

# Starting Electronics

Electronics for Beginners and Beyond

**Starting Electronics needs your help!** Please make a [donation](#) to help cover our hosting and other costs. Click the donate button to send a donation of any amount.

Donate



[Home](#) ▶ [Software](#) ▶ [Arduino](#) ▶ [Learn To Program Course](#) ▶ 13 Switch Break

## Arduino *switch* and *break* Statements

Created on: 31 December 2014

### Part 13 of the Arduino Programming Course

The *switch* statement is similar to using *if* with multiple *else-if* constructs. *switch* is used in conjunction with *break* which will also be explained in this part of the course.

Using *switch* instead of multiple *else-if* constructs is easier to read and has more flexibility.

### Arduino Programming Course



Software



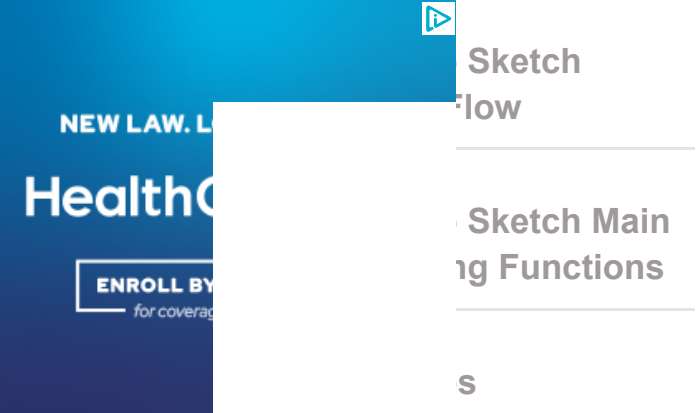
Arduino Software



Arduino Programming Course



Introduction and Requirements



## ***switch* Statement Example**

The following Arduino sketch shows the *switch* statement being used in conjunction with the *break* statement.

Load the sketch to the Arduino and then start the Serial Monitor window. Sending 1 from the serial monitor window to the Arduino will switch the on-board LED on and sending 2 will switch the LED off.

Sending 3 will show the menu of options that the sketch operates on. Sending any other character will bring up a default message showing that the option chosen is invalid.

 **Part 4: Arithmetic Operators**

 **Part 5: Relational Operators**

 **Part 6: Increment Operator and Commenting**

 **Part 7: The Arduino for Loop**

 **Part 8: The Arduino while Loop**

 **Part 9: The Arduino if Statement**

 **Part 10: Making Decisions with if-else**

 **Part 11: Decisions with if-else-if**

```

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT); // LED on pin 13 of UNO
}

char rx_byte = 0;

void loop() {
  if (Serial.available() > 0) { // is a character av
    rx_byte = Serial.read();


    switch (rx_byte) {
      case '1':
        digitalWrite(13, HIGH);
        Serial.println("LED is ON");
        break;

      case '2':
        digitalWrite(13, LOW);
        Serial.println("LED is OFF");
        break;

      case '3':
        Serial.println("----- MENU -----");
        Serial.println("1. Switch LED on.");
        Serial.println("2. Switch LED off.");
        Serial.println("3. This menu.");
    }
  }
}

```

 [Part 12: Logical Operators](#)

 [Part 13: Switch & Break](#)

 [Part 14: Conditional Operator](#)

 [Part 15: Functions](#)

 [Part 16: Returning a Value from a Function](#)

 [Part 17: Arrays](#)

 [Part 18: Strings](#)

 [Part 19: Serial Input](#)

How the sketch works will be explained later on this page, but first we must look at the structure of the *switch* statement and how the *break* statement works.

This video shows the above sketch in operation.

