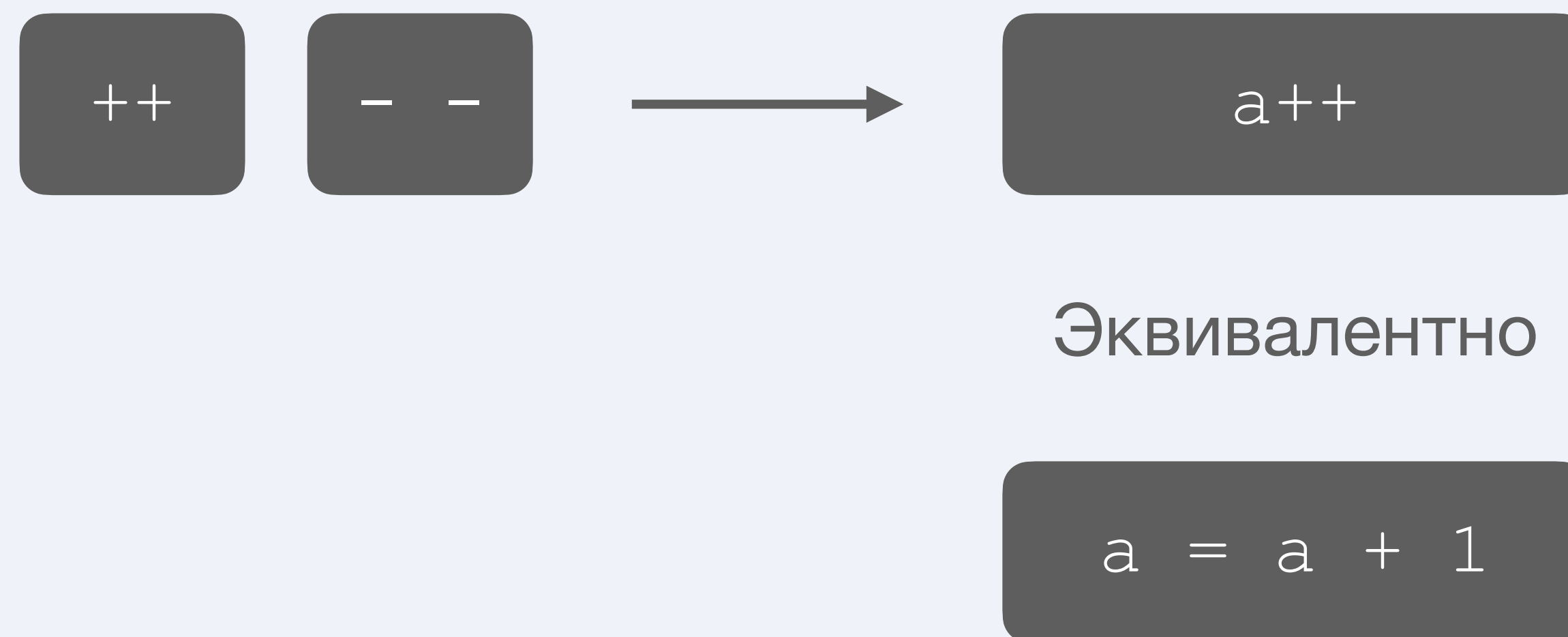


```
let a += b
```

Эквивалентно

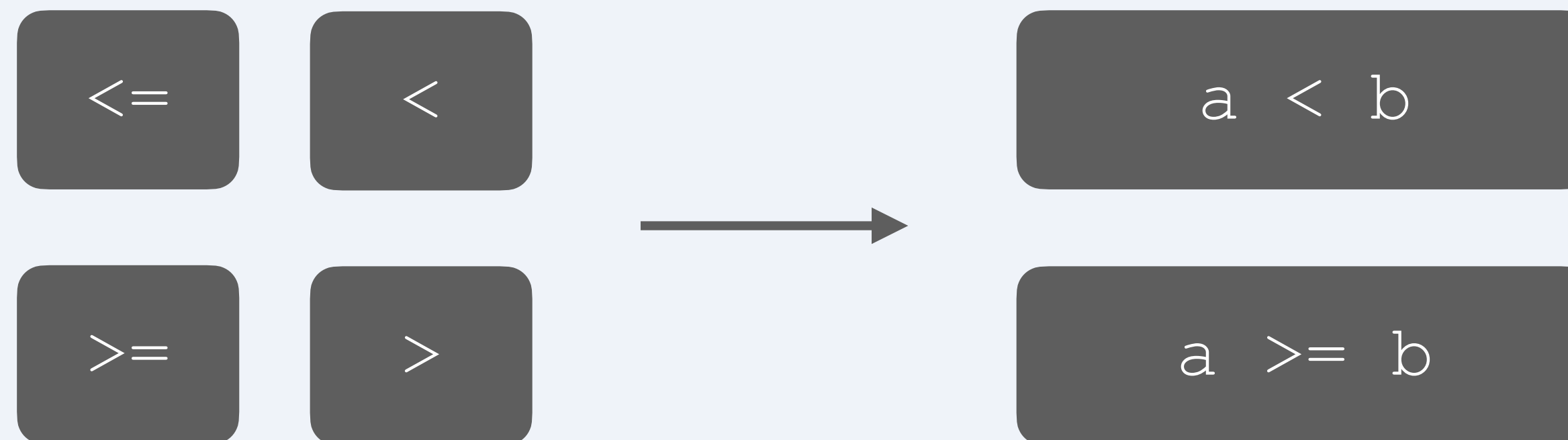
```
let a = a + b
```

# Инкремент / Декремент



# Сравнение

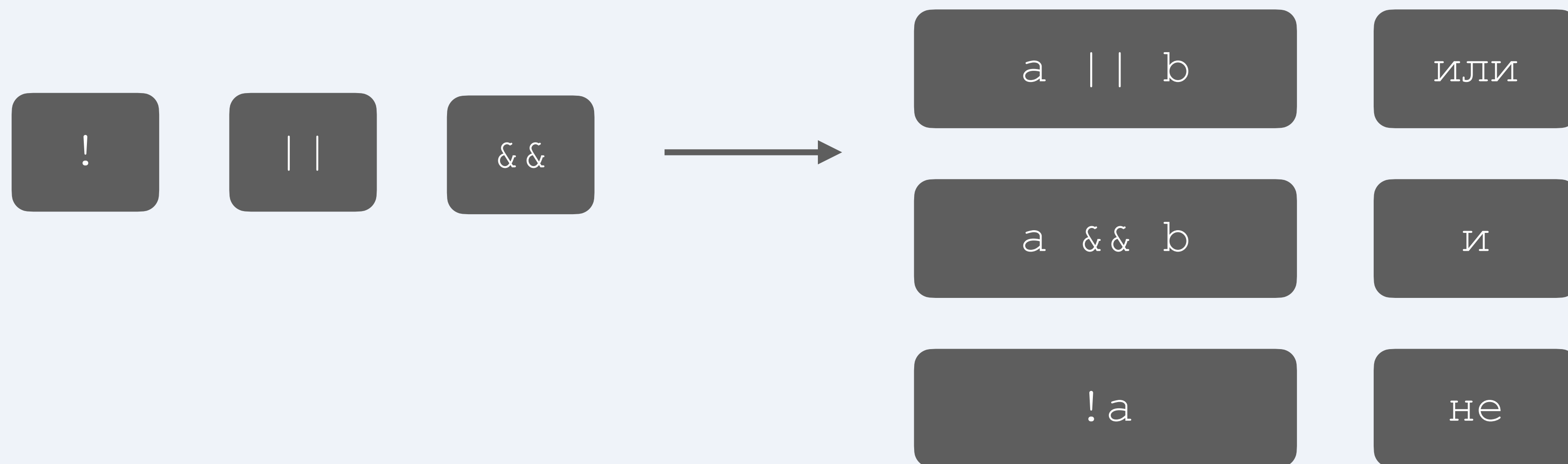
Результат сравнения возвращает логическое значение true / false





# Логические

Возвращаемое логическими операторами значение также является булевым. Однако операторы `&&` и `||` фактически возвращают значение одного из операндов, поэтому, если эти операторы используются с небулевыми величинами, то возвращаемая ими величина также может быть не булевой.

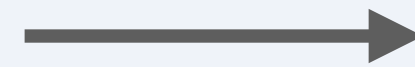


# Равенство

Результат равенства возвращает логическое значение true / false

Не строгое

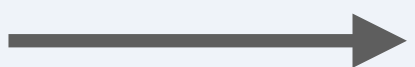
==



a == b

Строгое

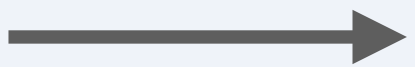
===



a === b

Не строгое не равно

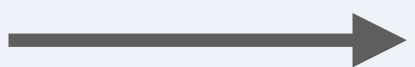
!=



a != b

Строгое не равно

!==

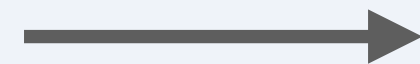


a !== b

# Оператор объединения с null

оператор `??` возвращает первый аргумент, если он не `null/undefined`, иначе второй

`??`



```
animal ?? 'неизвестное животное'
```

# Условное ветвление if( )

Инструкция if (...) вычисляет выражение в скобках и преобразует результат к логическому типу.

```
if ( 1 > 0 ) {  
    console.log( 'Правда !' )  
}
```

# Условное ветвление if else

Инструкция if может содержать необязательный блок «else» («иначе»). Он выполняется, когда условие ложно.

```
if ( 1 > 0 ) {  
    console.log('Правда !')  
} else {  
    console.log('Ложь !')  
}
```

