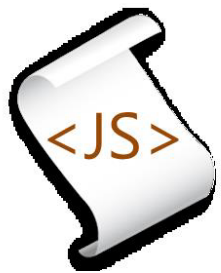


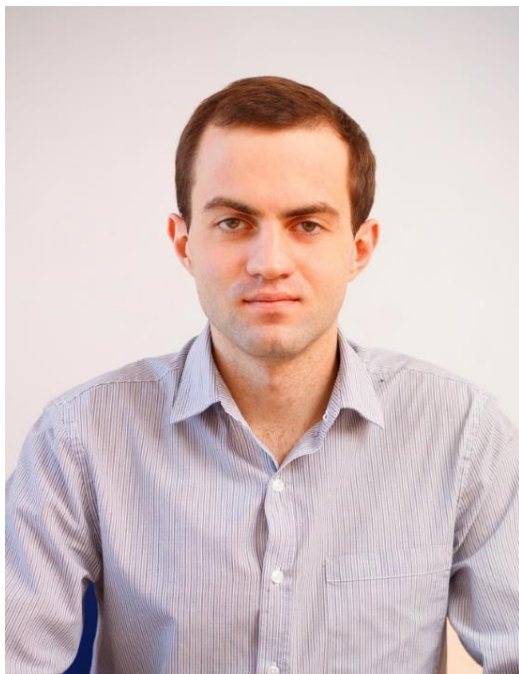
# JavaScript Essential



Условные и циклические конструкции

# JavaScript Essential

Автор курса



Дмитрий Охрименко  
МСТ



MCID: 9210561

# JavaScript Essential

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)

Доступ можно получить через руководство Вашего учебного центра



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

# JavaScript Essential

Тема

## Условные и циклические конструкции

# Условные конструкции

## Selection Statements

Оператор ветвления (условный оператор) – конструкция языка программирования, обеспечивающая выполнение определённой команды (набора команд) только при условии истинности некоторого логического выражения

**ЕСЛИ:** (*Условие удовлетворяет истинности*)

**ТО:** Выполнить эту серию инструкций

**ИНАЧЕ:** Выполнить эту серию инструкций

# Условные конструкции

## Selection Statements

Условный оператор: `if ... else`

Тернарный оператор: `... ? ... : ...`

Оператор многозначного выбора: `switch – case`

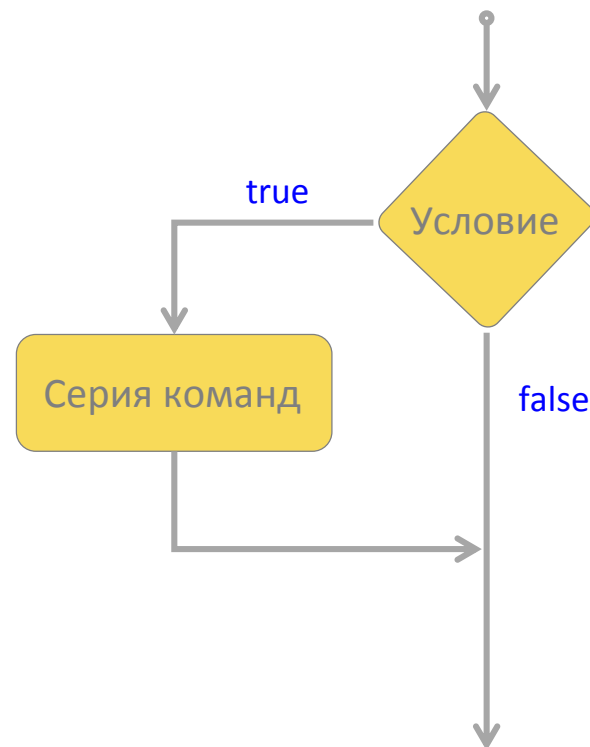
# if

## Условный оператор

Реализует выполнение определённых команд при условии, что используемое логическое выражение в условии удовлетворяет истинности. Результатом вычисления логического выражения должно быть **true** или **false**.

```
var a = 10;
```

```
if (a == 10) {  
    alert("a = 10");  
};
```



