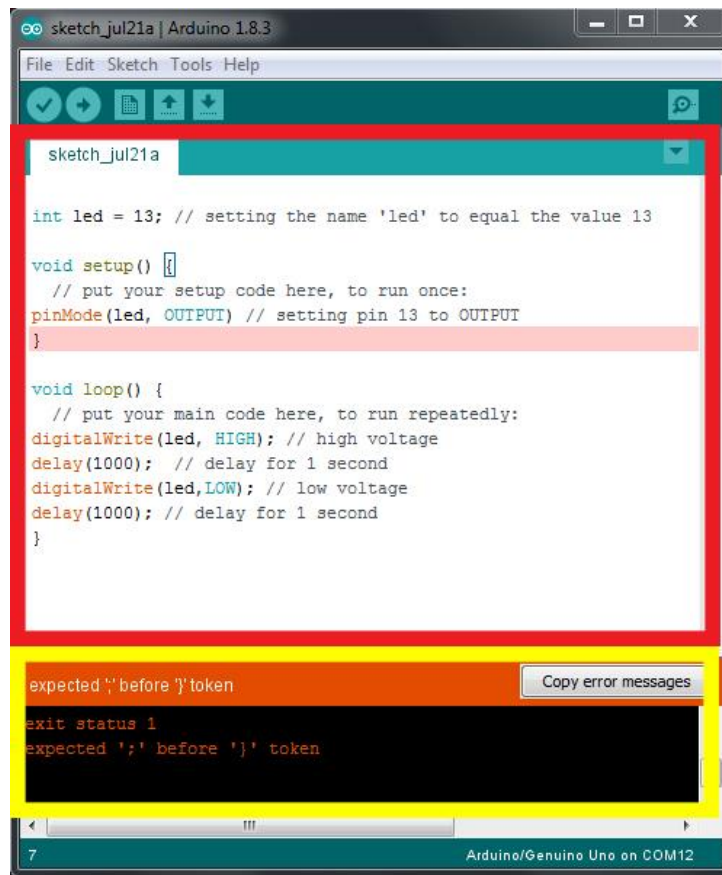


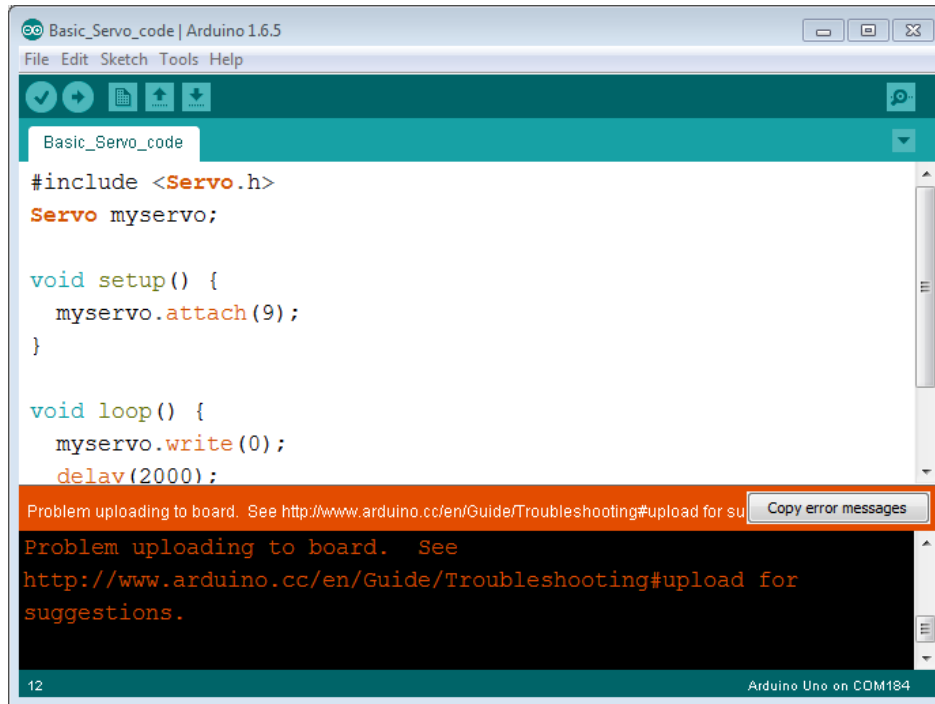
We have identified common errors encountered when working with the Arduino software. Below are 6 common issues, along with how to fix the problem. For more information please visit the Arduino Troubleshooting Page at <https://www.arduino.cc/en/guide/troubleshooting>

Issue 1: How do I read an error message?



