



JavaScript Syntax

JavaScript Syntax

Conditional Statements, Loops

Au Mau Duong
Technical Trainers

JavaScript Syntax

Table of Contents

1. Conditional Statements

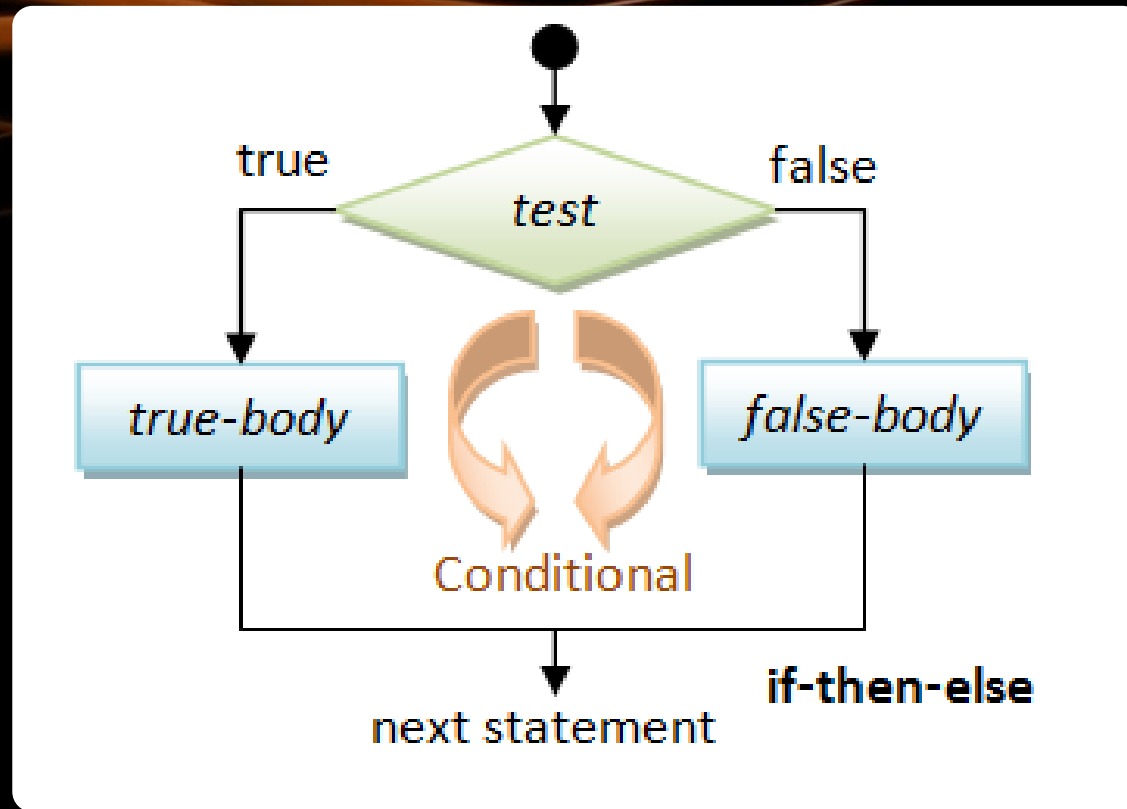
- If-else, switch-case

2. False-like Conditions

- Falsy/Truthy conditions

3. Loops: for, while, do-while, ...





if and if-else

Implementing Conditional Logic

Conditions: if-else

- JavaScript implements the classical **if** / **if-else** statements:

```
let number = 5;  
  
if (number % 2 == 0) {  
    console.log("Even number");  
}  
else {  
    console.log("Odd number");  
}
```


if-else practice - Exercise 1

Write a JS program to find maximum between two numbers using if else:

- *Input*

- Input num1: 10
- Input num2: 20

- *Output*

- Maximum = 20

if-else practice - Exercise 2

Write a JS program to find maximum between three numbers using ladder if else or nested if

■ *Input*

- *Input num1: 10*
- *Input num2: 20*
- *Input num3: 15*

■ *Output*

- *Maximum is: 20*

if-else practice - Exercise 3

Write a JS program to check whether a number is divisible by 5 and 11 or not, using if else

- *Input*

- *Input number: 55*

- *Output*

- *Number is divisible by 5 and 11*

if-else practice – Exercise 4

Write a JS program to check leap year using if else

- *Input*

- *Input year: 2004*

- *Output*

- *2004 is leap year.*

if-else practice – Exercise 5

Write a JS program to input week number(1-7) and print the corresponding day of week name using if else

- *Input*

- *Input week number: 1*

- *Output*

- *Monday*



switch-case

Making Several Comparisons at Once

The switch-case Statement

- Selects for execution a statement from a list depending on the value of the **switch** expression

```
switch (day) {  
    case 1: console.log('Monday'); break;  
    case 2: console.log('Tuesday'); break;  
    case 3: console.log('Wednesday'); break;  
    case 4: console.log('Thursday'); break;  
    case 5: console.log('Friday'); break;  
    case 6: console.log('Saturday'); break;  
    case 7: console.log('Sunday'); break;  
    default: console.log('Error!'); break;  
}
```

How switch-case Works?

