

## Условные операторы в языке JavaScript

Условные конструкции позволяют проверить некоторое выражение и в зависимости от его результата выполнить необходимый код. В уроке мы познакомимся с конструкцией «if - else» и конструкцией «switch - case».

Условные операторы очень важны, ведь они позволяют выполнять разный код в зависимости от исхода условия. В языке JavaScript существует три конструкции позволяющих осуществлять проверку.

### Конструкция if - else

За счёт if else можно проверить одно или несколько условий и в случае их успешной проверки будет выполнен один, иначе другой.

Рассмотрим пример:

```
var a = 2, b = 10;
if (a == b) { // Если a будет равным b, тогда будет выполнен код
    // Здесь код который будет выполнен
    // Если все одна строка кода, то фигурные скобки {}
    // можно не ставить
    console.log("a == b");
} else if (a <= b) { // Если a будет меньшим или равным b
    // Если предыдущее условие не будет выполнено,
    // а здесь условие окажется верным,
    // то будет выполнен этот код
    console.log("a <= b");
} else {
    // Этот код сработает, если другие условия не будут выполнены
    console.log("a > b");
}
```

Вы можете прописать структуру лишь с одним условием `if`, а можете дописать в неё сколько-угодно вложенных условий `else if`.

Внутри каждого из условий можно прописывать другие конструкции `if else`, что будут проверять новые выражения.

Если необходимо проверить несколько условий в одном операторе, то можно воспользоваться логическим «и» или же логическим «или»:

```
if (a !== b && a > b) {  
    // Код будет выполнен, если и первое, и второе условие  
    // окажутся верными  
}  
  
if (a < b || a == b) {  
    // Код будет выполнен, если или первое, или второе  
    // условие  
    // окажется верным  
}
```

Конструкция switch

Конструкция `case` обладает более удобным форматом для проверки множественных условий на совпадение значения. В конструкцию записывается переменная, что проверяется, а также значения на которые происходит проверка.

Пример оператора:

```
var x = 23;  
switch (x) { // Проверяем переменную x  
    case 1: // Если переменная будет равна 1, то здесь  
        // сработает код  
        // Может быть множество строк, а не только одна  
        console.log("Переменная равна 1");  
        break; // Указываем конец для кода для этой  
        // проверки  
    case 56: // Если переменная будет равна 56, то здесь  
        // сработает код  
        // Может быть множество строк, а не только одна  
        console.log("Переменная равна 56");  
        break; // Указываем конец для кода для этой  
        // проверки  
    // По аналогии таких проверок может быть множество  
    // Также можно добавить проверку, которая сработает в  
    // случае
```

```
// если все остальные проверки не сработают
default:
    console.log("Что-то другое");
    break; // Можно и не ставить, так как это последнее
условие
}
```

Тернарная операция

Это сокращенная форма `if else`. Записывается следующим образом:

```
var x = 90, a = 8;
var res = x < a ? (x + a) : (x - a);
console.log(res);
```

Мы проверяем, если "x" будет меньше за "a", то мы в переменную `res` устанавливаем их сумму, иначе - их разницу.

```
var number = 15;
var isHasHouse = true;

if(number == 15 && !isHasHouse) {
    console.log("Ok");
} else if(number < 10) {
    console.log("Ok!");
} else if(number == 7) {
    console.log("7!");
} else if(number > 15) {
    console.log("> 15!");
} else {
    console.log("Else!");
}
```

Оператор «switch case»

```
var stroka = "word23";

switch(stroka) {
    case 4:
        console.log("Переменная со значением 4");
        break;
    case "45":
        console.log("Переменная со значением 45");
}
```

```
    break;
case "word":
    console.log("Переменная со значением «word»");
    break;
default:
    console.log("Default");
    break;
}
```