**Scope, If-else Conditionals & Switch Case in JavaScript | Web Development Tutorials #51**

In this tutorial, we will learn about scoping in JavaScript and then we will move on to If-Else and Switch Case conditions. Make a new file as *tut51.html* and add the boilerplate to get the basic HTML template. Then give a title as **Scope and Conditionals** under the <title> tag.

Let us now write the JavaScript under the <script> tag and understand the scope. If we write the following code-

var string1 = "This is a string";

console.log(string1);

Copy

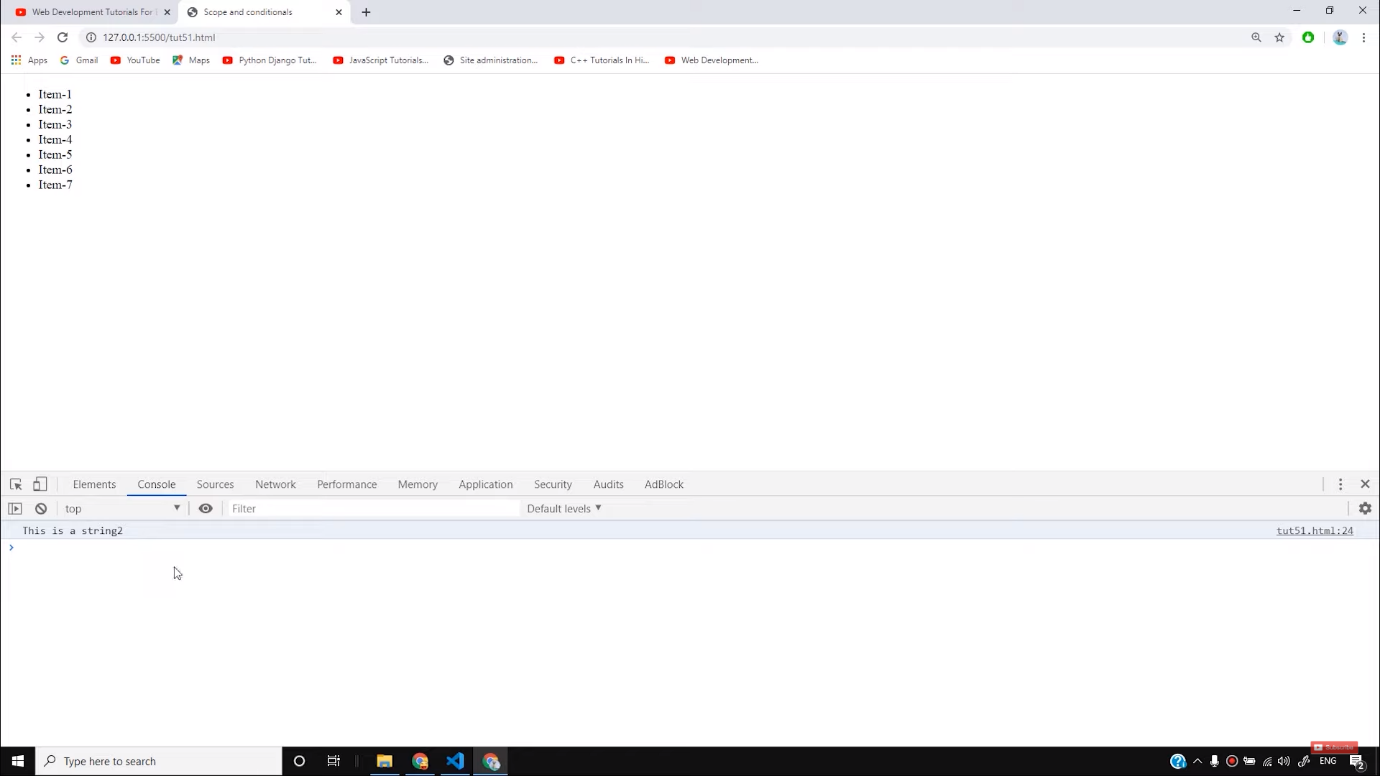
We will expect the output as “This is a string”. But if we update it as-

var string1 = "This is a string2";

console.log(string1);

Copy

Then the output will be as follows-



So here, we have noticed that the value of the variable gets easily changed and it can create a problem if we do not remember the initial value given the variable. It is because the scope of **var** is global here. But if we declare the *var*inside any function, then its scope will remain under that function.

