

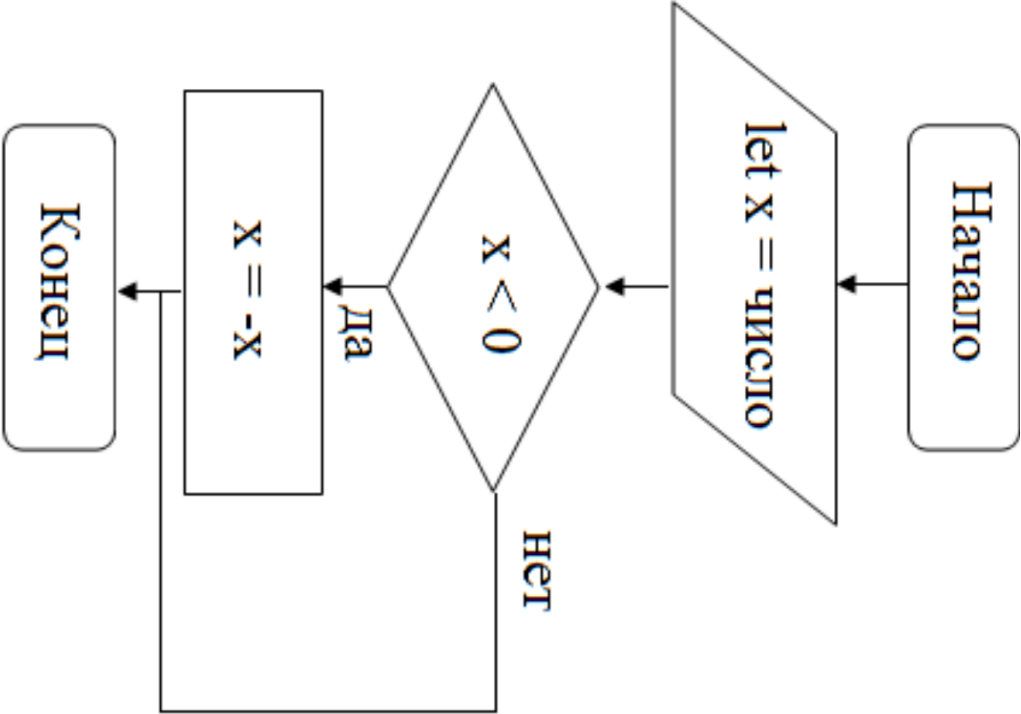
# Основные конструкции в JavaScript

Операторы `if` и `switch`, `while`, `do while`, `for`, `break`, `continue`

# Блок схема условия

$a > b$	Истинно, если $a$ больше $b$
$a < b$	Истинно, если $a$ меньше $b$
$a >= b$	Истинно, если $a$ больше или равно $b$
$a <= b$	Истинно, если $a$ меньше или равно $b$
$a == b$	Истинно, если $a$ равно $b$ (обратите внимание, для сравнения используется двойной знак равенства)
$a != b$	Истинно, если $a$ не равно $b$

```
let x = -5;  
if (x < 0) x = -x;  
console.log(`|x| = ${x}`);
```



# Оператор if и else , тернарный условный оператор

```
let x = -10, sgn = 0;
if (x < 0) {
    sgn = -1;
    console.log("x отрицательное число", sgn);
}
else if (x > 0) {
    sgn = 1;
    console.log("x положительное число", sgn);
}
else console.log("x равен 0", sgn);
```