In the above image, we have drawn a *Red Box* to show where the code is written. Any line of code with an error is highlighted in pink. We have also drawn a *Yellow Box* to show where to find the exact location of the error. This section has an orange bar which only turns orange when there is an error in compiling and is a short one-liner description of the error; and, in the black box below it, there is a more detailed explanation of the error.

Issue 2: Problem Uploading to the board



```
Basic_Servo_code | Arduino 1.6.5
File Edit Sketch Tools Help
Basic_Servo_code
#include <Servo.h>
Servo myservo;

void setup() {
  myservo.attach(9);
}

void loop() {
  myservo.write(0);
  delay(2000);
}

Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#upload for suggestions.
Copy error messages

12 Arduino Uno on COM184
```

What's the problem?

This error shows that there is a connection problem with the Arduino. Connection problems usually happen because you haven't selected the right port in your Arduino software or because your USB cable isn't connected.

Connection problems arise because:

1. USB cable is not connected properly.
2. The right port for the Arduino is not selected.

How do I fix it?

- Make sure that the USB cable is completely plugged into the Arduino and the computer.
- Make sure the Arduino software is connected to the correct communication port (COM#)?
 - a) Go to Tools>Port
 - b) Choose the port (COM#) with "Arduino Uno" next to it
 - c) If there is more than one Port with the "Arduino Uno", select the port with the highest number (each port is labeled with a number).
 - d) Some versions of the Arduino software do NOT write "Arduino Uno" next to the COM#.
 - e) In that case, unplug the USB cable and plug it back in.
 - f) If still none of the ports have "Arduino Uno", use another USB cable, or get another Arduino board, restart the computer, or ALL the above.

Issue 3: Expected ';' before '____'

```
sketch_jan07a | Arduino 1.6.5
File Edit Sketch Tools Help
sketch_jan07a $
myservo.write(0);
delay(2000);
myservo.write(180);
delay(2000);
expected ';' before 'delay'
';' before 'delay'
expected ';' before 'delay'
```

What's the problem?

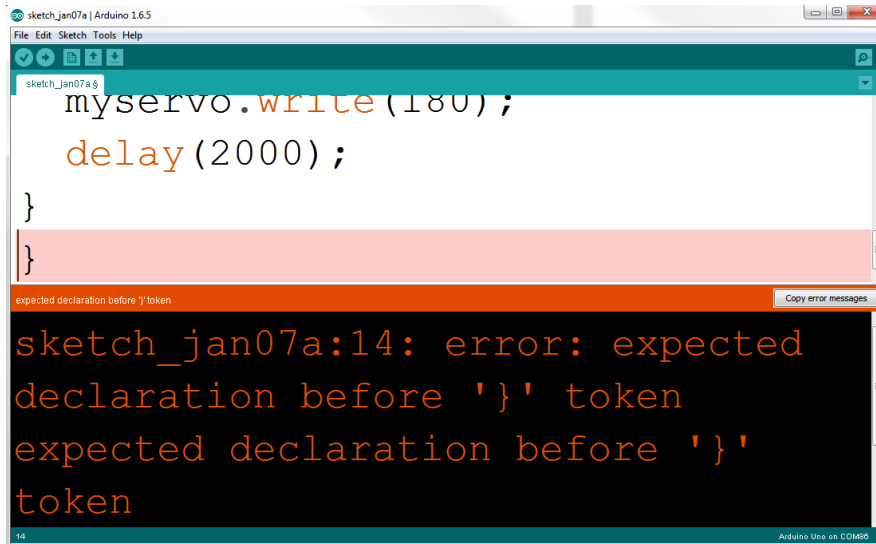
The Arduino programming language needs each command to end with a semicolon (;). This error states that a semicolon is missing from one of the lines of code in the program.

