

# SWITCH COURSE

APRIL 2022

YONATAN BENEZRA

**- DO NOT DISTRIBUTE –**

**- DO NOT COPY – ALL RIGHTS RESERVED TO THE  
AUTHOR –**

The switch statement is used in decision-making.  
It evaluates an expression and executes the corresponding body that matches the expression's result.

The syntax of the switch statement is:

```
switch(variable/expression) {  
    case value1:  
        // body of case 1  
        break;  
  
    case value2:  
        // body of case 2  
        break;  
  
    case valueN:  
        // body of case N  
        break;  
  
    default:  
        // body of default  
}
```

The switch statement evaluates a variable/expression inside parentheses ().

- If the result of the expression is equal to value1, its body is executed.
- If the result of the expression is equal to value2, its body is executed.
- This process goes on. If there is no matching case, the default body executes.

**\*Notes:**

- The break statement is optional. If the break statement is encountered, the switch statement ends.
- If the break statement is not used, the cases after the matching case are executed as well.
- The default clause is also optional.

**Example 1:**

```
// program using switch statement  
let a = 2;  
  
switch (a) {  
  
    case 1:  
        a = 'one';  
        break;  
    case 2:
```

```
        a = 'two';
        break;
    default:
        a = 'not found';
        break;
}
console.log(`The value is ${a}`);
```

The value is two.

In the above program, an expression `a = 2` is evaluated with a switch statement.

- The expression's result is evaluated with case 1, which is false.
- Then, the switch statement goes to the second case. Here, the expression's result matches case 2. As a result, "The value is two" is displayed.
- The break statement terminates the block, and the control flow of the program jumps outside the switch block.

### Example 2:

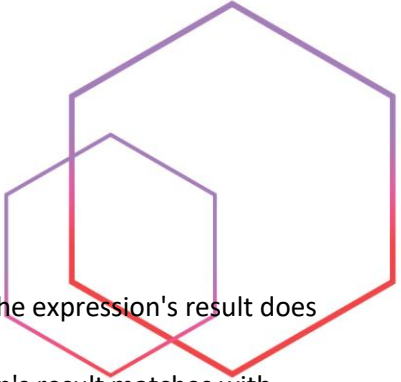
Type Checking in switch Statement

```
// program using switch statement
let a = 1;

switch (a) {
    case "1":
        a = 1;
        break;
    case 1:
        a = 'one';
        break;
    case 2:
        a = 'two';
        break;

    default:
        a = 'not found';
        break;
}
console.log(`The value is ${a}`);
```

The value is one.



In the above program, an expression `a = 1` is evaluated with a switch statement.

- In JavaScript, the switch statement checks the value strictly. Therefore, the expression's result does not match with case `"1"`.
- Then, the switch statement goes to the second case. Here, the expression's result matches with case `1`. As a result, "The value is one" is displayed.
- The break statement terminates the block, and control flow of the program jumps to outside of the switch block.


**\*Note:**

In JavaScript, the switch statement checks the cases strictly (should be of the same data type) with the expression's result. Notice in the above example, `1` does not match with `"1"`.

Let's write a program to make a simple calculator using the switch statement.

**Exercise:**

Create a simple calculator using the switch statement



## JavaScript switch With Multiple Case

In a JavaScript switch statement, cases can be grouped to share the same code.

### Example:

switch With Multiple Case

```
// multiple case switch program
let fruit = 'apple';
switch(fruit) {
  case 'apple':
  case 'mango':
  case 'pineapple':
    console.log(`${fruit} is a fruit.`);
    break;
  default:
    console.log(`${fruit} is not a fruit.`);
    break;
}
```

apple is a fruit.

In the above program, multiple cases are grouped. All the grouped cases share the same code.

If the value of the fruit variable held a value of mango or pineapple, the output would have been the same.