static

[Variable Scope & Qualifiers]

Description

The static keyword is used to create variables that are visible to only one function. However unlike local variables that get created and destroyed every time a function is called, static variables persist beyond the function call, preserving their data between function calls.

Variables declared as static will only be created and initialized the first time a function is called.

Example Code

/\* RandomWalk

Paul Badger 2007

RandomWalk wanders up and down randomly between two

endpoints. The maximum move in one loop is governed by

the parameter "stepsize".

A static variable is moved up and down a random amount.

This technique is also known as "pink noise" and "drunken walk".

\*/

#define randomWalkLowRange -20

#define randomWalkHighRange 20

int stepsize;

int thisTime;

void setup() {

Serial.begin(9600);

}

void loop() {

// test randomWalk function

stepsize = 5;

thisTime = randomWalk(stepsize);

Serial.println(thisTime);

delay(10);

}

int randomWalk(int moveSize) {

static int place; // variable to store value in random walk - declared static so that it stores

// values in between function calls, but no other functions can change its value

place = place + (random(-moveSize, moveSize + 1));

if (place < randomWalkLowRange) { // check lower and upper limits

place = randomWalkLowRange + (randomWalkLowRange - place); // reflect number back in positive direction

}

else if (place > randomWalkHighRange) {

place = randomWalkHighRange - (place - randomWalkHighRange); // reflect number back in negative direction

}

return place;

}