

Conditional Statements

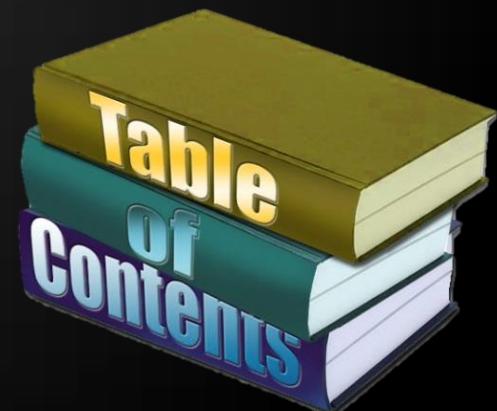
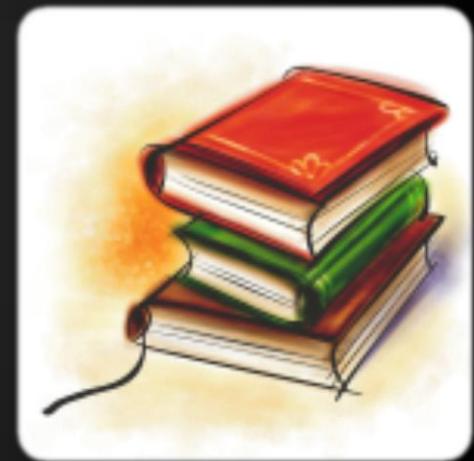
Implementing Control Logic in JavaScript

Telerik Software Academy
Learning & Development Team
<http://academy.telerik.com>



Table of Contents

1. The if Statement
2. The if-else Statement
3. Nested if Statements
4. The switch-case Statement



if and if-else

Implementing Conditional Logic



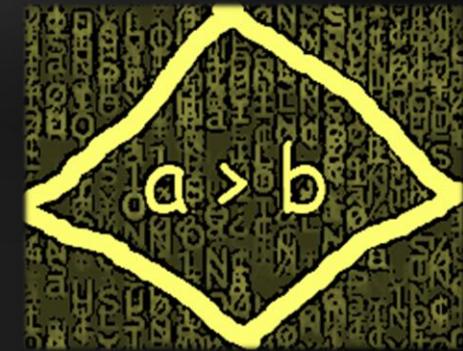
The if Statement

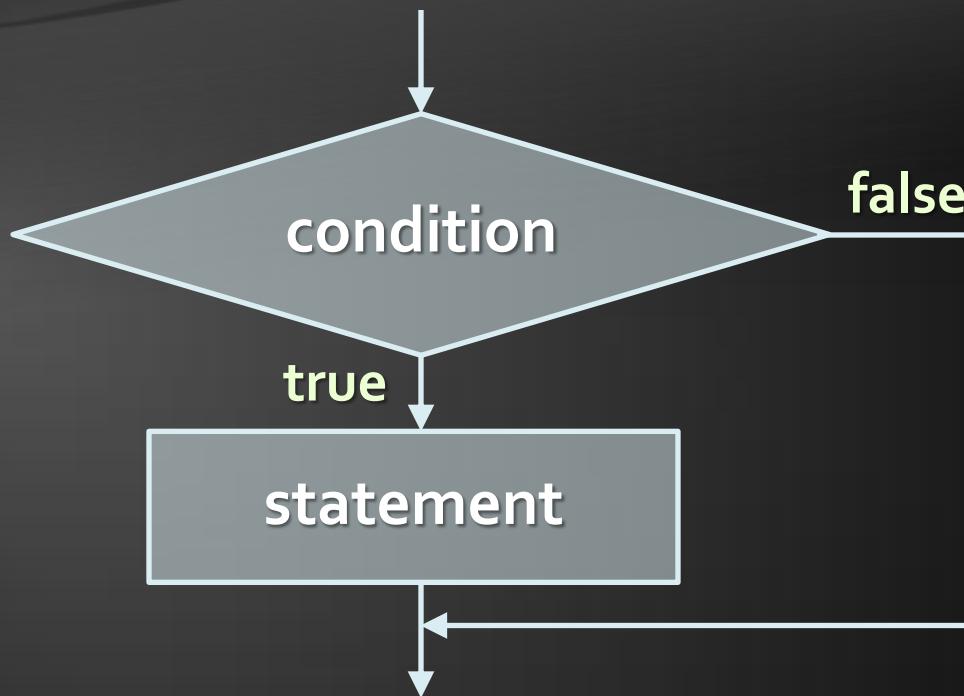
- ◆ The most simple conditional statement
- ◆ Enables you to test for a condition
- ◆ Branch to different parts of the code depending on the result
- ◆ The simplest form of an **if** statement:

```
if (condition)
{
    statements;
}
```

