Switch

Justin Besteman

December 30, 2016

Synopsis

Switch are powerful tool for a coder to test many conditions

Why

If a coder is testing only a few conditions then a if, else if, else if are perfect tools to use.
But want if there are vast number of things to test. A coder could still use those tools mentioned before but you better get some ice to cool off the fingers when you are done.
A switch statement is way to reduce the amount code that needs to be typed
Also, the code is easy to understand at a glance

Syntax

The switch statement is structured like a train track system where the rails switch (you see what I did there) base on a certian case is achieve.
With this tool, you have a switch and cases that will be excuted based on the condition of the switch.
After each case, you must use the keywork break for if you don't bad things happen to good people.
A switch statement is block level code so without the break it excuted all the condition after the condition that is met.

Advanced

Switch statement have the ability to "Stack case on top of each other" (My mentor, Sean, taught me that)
Meaning you can have more then one condition or case that can be met for that code to excute.
At the end of the switch statement, a coder may put the keyword "default" This will be triggered if none of the condition are met.

Examples

```
1
2  // ----------------- Example 1 ------------------
3
4  var traffic_light = "green";
5
6  switch (traffic_light) {
7
8    case "green":
9      console.log("Go");
10     break;
11   case "yellow":
12     console.log("Speed up or slow down");
13     break;
14   case "red":
15     console.log("Stop, Stop, Stoopppp!!!");
16     break;
17
18 } // End of Switch Statement
19
20 // ----------------- Example 2 ------------------
21
22 var user_pick = 3;
23
24 switch (user_pick) {
25
26   case 1:
27     console.log("User 1");
28     break;
29   case 2:
30     console.log("User 2");
31     break;
```

```
  case 3:
    console.log("User 3");
    break;
  case 4:
    console.log("User 4");
    break;
  case 5:
    console.log("User 5");
    break;

} // End of Switch Statement

// ----------------- Example 3 ------------------

// This shows how cases can be stacked

var user_pick = 3;

switch (user_pick) {

  case "one":
  case 1:
    console.log("User 1");
    break;
  case "two":
  case 2:
    console.log("User 2");
    break;
  case "three":
  case 3:
    console.log("User 3");
    break;
  case "four":
  case 4:
    console.log("User 4");
    break;
  case "five":
  case 5:
    console.log("User 5");
    break;

} // End of Switch Statement

// So this is powerful because if you allow the,
// user enter their choices and if they entered in,
```

```
77  // a integers or string it will still work
78
79  // ----------------- Example 4 ------------------
80
81  // This shows the default keyword
82
83  var user_pick = 3;
84
85  switch (user_pick) {
86
87    case "one":
88    case 1:
89      console.log("User 1");
90      break;
91    case "two":
92    case 2:
93      console.log("User 2");
94      break;
95    case "three":
96    case 3:
97      console.log("User 3");
98      break;
99    case "four":
100   case 4:
101     console.log("User 4");
102     break;
103   case "five":
104   case 5:
105     console.log("User 5");
106     break;
107   default:
108     console.log("Invalid input");
109     break;
110
111 } // End of Switch Statement
112
113 // Very awesome here
114 // With this default statement you test,
115 // The user input and say if it is not valid
```