1. The expression is evaluated
2. When one of the constants specified in a case label is equal to the expression
 - The statement that corresponds to that case is executed
3. If no case is equal to the expression
 - If there is default case, it is executed
 - Otherwise the control is transferred to the end point of the switch statement
4. The **break** statement exits the switch-case statement

switch-case practice – Exercise

Write a JS program to enter month number between(1-12) and print number of days in month using switch-case

- *Input*

- *Enter month number: 1*

- *Output*

- *It contains 31 days.*

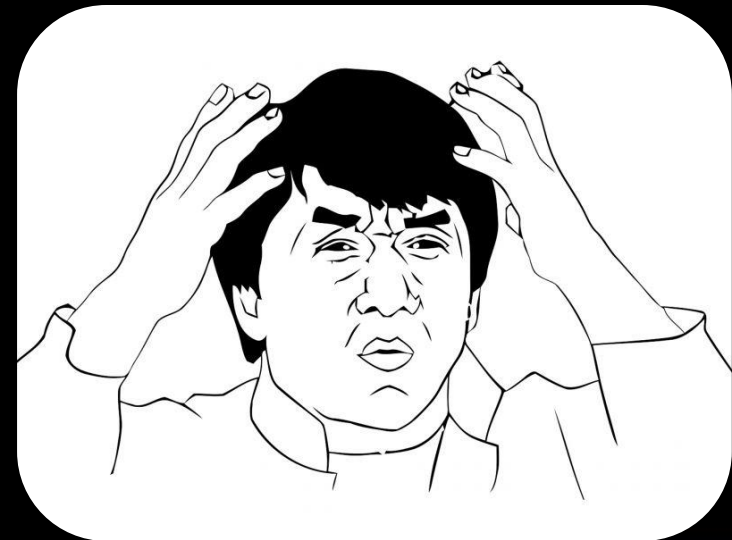
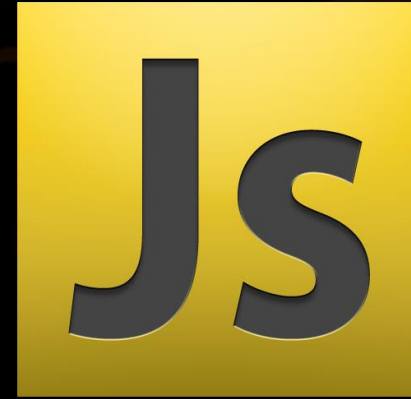


False-like Conditions

Unexpected (for Some People) Behavior

False-like Conditions

- Values converted to **false**
 - **0** == **false** (zero)
 - **"0"** == **false** (zero as string)
 - **""** == **false** (empty string)
 - **[]** == **false** (empty array)
- Values converted to **true**
 - **1** == **true** (one)
 - **"1"** == **true** (one as string)
 - **!0** == **true** (the opposite of **0**)



Truthy values in conditions

- Values evaluated as truthy in conditions
 - **true** // Truthy!
 - **{}** // Truthy!
 - **[]** // Truthy!
 - **"some string"** // Truthy!
 - **3.14** // Truthy!
 - **new Date()** // Truthy!



Falsy values in conditions

- Values evaluated as falsy in conditions
 - **false** // Falsy.
 - **null** // Falsy.
 - **undefined** // Falsy.
 - **NaN** // Falsy.
 - **0** // Falsy.
 - **""** // Falsy.

Truthy and Falsy

- Falsy Values:

- false
- 0
- ""
- null
- undefined
- NaN

```
1 Ti.API.info(false == 0); //true
2 Ti.API.info(false == ''); //true
3 Ti.API.info(0 == ''); //true
4
5 if (5) {
6     alert('this code will get executed');
7 }
8
9 //Dude. WTF.
```

- Truthy Values:

- Everything else

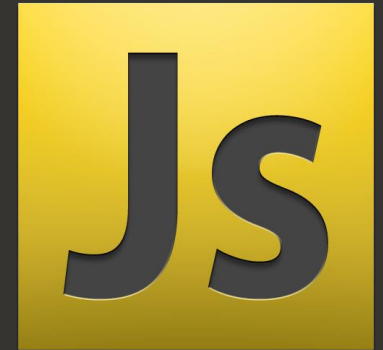
- Almost always, you want === and !==



Unexpected / Strange Behavior in JavaScript

- JavaScript is rich of unexpected (for some people) behavior

```
"0" == false // true  
if ("0") console.log(true); // true  
  
[] == false // true  
if ([]) console.log(true); // true  
  
null == false // false  
!null // true
```



Loops: for, while, do-while, ...

- The **for** / **while** / **do-while** loops work as in C++, C# and Java

```
for (let i = 0; i <= 10; i++)  
  console.log(i) // 0 1 2 3 4 ... 10
```

```
let count = 1  
while (count < 1024)  
  console.log(count *= 2) // 2 4 8 16 ... 1024
```

```
let s = "ha"  
do { console.log(s); s = s + s; }  
while (s.length < 10) // ha haha hahahaha
```

Loops practice – Exercise 1

Write a JS program to print all natural numbers from 1 to n using loop

- Input

- Input upper limit: 10

- Output

- Natural numbers from 1 to 10: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Loops practice – Exercise 2

Write a JS program to print all natural numbers in reverse from n to 1 using for loop

- Input

- Input N: 10

- Output

- Natural numbers from 10-1 in reverse:
- 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

Loops practice – Exercise 3

Write a JS program to print all even numbers from 1 to n using for loop

- Input

- Input upper range: 10

- Output

- Even numbers between 1 to 10:
- 2, 4, 6, 8, 10

Loops practice – Exercise 4

Write a JS program to find the sum of all natural numbers between 1 to n using for loop

- Input

- Input upper limit: 10

- Output

- Sum of natural numbers 1-10: 55

Loops practice – Exercise 5

Write a JS program to find sum of all odd numbers from 1 to n using for loop

- Input

- Input upper limit: 10

- Output

- Sum of odd numbers from 1-10: 25

Summary

- If-else statements (same as in C#, Java and C++)
- Switch-case statement (similar to Java / C#)
- False-like Conditions
 - Falsy/Truthy conditions
- The **for** / **while** / **do-while** loops work as in C++, C# and Java



JS Basics

JavaScript Syntax



Questions?