Condition and Statement

- ◆ The condition can be:
 - ◆ Boolean variable
 - ◆ Boolean logical expression
 - ◆ Comparison expression
 - ◆ Integer, object, function... everything
- ◆ The condition can be of any type
- ◆ The statement can be:
 - ◆ Single statement ending with a semicolon
 - ◆ Block enclosed in braces





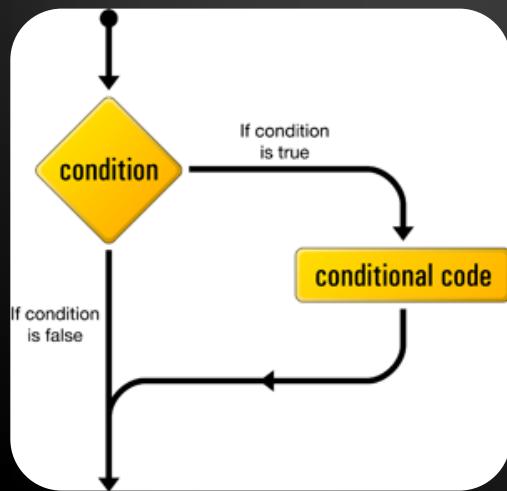
- ◆ The condition is evaluated
 - If it is true, the statement is executed
 - If it is false, the statement is skipped

The if Statement – Example

```
var bigger = parseInt(document.  
                      getElementById('first-tb').value);  
var smaller = parseInt(document.  
                      getElementById('second-tb').value);  
  
if (smaller > bigger)  
{  
    bigger = smaller;  
}  
  
console.log('The greater number is: ' + bigger);  
}
```