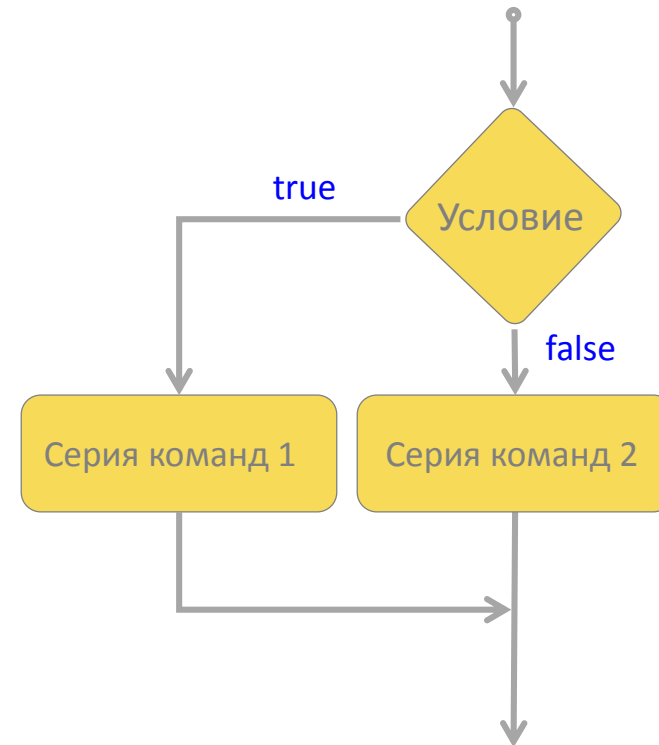
# if-else

## Условный оператор

В зависимости от условия выполняется только одна из двух серий команд, входящих в команду ветвления. Если условие соблюдено, то надо выполнить серию 1, а если нет – то серию 2.

```
var a =10;
```

```
if (a ==10){  
    alert("a = 10");  
}  
else {  
    alert("a != 10");  
};
```





? :

## Тернарный (третичный) оператор

**Тернарный оператор** – операция, возвращающая свой второй или третий операнд в зависимости от значения логического выражения, заданного первым операндом.



```
var a = 10;
```

```
var msg = (a == 10) ? "a = 10" : "a != 10";
```

```
alert(msg);
```

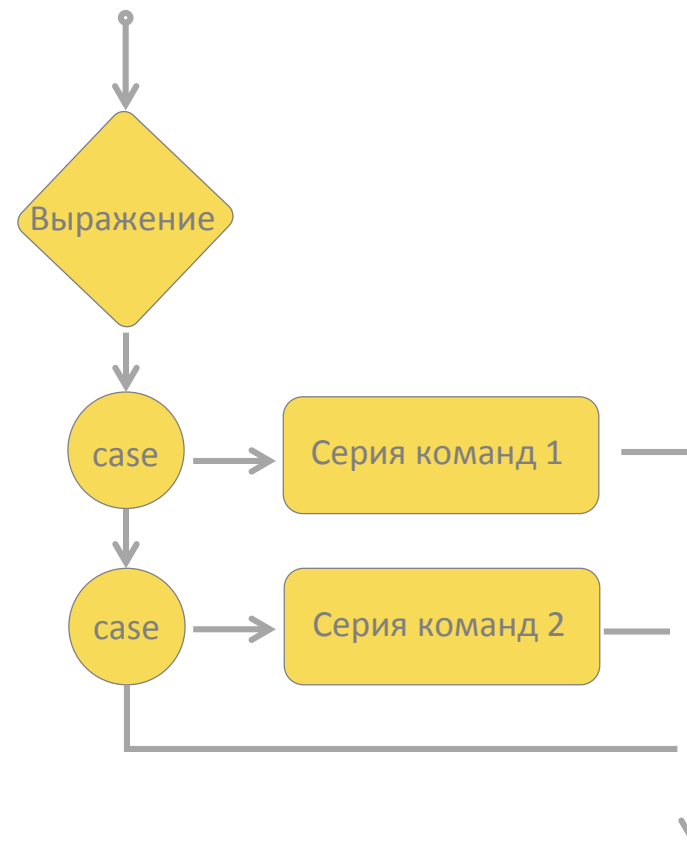
# switch-case

## Оператор многозначного выбора (переключатель)

Конструкция `switch` служит для сравнения значения на равенство с различными вариантами.

```
var day = "10";
```

```
switch (day) {  
  case "10":  
    {  
      alert("a = 10");  
    }  
    break;  
  case "11":  
    {  
      alert("a = 11");  
    }  
    break;  
};
```



