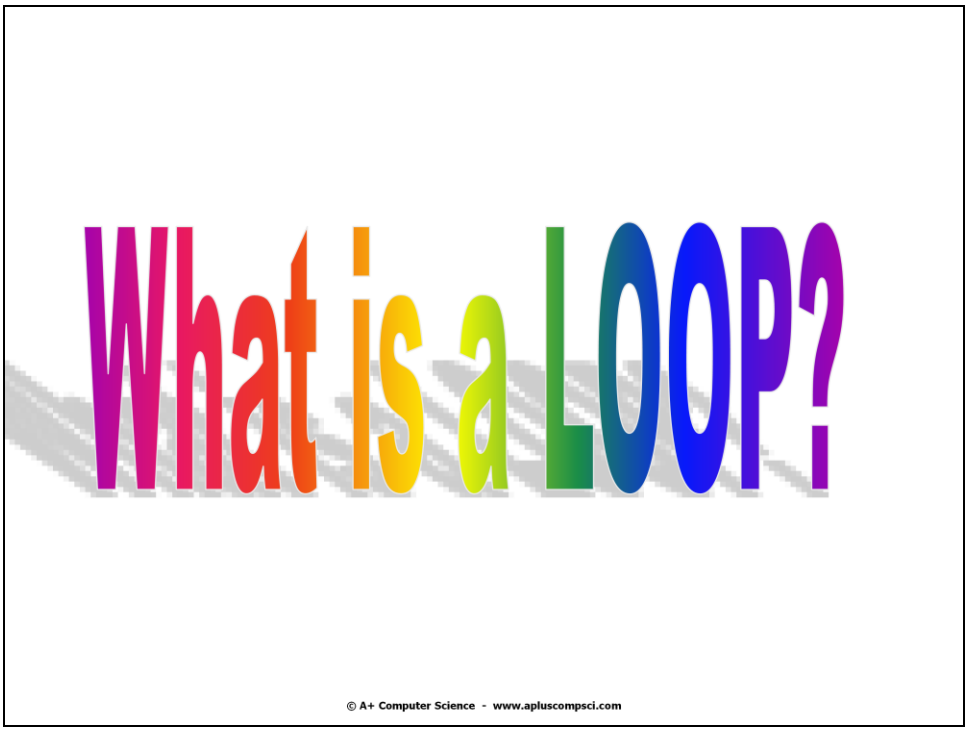
for loops


What is a LOOP?
© A+ Computer Science - www.apluscompsci.com

the loop

Until I can hear the song
Make it louder

Loops repeat as long
as something is true

If statements are just simple decision-making statements.

A condition is checked and something may or may not happen based on the evaluation of that condition.

the loop

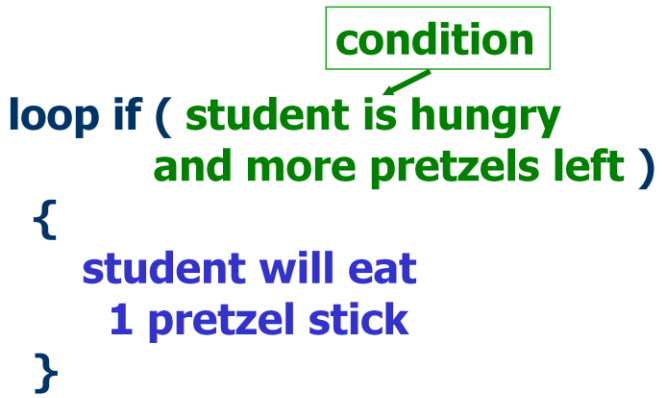
As long as I am hungry
I eat something

While I have pretzels
I eat one

If statements are just simple decision-making statements.

A condition is checked and something may or may not happen based on the evaluation of that condition.

Loop Demonstration



Loop Definition

A loop is a tool used to repeat a block of code. As long as the loop condition is true, the block of code associated with the condition is executed.

Loop Demonstration

```
total = 1
loop if (total < 2)
{
   The class will :
      1.  stand up
      2.  sit down

    total = total + 1
}
```

condition

block of code

total
1
2

Loop Condition

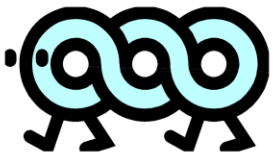
Every loop has a condition that must be true before the loop can run.

total<2 was the loop condition in the loop demonstration.