The if Statement

Live Demo

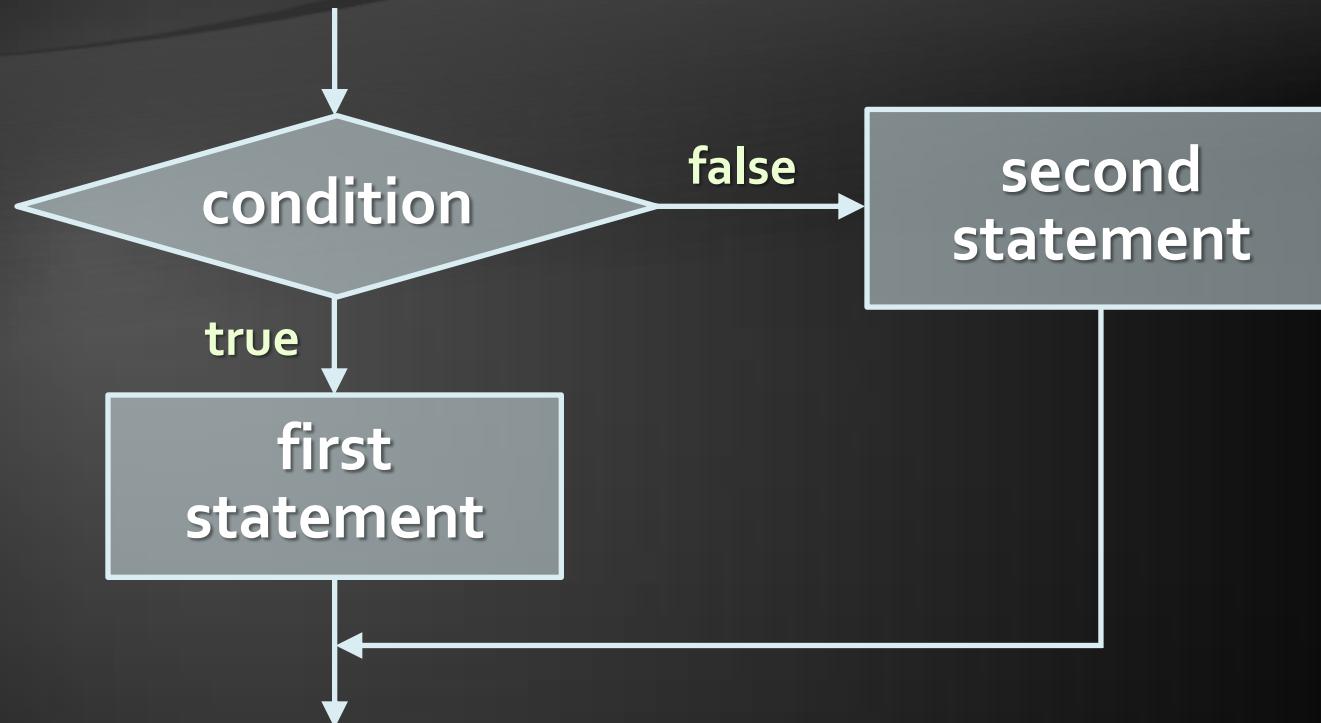


if
!

The if-else Statement

- ◆ More complex and useful conditional statement
- ◆ Executes one branch if the condition is true, and another if it is false
- ◆ The simplest form of an if-else statement:

```
if (expression)
{
    statement1;
}
else
{
    statement2;
}
```



- ◆ The condition is evaluated
 - If it is true, the first statement is executed
 - If it is false, the second statement is executed

if-else Statement – Example

- ◆ Checking a number if it is odd or even

```
var s = document.getElementById('number-tb').value;
var number = parseInt(s);

if (number % 2 === 0)
{
    console.log('This number is even.');
}
else
{
    console.log('This number is odd.');
}
```

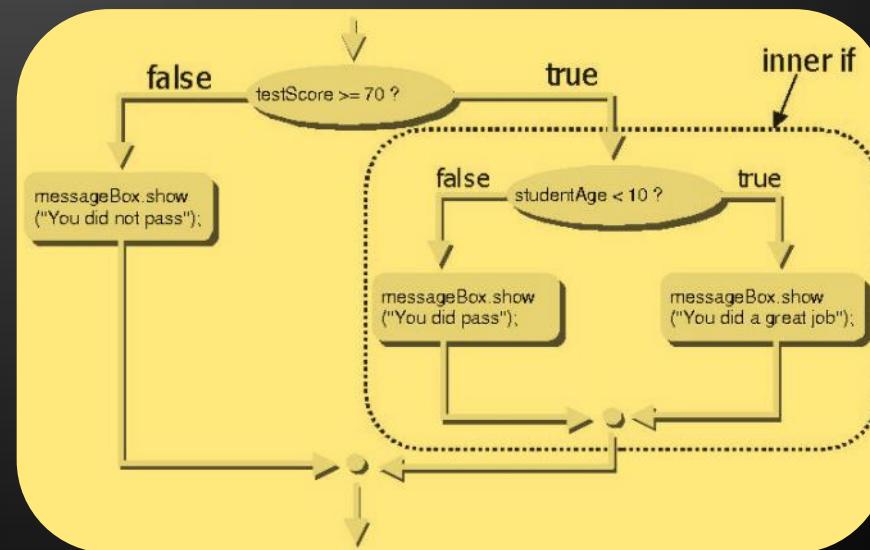
The if-else Statement

Live Demo



Nested if Statements

Creating More Complex Logic



Nested if Statements

- ◆ if and if-else statements can be nested, i.e. used inside another if or else statement
- ◆ Every else corresponds to its closest preceding if

```
if (expression)
{
    if (expression)
    {
        statement;
    }
    else
    {
        statement;
    }
}
else
    statement;
```