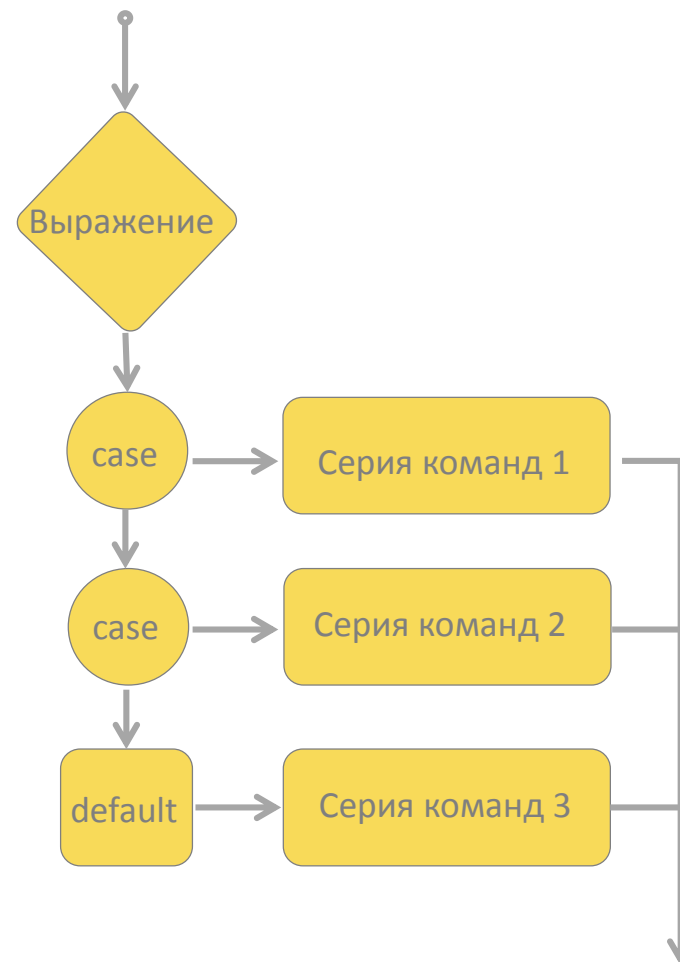
# switch-case

## Оператор многозначного выбора (переключатель)

В конструкции переключателя `switch-case` может присутствовать блок `default`.

```
var day = "10";
```

```
switch (day) {  
  case "10":  
  {  
    alert("a = 10");  
  };  
  break;  
  case "11":  
  {  
    alert("a = 11");  
  };  
  break;  
  default:  
  {  
    alert("совпадений нет");  
  };  
};
```



# Циклические конструкции

## Команды повторения

**Цикл** – управляющая конструкция, предназначенная для организации многократного исполнения набора инструкций. Один проход цикла называется итерацией.

Циклом может называться любая многократно исполняемая последовательность инструкций, организованная любым способом (например, с помощью условного или безусловного перехода).

**ПОКА:** (*Условие удовлетворяет истинности*)

**НАЧАЛО ЦИКЛА:** Выполнить эту серию инструкций;

**КОНЕЦ ЦИКЛА:**

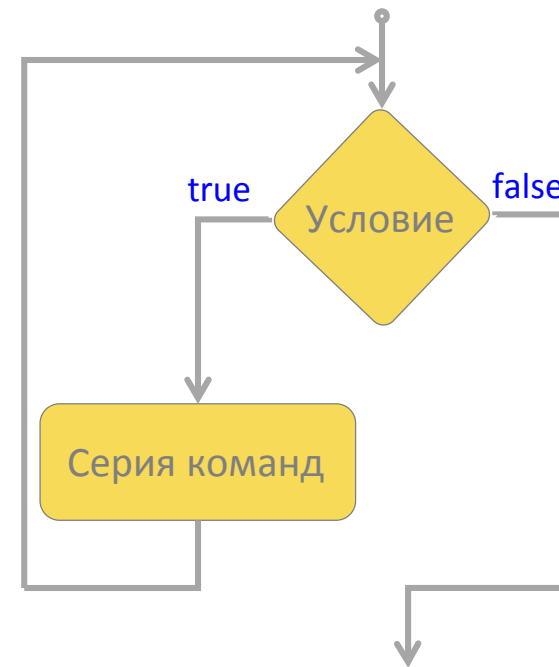
# while

## Цикл с предусловием

Цикл, с предусловием **while** – это цикл, который выполняется до тех пор, пока условие удовлетворяет истинности.

```
var counter = 0;
```

```
while (counter < 10) {  
    counter++;  
    document.write(counter + "<br />");  
}
```