How do I fix it?

- Add the missing semicolon where appropriate.
- The Arduino programming software highlights the general area with the error in the code. In the picture above, the "delay(2000);" is highlighted in pink. This is explained in the black box as "';' before 'delay'." It means the line "myservo.write(180)" is missing a semicolon.
- To fix this error in the example above, add a semicolon to the end of the line that says "myservo.write(180)".

NOTE: If semicolons is missing on multiple lines, you might get this error more than once.

Issue 4: Expected declaration before '}' token



```
sketch_jan07a | Arduino 1.6.5
File Edit Sketch Tools Help
sketch_jan07a.g
myservo.write(180);
delay(2000);
}
}
sketch_jan07a:14: error: expected
declaration before '}' token
expected declaration before '}'
token
```

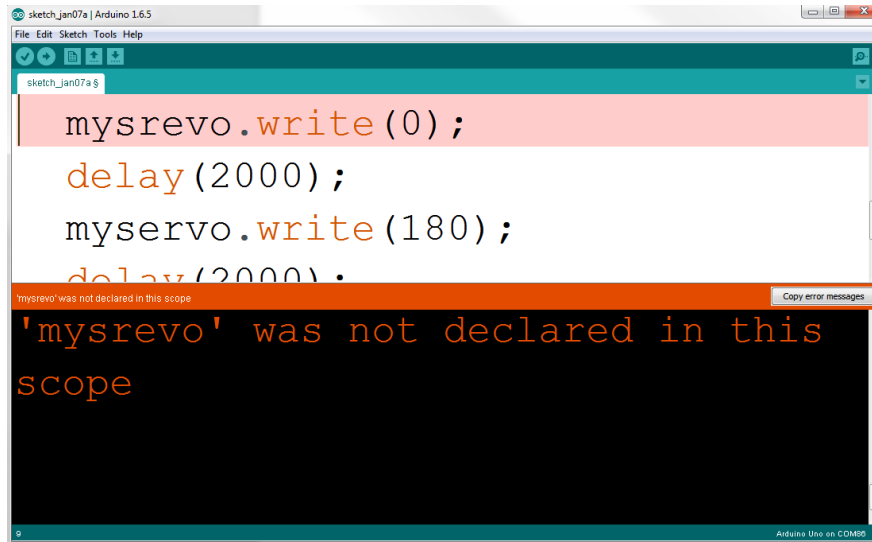
What's the problem?

The Arduino programming language uses curly brackets (also called “braces”), to mark the beginning and the end of sections of code. Curly brackets always need to come in a pair. Open curly bracket “{” at the beginning of the section of code within a function, and close curly bracket “}” at the end. For example, curly brackets are used to tell the computer where the setup() code starts and where it ends. The above error happens when one of those curly brackets is missing, or there is an extra curly bracket in the code.

How do I fix it?

- Look through the code and find the matching brackets (e.g., make sure that one of { bracket matches with one of } bracket).
- Delete extra brackets, or add the ones missing.
- In the example above, there are two closed brackets at the end of the code. One of those brackets is extra and doesn't have a matching bracket, so deleting it will fix the error.

Issue 5: '_____' was not declared in this scope



```
sketch_jan07a | Arduino 1.6.5
File Edit Sketch Tools Help
sketch_jan07a.g
mysrevo.write(0);
delay(2000);
myservo.write(180);
delay(2000);
'mysrevo' was not declared in this scope
Copy error messages
'mysrevo' was not declared in this scope
Arduino Uno on COM5
```

What's the problem?