## Arduino switch statement - Part 13 of the Arduino Progr...



Can't see the video? [View on YouTube →](#)



On cat scratchers

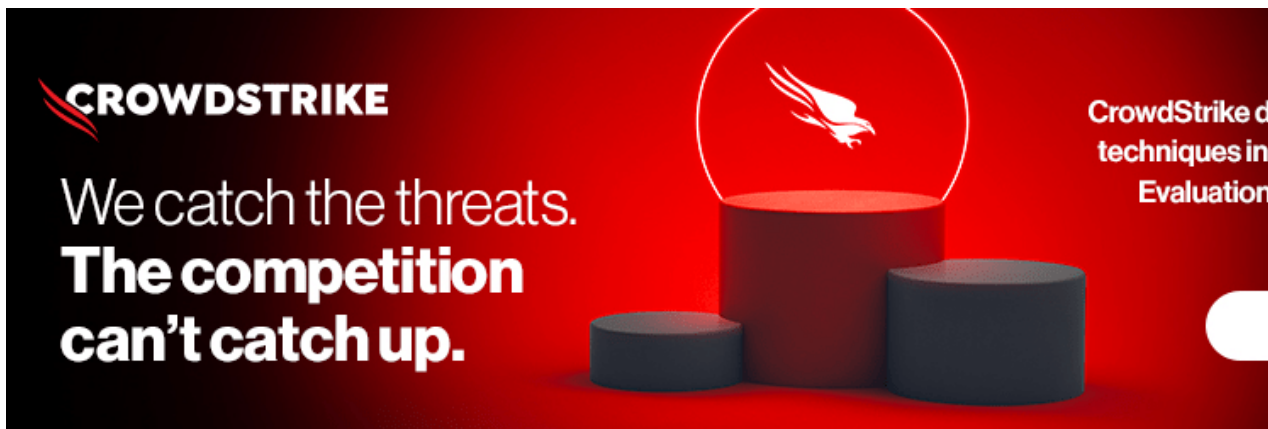
Shop now

As an Amazon Associate I earn from qualifying purchases:

[Amazon Website](#)

[Computer Best Sellers](#)

Shop Now



Raspberry Pi 4 Model B 2019 Quad Core 8GB 64-bit WiFi 6E

~~\$167.00~~

★★★★★  
(16538)



Raspberry Pi 4 Model B (8gb)

\$219.90

★★★★★ (438)

## Structure of the *switch* Statement

The image below shows the structure of a switch statement.

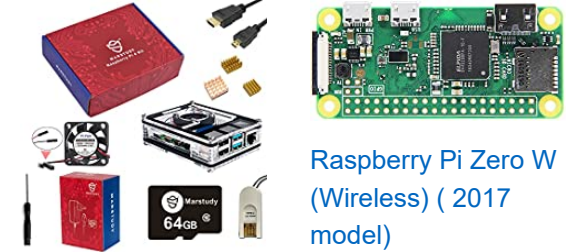
```
switch keyword  Switch variable - can be char or int
switch (switch_var) { ← Opening brace of switch body
  case '1': ← int or char constant to check for
    // statements placed here run if switch_var == '1'
    break;
  case '2':
    // statements placed here run if switch_var == '2'
    break;
  default:
    // statements placed here run if if no case found
    break;
} ← Closing brace of switch body
```

If case matches, statements are run followed by break which breaks out of the switch body

### Structure of an Arduino *switch* Statement

The *switch* statement has a variable (**switch\_var** in the above image or **rx\_byte** in the example sketch) which can be an integer (int) or character (char) variable.

The switch variable will be tested against the value in each case to see if they match. When a case is found that matches, the statements below the case will be run until the *break* keyword is reached. This will break the program flow out of the body of the *switch* statement and execution of the sketch will continue below the closing brace of the *switch* statement.



**MAPSTUDY**  
Raspberry Pi 4B  
Starter kit (64GB  
Edition) / ...  
\$64.99 ✓prime  
★★★★★ (9)

Raspberry Pi Zero W  
(Wireless) ( 2017  
model)  
\$68.99 ✓prime  
★★★★★ (1270)

Ads by Amazon

### Shop Related Products



MOGAOPI Electronic  
Component Kit &  
Total 139...  
\$25.99 ✓prime  
★★★★★ (224)



ELEGOO Mega R3  
Project The Most  
Complete Ultim...  
\$65.99 ✓prime  
★★★★★ (6205)



Elenco Snap Circuits  
\$24.99 \$36.05  
✓prime  
Exploratio...  
★★★★★  
(14924)



MiOYOOW DIY  
Bluetooth-  
Compatible Speaker  
\$28.08 ✓prime  
★★★★★ (228)

If no matching case is found, then the code under the *default* keyword will be run until its *break* statement is found.