Nested if – Good Practices

- ◆ Always use { ... } blocks to avoid ambiguity
 - Even when a single statement follows
- ◆ Avoid using more than three levels of nested if statements
- ◆ Put the case you normally expect to process first, then write the unusual cases
- ◆ Arrange the code to make it more readable

Nested if Statements – Example

```
if (first === second)
{
    console.log(
        'These two numbers are equal.');
}
else
{
    if (first > second)
    {
        console.log(
            'The first number is bigger.');
    }
    else
    {
        console.log('The second is bigger.');
    }
}
```

Nested if Statements

Live Demo

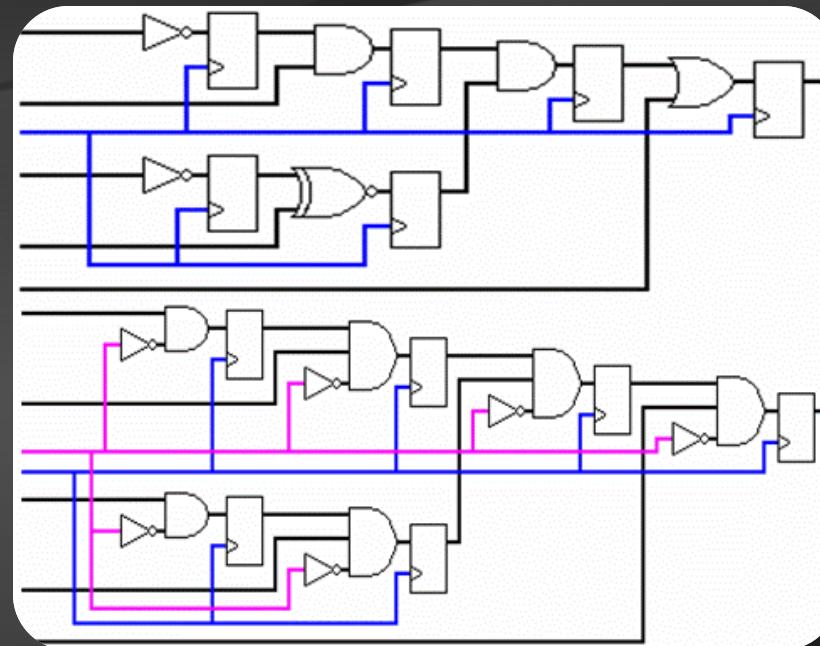


```
if (expression)
{
    if (expression)
    {
        if (expression)
        {
            if (expression)
            {
                else
            }
        }
        else
    }
}
else
```

Multiple if-else-if-else-...

- ◆ Sometimes we need to use another if-construction in the else block
 - ◆ Thus else if can be used:

```
var ch = 'X';
if (ch === 'A' || ch === 'a')
{
    console.log('Vowel [ei]');
}
else if (ch === 'E' || ch === 'e')
{
    console.log('Vowel [i:]');
}
else if ...
else ...
```