* If we write as follows-

let a = "u";

{

let a = "u6";

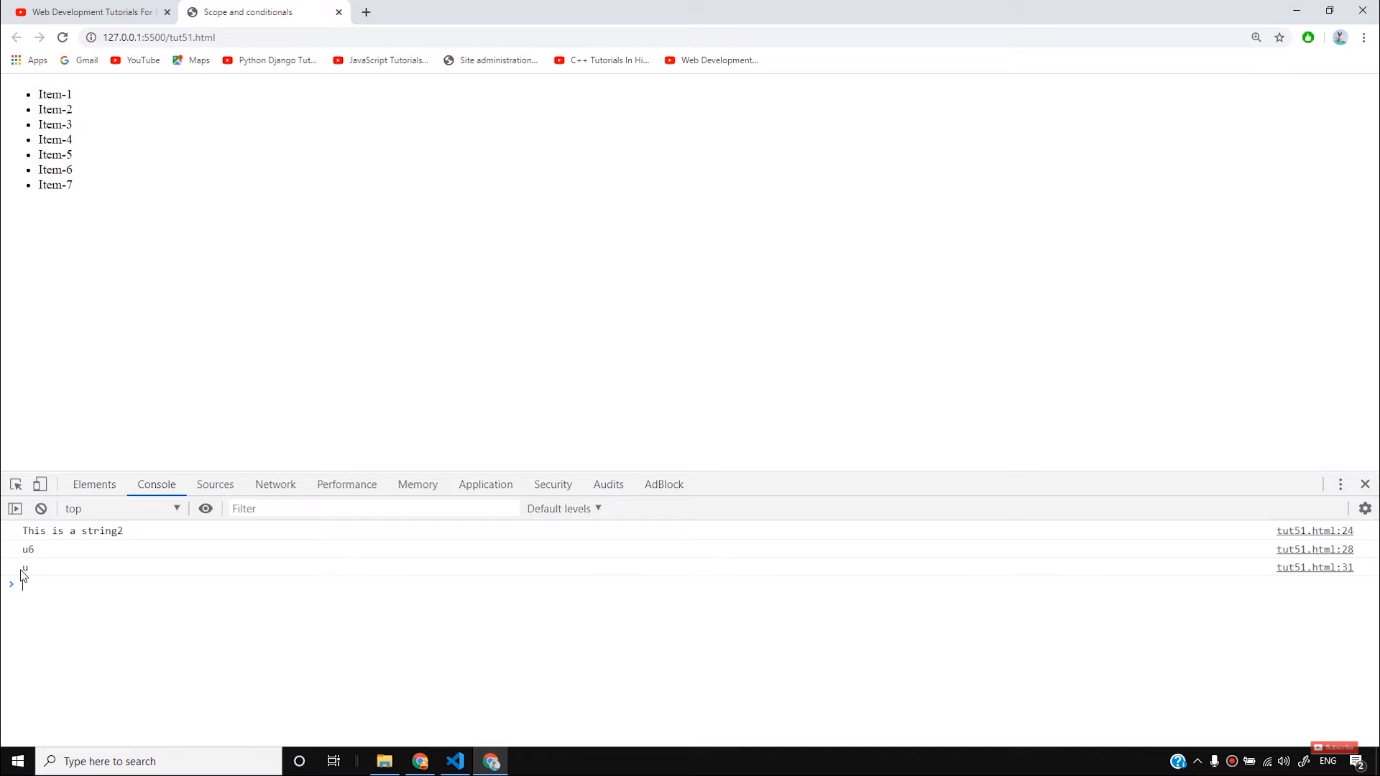
console.log(a)

}

console.log(a)

Copy

In this example, the value of a is ‘u6’ under a particular scope but the value of a declared outside the scope is ‘u’. You will get the output as follows-



However, it is highly recommended to use *let*instead of *var*to avoid confusion of the scope of the variable.