## How the Example Sketch Works

---

In the example sketch, the *switch* statement is placed inside an *if* statement in the main loop. The *switch* statement will then only run if a new character is received from the Serial Monitor window.

When a character is received from the Serial Monitor window, the *switch* statement will check for a matching case value. If the character '1' is received, then the LED is switched on and a message displayed in the Serial Monitor window.

If '2' is received, the LED is switched off. '3' displays a menu of the options available in the sketch.

If any character is sent that does not match the characters in any of the *case* statements, then the code in the *default* part of the *switch* body is run which displays a default message.

## The *break* Statement

---

The *break* statement is used in the example sketch to break out of the body of the *switch* statement.

*break* can also be used to break out of any loop such as a *while* or *for* loop. As an example, a certain condition can be tested for in a loop using an *if* statement and if the statement evaluates to true, the *break* statement can be run to break out of the loop.



Part 12:

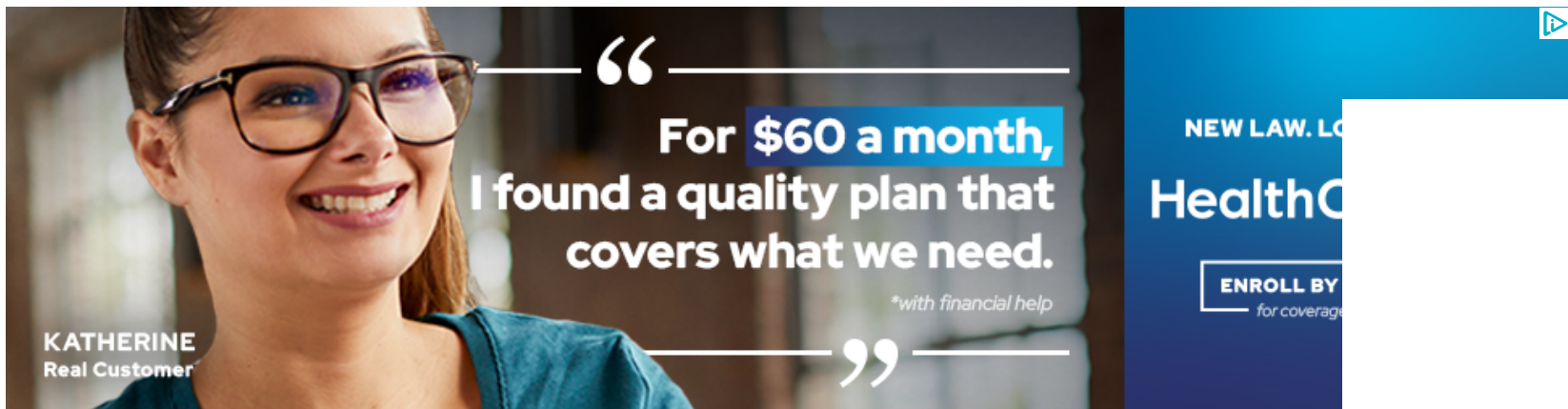
Logical Operators

Part 14:

Conditional Operator







## Comments



There are no comments posted yet. [Be the first one!](#)

## Post a new comment

Enter text right here!

Comment as a Guest, or login:   [WORDPRESS.COM](#)

Name

*Displayed next to your comments.*

Email

*Not displayed publicly.*

Website (optional)

*If you have a website, link to it here.*

Subscribe to

**Submit Comment**

[Arduino](#)

[Pinout](#)

[About](#)

[Contact](#)

[Donate](#)

[Privacy Policy](#)