# Условное ветвление if else if

Иногда, нужно проверить несколько вариантов условия. Для этого используется блок else if.

```
if (num === 1982 ) {  
    console.log('Правда !')  
} else if(num === 1975) {  
    console.log('Правда !')  
} else {  
    console.log('Ложь !');  
}
```

# Falsy значения

Ложное (falsy) значение - это значение, которое считается ложным, когда встречается в контексте Boolean

`undefined`

`NaN`

`null`

`false`

`0`

`''`

# Для многократного повторения одного участка кода предусмотрены **ЦИКЛЫ**

**while**

Выполняется  
пока условие  
истинно

**for**

Выполняется пока  
условие истинно и  
имеет более широкую  
настройку условий

**do while**

Устарел



# Цикл while

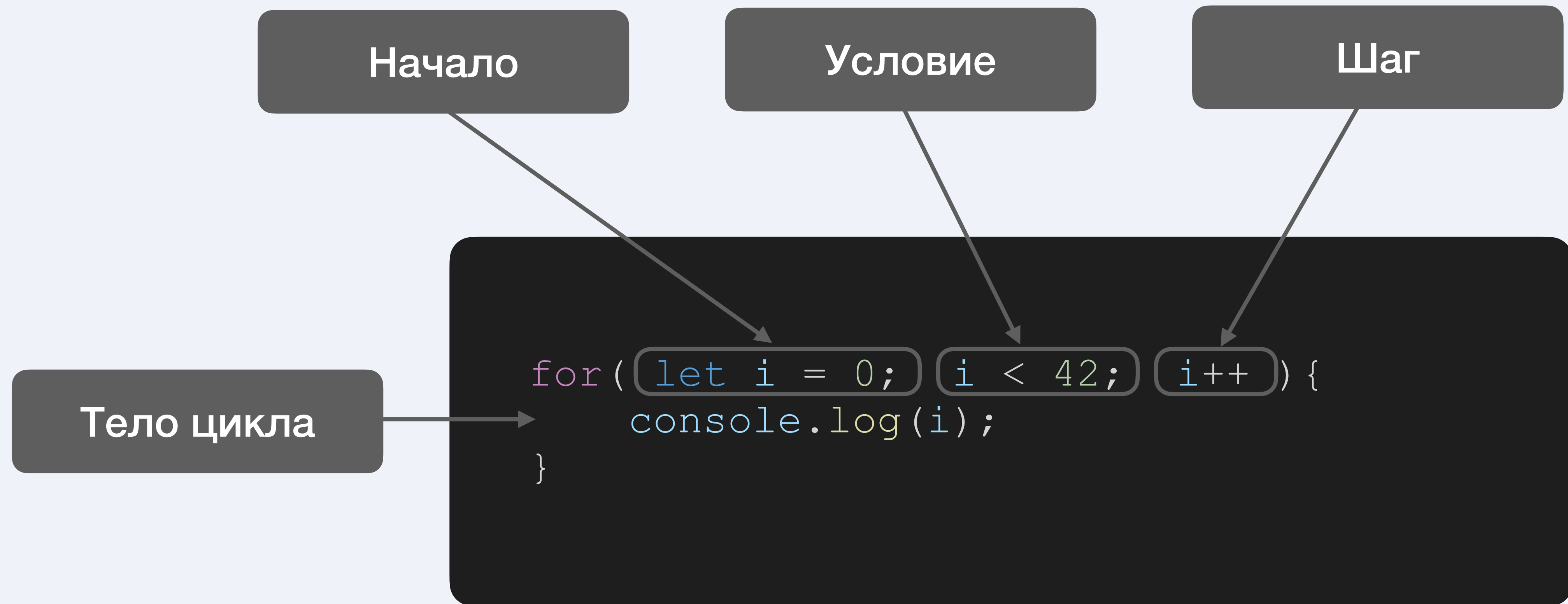
Условие

Тело цикла

```
let num = 0
while (num < 5) {
  alert(num)
  num++
}

// 0 1 2 3 4
```

# Цикл for



# Директивы break & continue

`break`

**Прерывает  
цикл**

`continue`

**Продолжает цикл и  
пропускает текущую  
итерацию**

# Конструкция switch

Конструкция **switch** за  
меняет собой сразу  
несколько **if**

- **case** проверяет соответствие с параметрами которые были переданы в конструкцию.
- **break** останавливает поиск.
- Если ни один **case** не совпал – выполняется (если есть) вариант **default**.

```
let userName = 'Donald';

switch (userName) {
  case 'Bill':
    alert( 'No! I want to see another person' );
    break;
  case 'Andy':
    alert( 'No! I want to see another person' );
    break;
  case 'Donald':
    alert( 'Hello, Donald!' );
    break;
  default:
    alert( "We don't find Donald ; (" );
}
```