* If we write as follows-

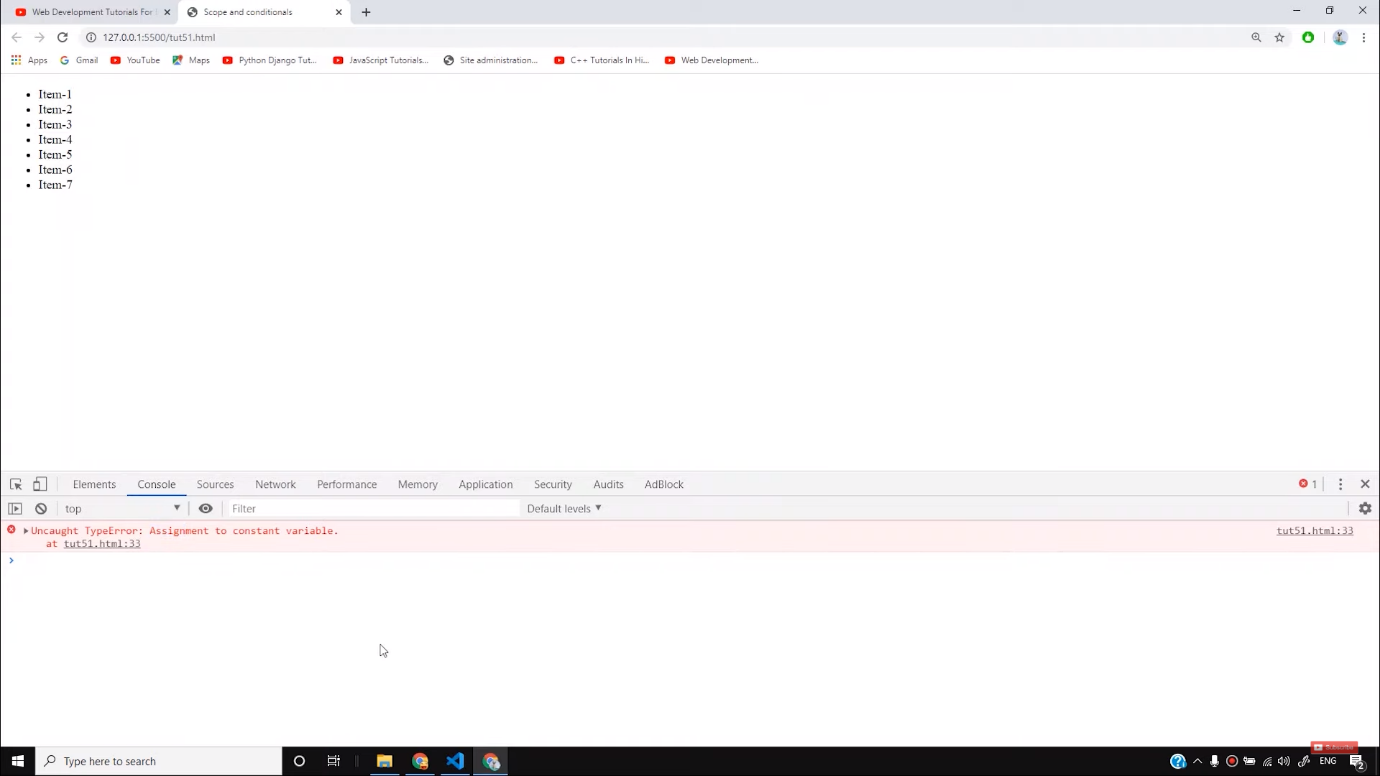
const a = "This cannot be changed";

a = "I want to change this. This cannot be changed";

console.log(a);

Copy

We will see the output as-



In the above example, we have declared **a**as a **constant**, therefore, its value cannot be changed further by any means. Hence, it shows an error.

