

# For Loops in Arduino

ME 120

Mechanical and Materials Engineering

Portland State University

<http://web.cecs.pdx.edu/~me120>

# Motivation

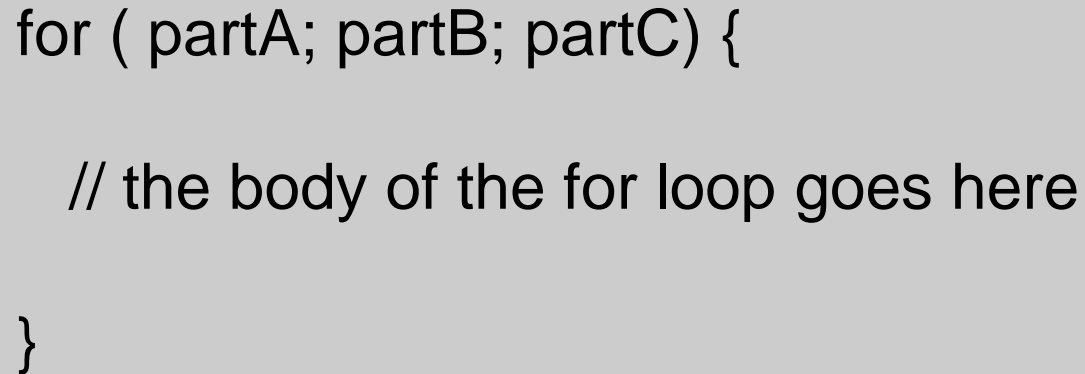
- “for” loops are essential code constructs
  - ❖ Allow for repetition of code blocks
  - ❖ Specify the number of repetitions in advance
  - ❖ Loop counters can be used as indices in arrays
- “for” loops are used to sweep a servo back and forth
  - ❖ Key component in desktop fan project

*A typographical convention:*

We will write “for” and “while” and “loop” in quotation marks when we are referring to the for-loop structure, the while-loop structure and the required loop function in Arduino sketches. This excessive use of quotes is generally considered bad form. We risk offending in order to distinguish “for” as a code construct and the ordinary word for as regular preposition.

# Basic “for” loop structure

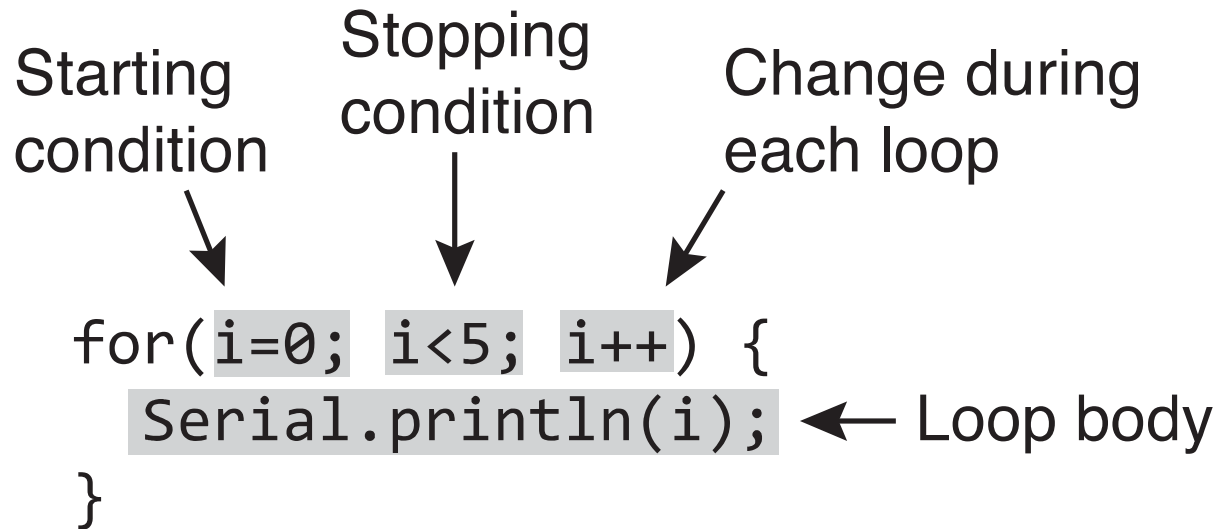
Index controls



```
for ( partA; partB; partC) {  
  
    // the body of the for loop goes here  
  
}
```

- Index controls consist of three parts
  - ❖ partA: Starting condition: the initial value of the loop variable
  - ❖ partB: Stopping condition: a logical test that must be true for the loop to execute
  - ❖ PartC: Increment/decrement: the rule for changing the loop variable
- The parts are separated by a “;” instead of the “,” we were getting used to in our built-in functions)

# Example



- `i=0` is the starting condition. It establishes the value of `i` on the first (and only the first) trip through the loop
- `i<5` must be true for the loop body to be executed
- `i++` means that the value of `i` is incremented by one *after* the loop body is executed.

