



HTML

CSS

JAVASCRIPT

MORE ▼



JavaScript Switch Statement

[< Previous](#)[Next >](#)

The `switch` statement is used to perform different actions based on different conditions.

The JavaScript Switch Statement

Use the `switch` statement to select one of many code blocks to be executed.

Syntax

```
switch(expression) {  
  case x:  
    // code block  
    break;  
}
```

```
case y:
  // code block
  break;
default:
  // code block
}
```

This is how it works:

- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.

Example

The `getDay()` method returns the weekday as a number between 0 and 6.

(Sunday=0, Monday=1, Tuesday=2 ..)

This example uses the weekday number to calculate the weekday name:

```
switch (new Date().getDay()) {
  case 0:
    day = "Sunday";
    break;
  case 1:
    day = "Monday";
    break;
  case 2:
    day = "Tuesday";
    break;
  case 3:
```

```
    day = "Wednesday";  
    break;  
case 4:  
    day = "Thursday";  
    break;  
case 5:  
    day = "Friday";  
    break;  
case 6:  
    day = "Saturday";  
}
```

The result of day will be:

Thursday

Try it Yourself »

The break Keyword

When JavaScript reaches a **break** keyword, it breaks out of the switch block.

This will stop the execution of inside the block.

It is not necessary to break the last case in a switch block. The block breaks (ends) there anyway.

Note: If you omit the break statement, the next case will be executed even if the evaluation does not match the case.

The default Keyword

The `default` keyword specifies the code to run if there is no case match:

Example

The `getDay()` method returns the weekday as a number between 0 and 6.

If today is neither Saturday (6) nor Sunday (0), write a default message:

```
switch (new Date().getDay()) {  
  case 6:  
    text = "Today is Saturday";  
    break;  
  case 0:  
    text = "Today is Sunday";  
    break;  
  default:  
    text = "Looking forward to the Weekend";  
}
```

The result of text will be:

Looking forward to the Weekend

Try it Yourself »

The **default** case does not have to be the last case in a switch block:

Example

```
switch (new Date().getDay()) {  
  default:  
    text = "Looking forward to the Weekend";  
    break;  
  case 6:  
    text = "Today is Saturday";  
    break;  
  case 0:  
    text = "Today is Sunday";  
}
```

Try it Yourself »

If **default** is not the last case in the switch block, remember to end the default case with a break.

Common Code Blocks

Sometimes you will want different switch cases to use the same code.

In this example case 4 and 5 share the same code block, and 0 and 6 share another code block:

Example

```
switch (new Date().getDay()) {  
  case 4:  
  case 5:  
    text = "Soon it is Weekend";  
    break;  
  case 0:  
  case 6:  
    text = "It is Weekend";  
    break;  
  default:  
    text = "Looking forward to the Weekend";  
}
```

Try it Yourself »

Switching Details

If multiple cases matches a case value, the **first** case is selected.

If no matching cases are found, the program continues to the **default** label.

If no default label is found, the program continues to the statement(s) **after the switch**.

Strict Comparison

Switch cases use **strict** comparison (===).

The values must be of the same type to match.

A strict comparison can only be true if the operands are of the same type.

In this example there will be no match for x:

Example

```
var x = "0";
switch (x) {
  case 0:
    text = "Off";
    break;
  case 1:
    text = "On";
    break;
  default:
    text = "No value found";
}
```

Try it Yourself »

Test Yourself With Exercises

Exercise:

Create a `switch` statement that will alert "Hello" if `fruits` is "banana", and "Welcome" if `fruits` is "apple".

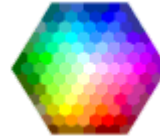
```
(fruits) {  
  "Banana":  
    alert("Hello")  
    break;  
  "Apple":  
    alert("Welcome")  
    break;  
}
```

[Submit Answer »](#)

[Start the Exercise](#)

[< Previous](#)[Next >](#)

COLOR PICKER



HOW TO

Tabs
Dropdowns
Accordions
Side Navigation
Top Navigation
Modal Boxes
Progress Bars
Parallax
Login Form
HTML Includes
Google Maps
Range Sliders
Tooltips
Slideshow
Filter List
Sort List

SHARE



CERTIFICATES

HTML

CSS

JavaScript

SQL

Python

PHP

jQuery

Bootstrap

XML

[Read More »](#)



[REPORT ERROR](#)

[PRINT PAGE](#)

[FORUM](#)

[ABOUT](#)

[Top Tutorials](#)

[Top References](#)

- HTML Tutorial
- CSS Tutorial
- JavaScript Tutorial
- How To Tutorial
- SQL Tutorial
- Python Tutorial
- W3.CSS Tutorial
- Bootstrap Tutorial
- PHP Tutorial
- jQuery Tutorial
- Java Tutorial
- C++ Tutorial

Top Examples

- HTML Examples
- CSS Examples
- JavaScript Examples
- How To Examples
- SQL Examples
- Python Examples
- W3.CSS Examples
- Bootstrap Examples
- PHP Examples
- jQuery Examples
- Java Examples
- XML Examples

- HTML Reference
- CSS Reference
- JavaScript Reference
- SQL Reference
- Python Reference
- W3.CSS Reference
- Bootstrap Reference
- PHP Reference
- HTML Colors
- jQuery Reference
- Java Reference
- Angular Reference

Web Certificates

- HTML Certificate
- CSS Certificate
- JavaScript Certificate
- SQL Certificate
- Python Certificate
- jQuery Certificate
- PHP Certificate
- Bootstrap Certificate
- XML Certificate

[Get Certified »](#)

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy. Copyright 1999-2020 by Refsnes Data. All Rights Reserved.

Powered by W3.CSS.