* Now let us move towards conditional statements. If we write the code as follows-

let age = 5;

if(age>18){

console.log("You can drink water");

}

else if(age==2){

console.log("Age is 2")

}

else if(age==5){

console.log("Age is 5")

}

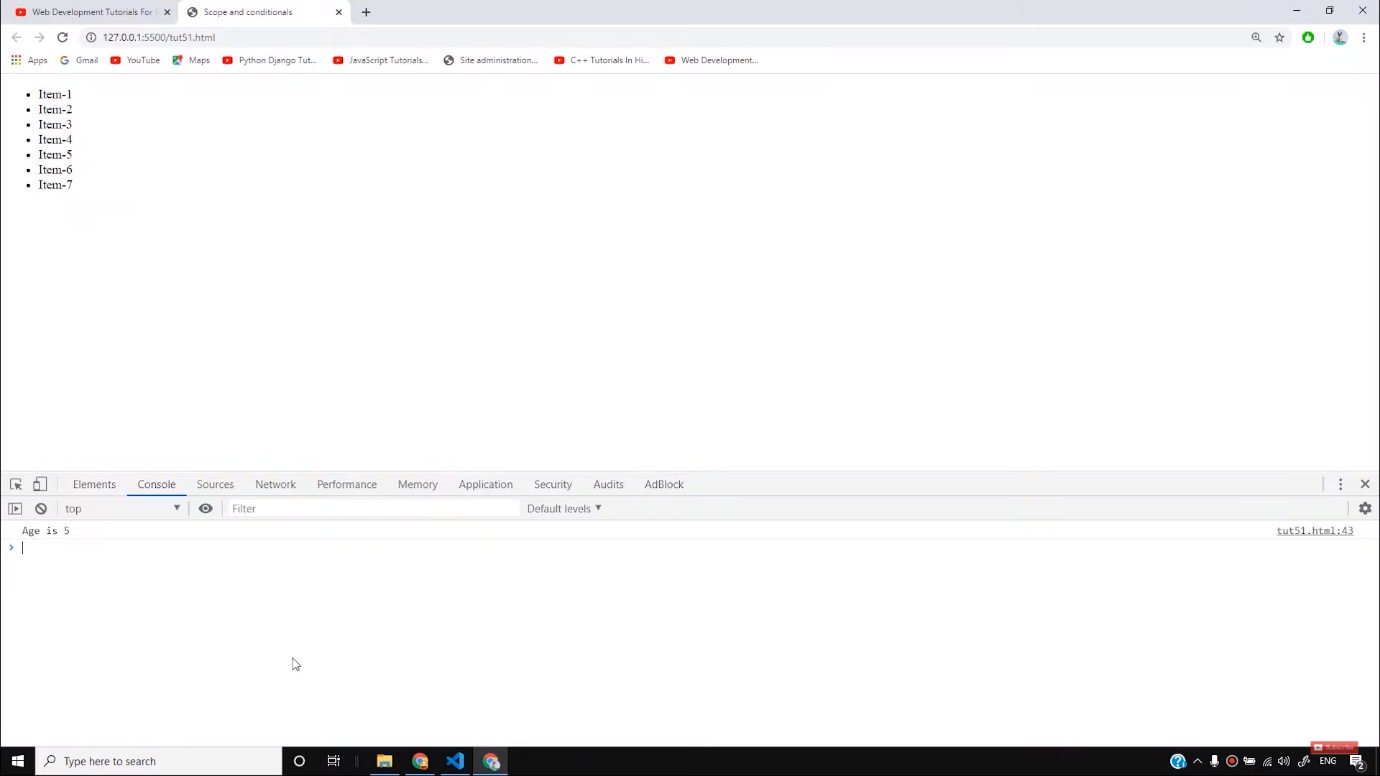
else{

console.log("You can drink Cold Drink");

}

Copy

The output of the above code will be 5 as shown below-



It is because we set the age initially as 5 and it prints the condition which is true for that particular statement. It is known as an **if-else ladder.**

* Let us now understand **switch-case statements.**If we write as follows-

const cups = 41;

switch (cups) {

case 4:

console.log("The value of cups is 4")

break;

case 41:

console.log("The value of cups is 41")

break;

case 42:

console.log("The value of cups is 42")

break;

case 43:

console.log("The value of cups is 43")

break;

default:

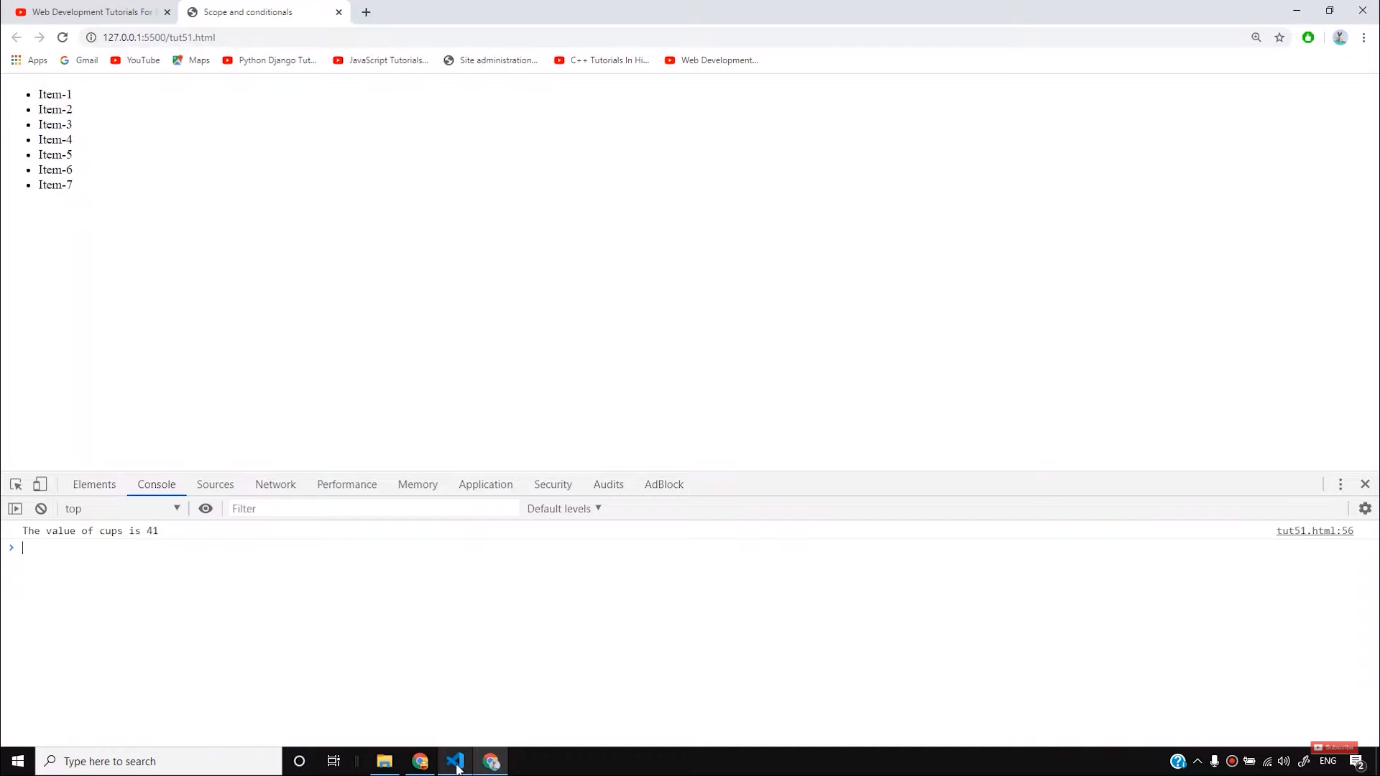
console.log("The value of cups is none of 4, 41, 42, 43")

break;

}

Copy

Here you will see the output as-



It is because the case has matched exactly with the value of *cups.*However, if we remove the break statement then all the values will be printed. And in case, no value matches with any case, then the **default** statement will be printed.

So I believe, you have understood the concepts of scope and conditional statements. In the upcoming videos, we will learn some more advanced concepts in JavaScripts. Till then keep practicing and stay with the tutorials.