# Common increment expressions

Expression	Meaning
<code>i++</code>	Increment by 1 and returns the old value of i
<code>++i</code>	Increment by 1 and returns the new value of i
<code>i--</code>	Decrement by 1 and returns the old value of i
<code>--i</code>	Decrement by 1 and returns the new value of i
<code>i+=1</code>	Same as <code>i++</code>
<code>i-=1</code>	Same as <code>i--</code>
<code>i+=2</code>	Increment by 2 and returns the old value of i
<code>i-=2</code>	Decrement by 2 and returns the old value of i

## Examples:

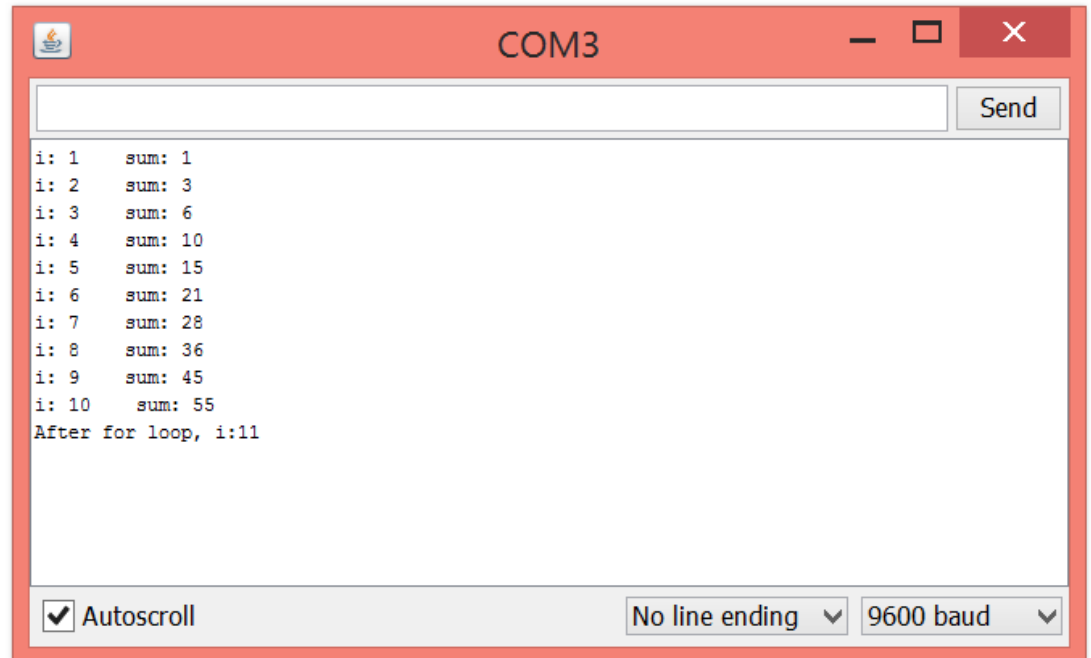
```
x = 2;  
y = ++x;    // x now contains 3, y contains 3  
y = x--;    // x contains 2 again, y still contains 3
```

# Example: add up 10 integers

$$\text{Compute: } s = \sum_{i=1}^{10} i$$



```
void setup () {  
  Serial.begin(9600);  
  
  int i, sum;  
  
  sum = 0;  
  for ( i=1; i<=10; i++) {  
    sum = sum + i;  
  
    Serial.print ("i: ");  
    Serial.print (i);  
  
    Serial.print ("    sum: ");  
    Serial.println (sum);  
  }  
  Serial.print ("After for loop, i:");  
  Serial.println (i);  
}  
  
void loop () {  
}
```



# Loop 1: What does this code do?

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int i = 0;  
  i = i + 1;  
  Serial.println(i);  
}
```

# Loop 1: What does this code do?

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int i = 0;  
  i = i + 1;  
  Serial.println(i);  
}
```



## Loop 2: What does this code do?

```
int i = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  i = i + 1;
  Serial.println(i);
}
```

## Loop 3: What does this code do?

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int i;  
  for ( i=0; i<5; i++ ) {  
    Serial.println(i);  
  }  
}
```

# Loop 4: What does this code do?

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int i;  
  for ( i=0; i<5; i+=2 ) {  
    Serial.println(i);  
  }  
}
```

# Loop 5: What does this code do?

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int i;  
  for ( i=0; i<10; i++) {  
    i = 5;  
    Serial.println(i);  
  }  
  Serial.println("for loop over\n");  
}
```

# Loop 6: What does this code do?

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int i;  
  for ( i=0; i<10; i++) {  
    Serial.println(i);  
    delay(100);  
  }  
  Serial.println("for loop over\n");  
}
```