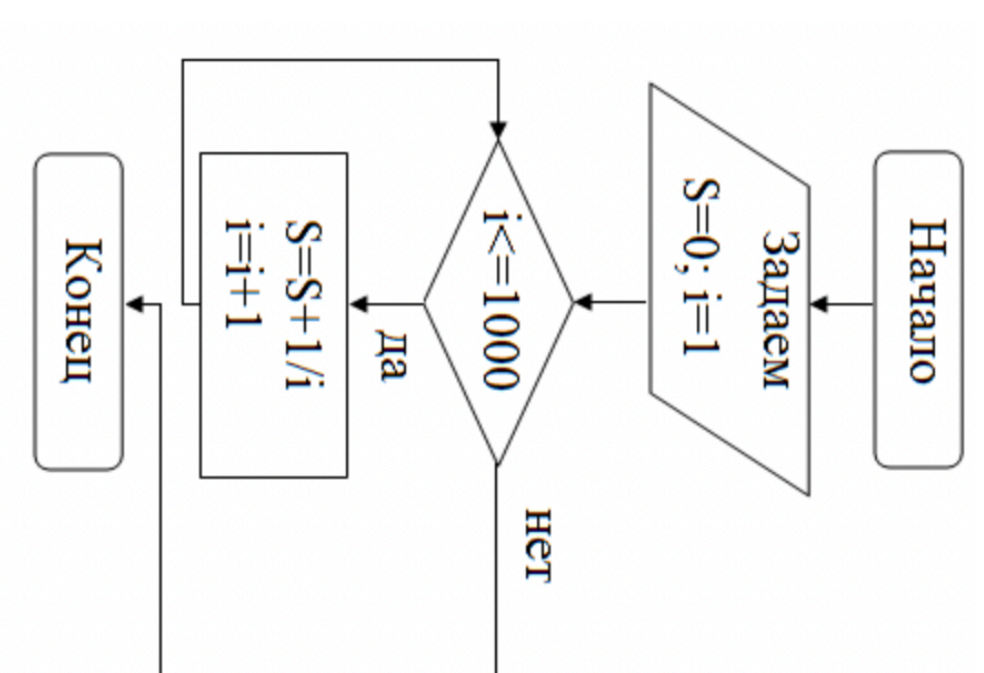
# Оператор switch

```
let item = 3;
switch (item) {
  case 1: console.log("item = 1");
  case 2: console.log("item = 2");
  case 3: console.log("item = 3");
  case 4: console.log("item = 4");
  default: console.log("item другое значение");
}
```

# Цикл While

```
while(<Выражение>) {  
    [тело цикла]  
}
```

```
let S=0, i=1;  
while(i <= 1000) {  
    S += 1/i;  
    ++i;  
}  
console.log(S);
```



# Оператор цикла for

Самый часто используемый оператор цикла – это оператор for, который имеет такой синтаксис:

```
for(<инициализация счетчика>;<условие>;<изменение счетчика>) {  
  [тело цикла]  
}
```

$$y = f(x) = kx + b, \quad x = 0; 0.1; 0.2; \dots; 1$$

```
let f, k = 0.5, b = 2;  
for(let x=0; x <= 1; x += 0.1) {  
  f = k*x+b;  
  console.log(f);  
}
```

# Оператор цикла do while

Последний оператор цикла, который мы рассмотрим – это цикл do while. Он имеет следующий синтаксис:

```
do {  
    [тело цикла]  
}  
while(<выражение>);
```

Отличие этого цикла от двух предыдущих в том, что он сначала выполняет тело цикла и только потом проверяет условие цикла: если оно истинно, то цикл продолжается, если ложно, то завершается.

```
const PSW = "password";  
let psw = null;  
do {  
    psw = prompt("Введите пароль", "");  
}  
while(psw !== PSW);  
console.log("Вы вошли в систему!");
```

# Вложенные Циклы

```
let S=0, M=10, N=5;  
for (let i=1; i <= N; ++i)  
  for (let j=1; j <= M; ++j)  
    S += i*j;
```

```
console.log("S = " + S);
```



# Оператор break

Всеми тремя циклами можно управлять с помощью двух специальных операторов: break и continue. Например, здесь при вложенных циклах:

```
for (let i=1; i <= N; ++i)
    for (let j=1; j <= M; ++j) {
        if (j == 5) break;
        S += i*j;
    }
```

будет остановлен только один внутренний цикл и управление будет передано внешнему циклу. Чтобы прервать оба цикла можно использовать оператор break с меткой вот так:

```
all: for (let i=1; i <= N; ++i)
    for (let j=1; j <= M; ++j) {
        if (j == 5) break all;
        S += i*j;
    }
```

# Оператор continue

Следующий оператор continue позволяет пропускать тело цикла и перейти к следующей итерации, не прерывая работу самого цикла.

```
for (let i=-5; i <= 5; ++i) {  
    if (i == 0) continue;  
    console.log("i = " + i);  
}
```

При выполнении этой программы увидим, что в консоль выведены все значения кроме нуля. Так как при `i=0` срабатывает условие и выполняется оператор `continue`. Все что находится после этого оператора пропускается и цикл продолжается уже со значением `i=1`.

Оператор `continue` также можно использовать с меткой, подобно оператору `break`.