**Code as described/written in the video**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Scope and conditionals</title>

</head>

<body>

<div>

<ul>

<li>Item-1</li>

<li>Item-2</li>

<li>Item-3</li>

<li>Item-4</li>

<li>Item-5</li>

<li>Item-6</li>

<li>Item-7</li>

</ul>

</div>

<script>

// var string1 = "This is a string";

// var string1 = "This is a string2";

// console.log(string1);

// let a = "u";

// {

// let a = "u6";

// console.log(a)

// }

// console.log(a)

// const a = "This cannot be changed";

// a = "I want to change this. This cannot be changed";

// console.log(a);

// let age = 5;

// if(age>18){

// console.log("You can drink water");

// }

// else if(age==2){

// console.log("Age is 2")

// }

// else if(age==5){

// console.log("Age is 5")

// }

// else{

// console.log("You can drink Cold Drink");

// }

const cups = 47;

switch (cups) {

case 4:

console.log("The value of cups is 4")

break;

case 41:

console.log("The value of cups is 41")

break;

case 42:

console.log("The value of cups is 42")

break;

case 43:

console.log("The value of cups is 43")

break;

default:

console.log("The value of cups is none of 4, 41, 42, 43")

break;

}

</script>

</body>

</html>

# JavaScript if, else, and else if

[❮ Previous](https://www.w3schools.com/js/js_comparisons.asp)[Next ❯](https://www.w3schools.com/js/js_switch.asp)

Conditional statements are used to perform different actions based on different conditions.

## **Conditional Statements**

Very often when you write code, you want to perform different actions for different decisions.

You can use conditional statements in your code to do this.

In JavaScript we have the following conditional statements:

* Use if to specify a block of code to be executed, if a specified condition is true
* Use else to specify a block of code to be executed, if the same condition is false
* Use else if to specify a new condition to test, if the first condition is false
* Use switch to specify many alternative blocks of code to be executed

The switch statement is described in the next chapter.

## **The if Statement**

Use the if statement to specify a block of JavaScript code to be executed if a condition is true.

### **Syntax**

if (*condition*) {  
  //  block of code to be executed if the condition is true}

Note that if is in lowercase letters. Uppercase letters (If or IF) will generate a JavaScript error.

### **Example**

Make a "Good day" greeting if the hour is less than 18:00:

if (hour < 18) {  
  greeting = "Good day";  
}

The result of greeting will be:

Good day

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_ifthen)

ADVERTISEMENT

## **The else Statement**

Use the else statement to specify a block of code to be executed if the condition is false.

if (*condition*) {  
  //  block of code to be executed if the condition is true} else {  
  //  block of code to be executed if the condition is false}

### **Example**

If the hour is less than 18, create a "Good day" greeting, otherwise "Good evening":

if (hour < 18) {  
  greeting = "Good day";  
} else {  
  greeting = "Good evening";  
}

The result of greeting will be:

Good day

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_ifthenelse)

## **The else if Statement**

Use the else if statement to specify a new condition if the first condition is false.

### **Syntax**

if (*condition1*) {  
  //  block of code to be executed if condition1 is true} else if (*condition2*) {  
  //  block of code to be executed if the condition1 is false and condition2 is true  
} else {  
  //  block of code to be executed if the condition1 is false and condition2 is false}

### **Example**

If time is less than 10:00, create a "Good morning" greeting, if not, but time is less than 20:00, create a "Good day" greeting, otherwise a "Good evening":

if (time < 10) {  
  greeting = "Good morning";  
} else if (time < 20) {  
  greeting = "Good day";  
} else {  
  greeting = "Good evening";  
}

The result of greeting will be:

Good day

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_elseif)

# JavaScript Switch Statement

[❮ Previous](https://www.w3schools.com/js/js_if_else.asp)[Next ❯](https://www.w3schools.com/js/js_loop_for.asp)

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.
* If there is no match, the default code block 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:

Friday

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_switch)

ADVERTISEMENT

## **The break Keyword**

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

This will stop the execution inside the switch 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 »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_switch2)

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 »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_switch4)

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 »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_switch3)

## **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**

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