scope

[Variable Scope & Qualifiers]

Description

Variables in the C++ programming language, which Arduino uses, have a property called scope. This is in contrast to early versions of languages such as BASIC where every variable is a *global* variable.

A global variable is one that can be seen by every function in a program. Local variables are only visible to the function in which they are declared. In the Arduino environment, any variable declared outside of a function (e.g. [setup()](https://www.arduino.cc/reference/en/language/structure/sketch/setup), [loop()](https://www.arduino.cc/reference/en/language/structure/sketch/loop), etc. ), is a *global* variable.

When programs start to get larger and more complex, local variables are a useful way to insure that only one function has access to its own variables. This prevents programming errors when one function inadvertently modifies variables used by another function.

It is also sometimes handy to declare and initialize a variable inside a [for](https://www.arduino.cc/reference/en/language/structure/control-structure/for) loop. This creates a variable that can only be accessed from inside the [for](https://www.arduino.cc/reference/en/language/structure/control-structure/for)-loop brackets.

Example Code

int gPWMval; // any function will see this variable

void setup() {

// ...

}

void loop() {

int i; // "i" is only "visible" inside of "loop"

float f; // "f" is only "visible" inside of "loop"

// ...

for (int j = 0; j < 100; j++) {

// variable j can only be accessed inside the for-loop brackets

}

}