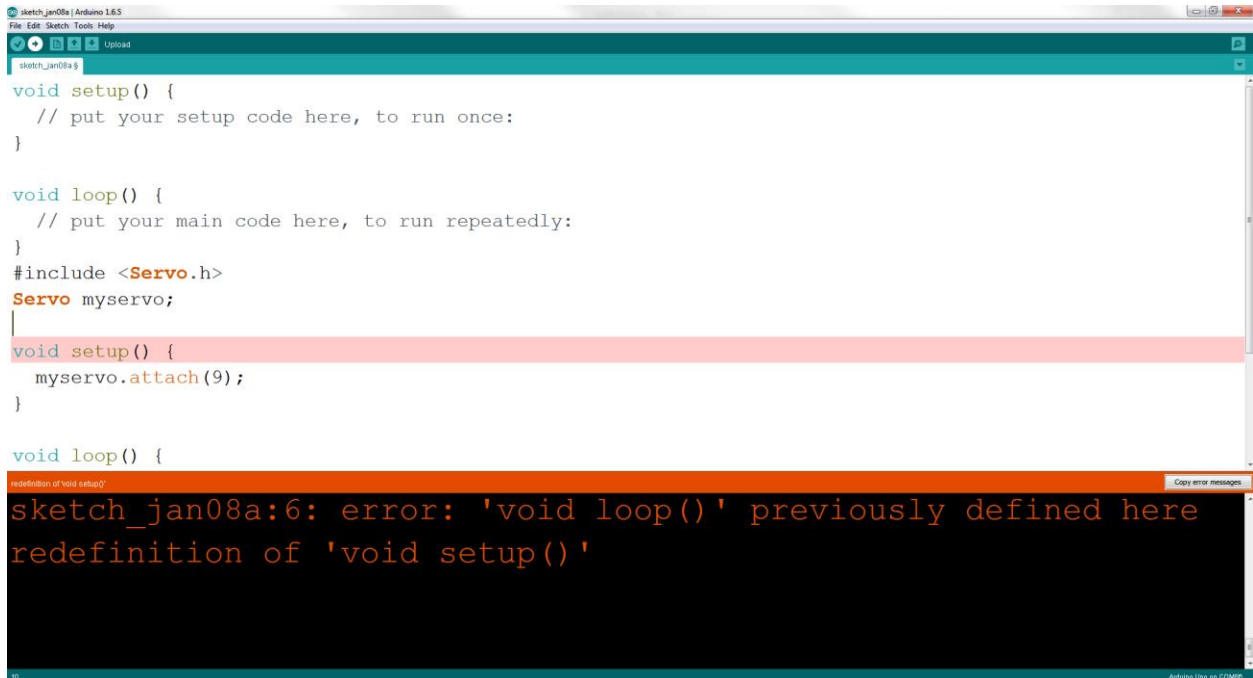
This error happens because a variable is not declared, or is misspelled. Therefore, the variable is seen by the computer as a new word. When the computer sees this new word that is not defined, it does not know what to do with it.

For Arduino software to understand what to do with a variable, the variable must be "declared". "Declaring a variable means to tell the computer the name of a variable and the type of data it holds (for example, writing "int myNumber" declares that a variable called "myNumber" will hold integers and writing "Servo myservo" declares that "myservo" will reference the Servo library).

How do I fix it?

- Make sure all variables and names are declared.
- Make sure all spelling is correct.
- If words like 'delay' and 'write' didn't turn colors, it means that they are misspelled or the cases don't match.
 - Keep in mind that the Arduino programming software is case sensitive which means reads "variable" differently than "VaRiAbLe" as capital letters are different characters than lower case letters.
- In the example above, "mysrevo" was spelled incorrectly.

Issue 6: Redefinition of 'void setup()' or 'void loop()'



```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}  
#include <Servo.h>  
Servo myservo;  
void setup() {  
  myservo.attach(9);  
}  
  
void loop() {  
  redefinition of 'void setup()'  
}
```

sketch_jan08a:6: error: 'void loop()' previously defined here
redefinition of 'void setup()'

What's the problem?

Every Arduino program has three main parts. The beginning of the program (where we declare the variables), the void setup() function (which runs once) and the void loop() function (which runs over and over). We can ONLY have one void setup and one void loop in the code. This error happens when you make more than one void setup() or void loop() functions.

How do I fix it?

- Since we can only have one void setup and one void loop in the code, this error usually happens when people try to copy and paste the entire code from one sketch to another.
- In the example above, notice that the first "void setup()", and "void loop()" are empty. Because they are empty, they can be deleted.