The for loop
© A+ Computer Science - www.apluscompsci.com

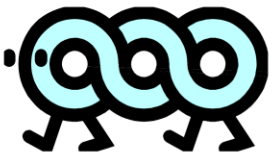
For Loop Definition

A for loop is a block of code associated with a condition. The block of code will run a set number of times depending on the loop condition and increment/decrement value.



the for loop

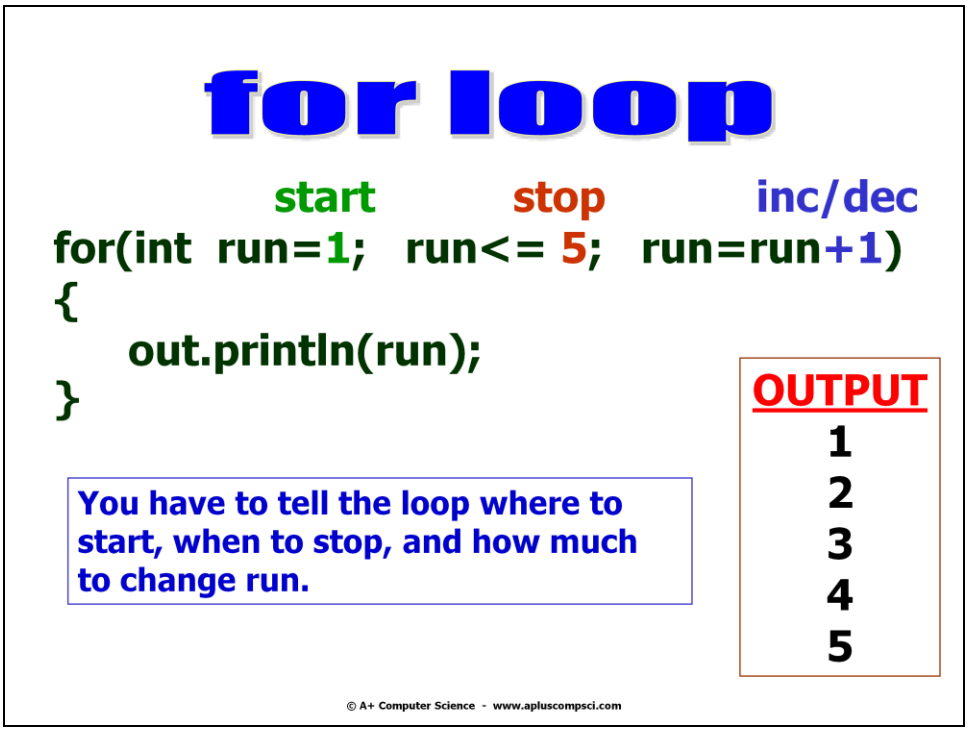
```
for(init value;  boolean condition placed here;  inc/dec)
{
  do something 1;
  do something 2;
}
```



`do something 1` and `do something 2` will occur if the condition is true.

If the condition is true, `do something 1` and `do something 2` will occur at least once.

`do something 1` and `do something 2` will continue to occur as long as the loop condition is true.



The for loop above starts run at 1. As long as run is less than or equal to 5 `( run<=5 )`, the loop will continue to run and print out the value of variable run. Run increases by one each iteration.

run begins with the value 1

Iteration 1 – print run(1) run = 1 + 1

Iteration 2 – print run(2) run = 2 + 1

Iteration 3 – print run(3) run = 3 + 1

Iteration 4 – print run(4) run = 4 + 1

Iteration 5 – print run(5) run = 5 + 1

The loop condition fails when run reaches the value 6 as 6 is not less than or equal to 5.

for loop

Some languages use a < start, stop, step > structure with for loops.

start – starting value of the loop
stop – ending value of the loop
step – amount to change the loop variable

