

## For loops:

```
for (i = 0; i < 10; i++) {...
```

In C, the **for** keyword is followed by a set of parentheses containing three parts separated by semicolons.

```
for (init; test; update)
```

The intent of C's **for** loop is to enable stepping a variable through a series of numbers, like counting from 0 to 9. The part before the first semicolon (*init*) is performed as soon as the **for** statement is reached; it is for initializing the variable that will count. The part between the two semicolons (*test*) is evaluated before each iteration to determine whether the iteration should be repeated. And the part following the final semicolon (*update*) is evaluated at the end of each iteration to update the counting variable for the following iteration.

In practice, **for** loops are used most often for counting out  $n$  iterations. The standard idiom for this is the following.

```
for (i = 0; i < n; i++) {  
    body  
}
```

Here we have a counter variable *i* whose value starts at 0. With each iteration, we test whether *i* has reached  $n$  or not; and if it hasn't, then we execute the **for** statement's body and then perform the *i++* update so that *i* goes to the following integer. The result is that the body is executed for each value of *i* from 0 up to  $n - 1$ .

But you can use a **for** loop for other purposes, too. In the following example, we display the powers of 2 up to 512. Notice how the update portion of the **for** statement has changed to "*p \*= 2*".

```
for (p = 1; p <= 512; p *= 2) {  
    printf("%d\n", p);  
}
```

## While loops:

```
while (test) {  
body  
}
```

The **while** statement works by checking the test condition, if it is true then the body will be executed. After the body is executed it will check the test condition again and repeat until the test condition is no longer true.

```
i=10  
while (i >= 0) {  
    printf("%d\n", i);  
    i--;  
}
```

Again, the test expression requires a set of parentheses around it, there is no colon, and we use braces to surround the loop's body.



C for Python programmers by Carl Burch is licensed under a Creative Commons Attribution-Share Alike 3.0 United States License.

Based on a work at [www.toves.org/books/cpy/](http://www.toves.org/books/cpy/) adapted for Algebra 1 with Robots by James Town.