Multiple if-else Statements

Live Demo

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



The **switch-case** **Statement**

Live Demo

False-like conditions

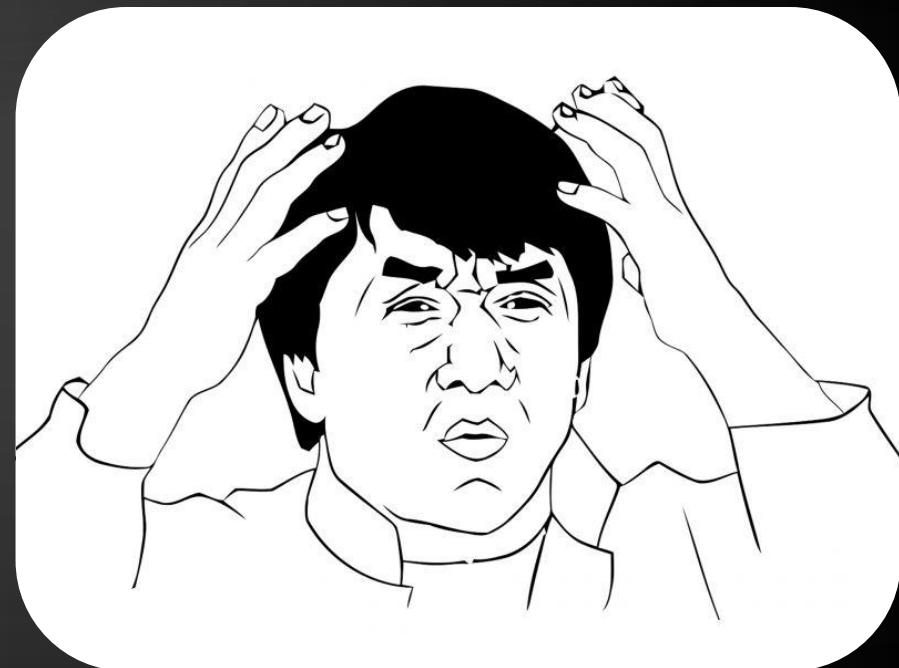
Strange behavior



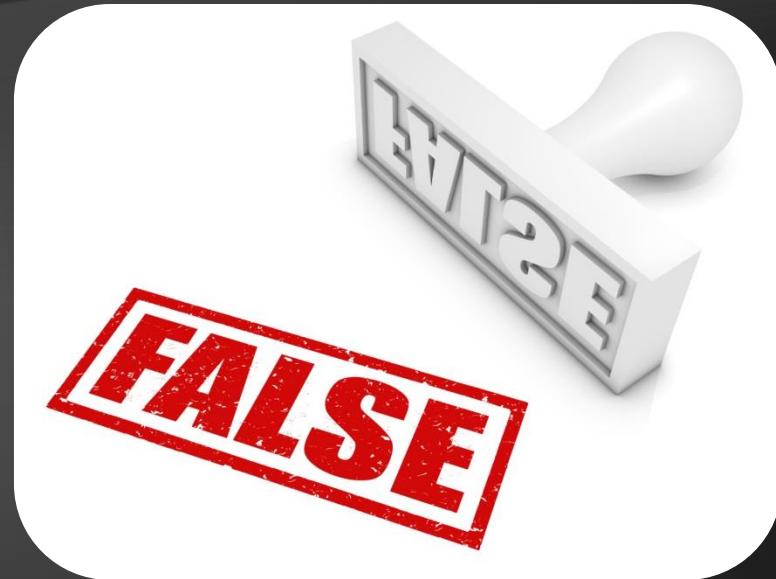
False-like conditions

- ◆ These values are always false

- ◆ `false`
 - ◆ `0` (zero)
 - ◆ `""` (empty string)
 - ◆ `null`
 - ◆ `undefined`
 - ◆ `NaN`



- ◆ All other values are true
- ◆ Info: <http://www.sitepoint.com/javascript-truthy-falsy/>



False-like Conditions

Live Demo

- ◆ Comparison and logical operators are used to compose logical conditions
- ◆ The conditional statements `if` and `if-else` provide conditional execution of blocks of code
 - Constantly used in computer programming
 - Conditional statements can be nested
- ◆ The `switch` statement easily and elegantly checks an expression for a sequence of values

Conditional Statements

Questions?



1. Write an **if** statement that examines two integer variables and exchanges their values if the first one is greater than the second one.
2. Write a script that shows the sign (+ or -) of the product of three real numbers without calculating it. Use a sequence of **if** statements.
3. Write a script that finds the biggest of three integers using nested **if** statements.
4. Sort 3 real values in descending order using nested **if** statements.

Exercises (2)

5. Write script that asks for a digit and depending on the input shows the name of that digit (in English) using a switch statement.
6. Write a script that enters the coefficients a, b and c of a quadratic equation

$$a*x^2 + b*x + c = 0$$

and calculates and prints its real roots. Note that quadratic equations may have 0, 1 or 2 real roots.

7. Write a script that finds the greatest of given 5 variables.

8. Write a script that converts a number in the range [0...999] to a text corresponding to its English pronunciation. Examples:

0 → 'Zero'

273 → 'Two hundred seventy three'

400 → 'Four hundred'

501 → 'Five hundred and one'

711 → 'Seven hundred and eleven'