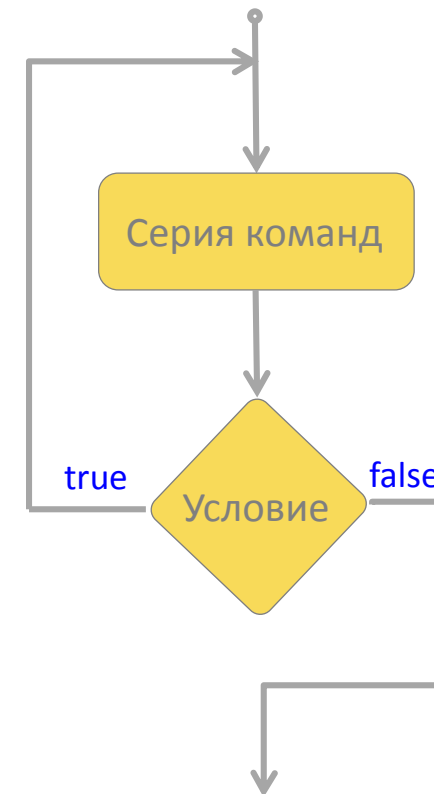
# do - while

## Цикл с постусловием

Цикл с постусловием **do-while** – это цикл, в котором условие проверяется после выполнения тела цикла. Тело цикла **do-while** выполняется хотя бы один раз.

```
var counter = 0;
```

```
do {  
    counter++;  
    document.write(counter + "<br />");  
}  
while (counter < 10);
```

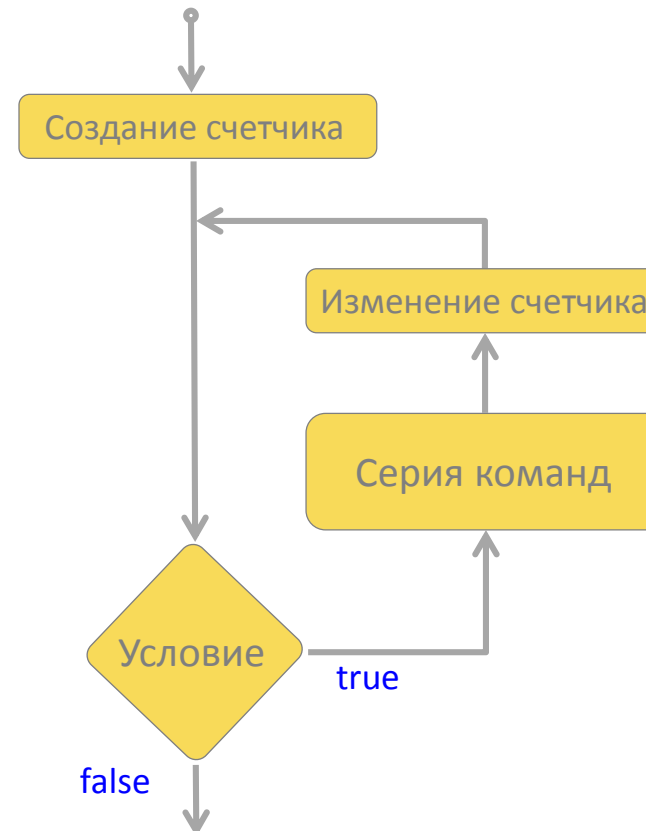


# for

## Цикл со счётчиком

Цикл со счётчиком **for** – это цикл, в котором переменная – счётчик итераций цикла, с определенным шагом, изменяет свое значение до заданного конечного значения.

```
for (var i = 0; i < 10; i++) {  
    document.write(i + "<br />");  
}
```



# Смотрите наши уроки в видео формате

ITVDN.com



Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.

Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics





# Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](http://TestProvider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



# JavaScript Essential

Q&A

# Информационный видеосервис для разработчиков программного обеспечения

