for

[Control Structure]

Description

The for statement is used to repeat a block of statements enclosed in curly braces. An increment counter is usually used to increment and terminate the loop. The for statement is useful for any repetitive operation, and is often used in combination with arrays to operate on collections of data/pins.

Syntax

for (initialization; condition; increment) {

// statement(s);

}

Parameters

initialization: happens first and exactly once.  
condition: each time through the loop, condition is tested; if it’s [true](https://www.arduino.cc/reference/en/language/variables/constants/constants), the statement block, and the **increment** is executed, then the **condition** is tested again. When the **condition** becomes [false](https://www.arduino.cc/reference/en/language/variables/constants/constants), the loop ends.  
increment: executed each time through the loop when condition is [true](https://www.arduino.cc/reference/en/language/variables/constants/constants).

Example Code

// Dim an LED using a PWM pin

int PWMpin = 10; // LED in series with 470 ohm resistor on pin 10

void setup() {

// no setup needed

}

void loop() {

for (int i = 0; i <= 255; i++) {

analogWrite(PWMpin, i);

delay(10);

}

}

Notes and Warnings

The C++ for loop is much more flexible than for loops found in some other computer languages, including BASIC. Any or all of the three header elements may be omitted, although the semicolons are required. Also the statements for initialization, condition, and increment can be any valid C++ statements with unrelated variables, and use any C++ datatypes including floats. These types of unusual for statements may provide solutions to some rare programming problems.

For example, using a multiplication in the increment line will generate a logarithmic progression:

for (int x = 2; x < 100; x = x \* 1.5) {

println(x);

}

Generates: 2,3,4,6,9,13,19,28,42,63,94

Another example, fade an LED up and down with one for loop:

void loop() {

int x = 1;

for (int i = 0; i > -1; i = i + x) {

analogWrite(PWMpin, i);

if (i == 255) {

x = -1; // switch direction at peak

}

delay(10);

}

}