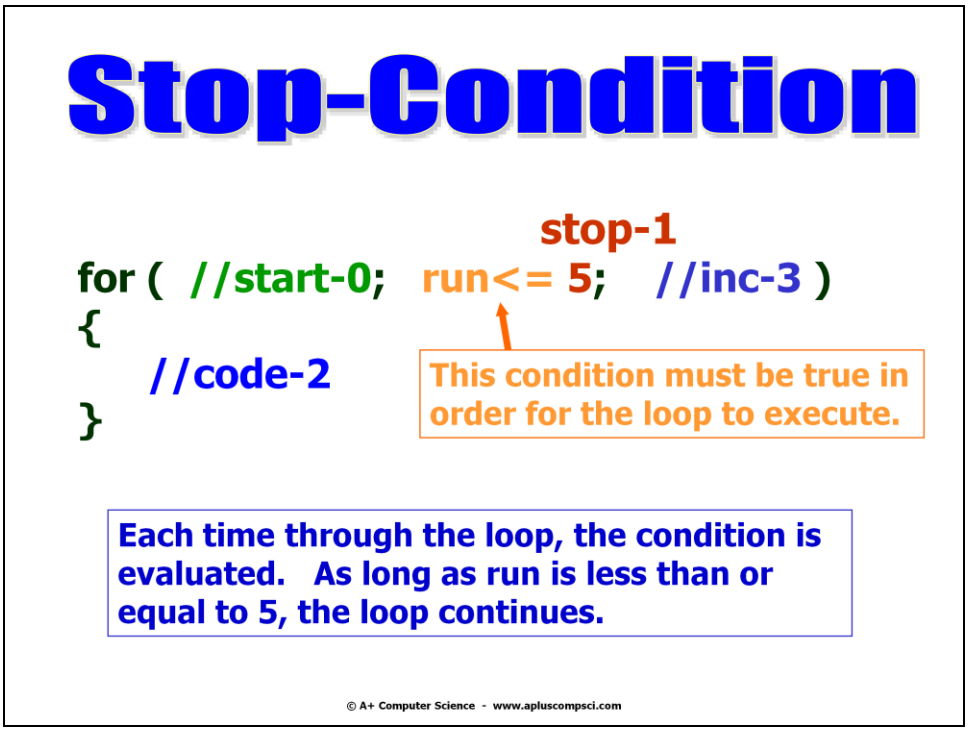
```
for x = start to stop step y   -   Visual Basic
  do something
next x
```

Start-Initialization

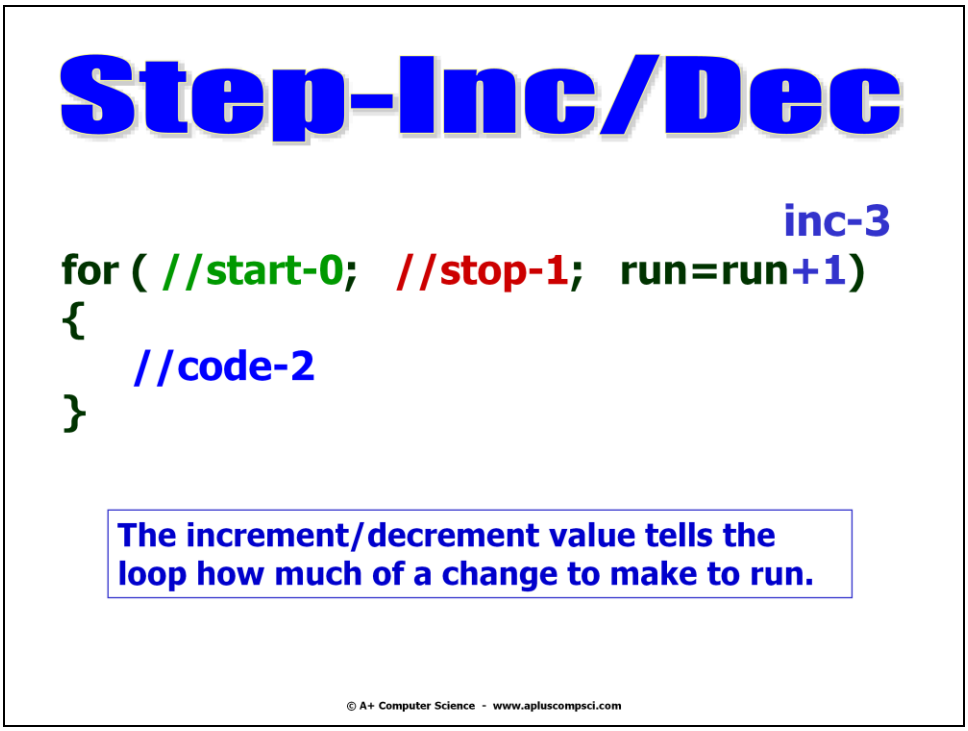
```
            start-0
for (int  run = 1;   //stop-1 ;   //inc-3 )
{
    //code-2
}
```

The start value tells the loop where to start.
run will start with a value of 1.

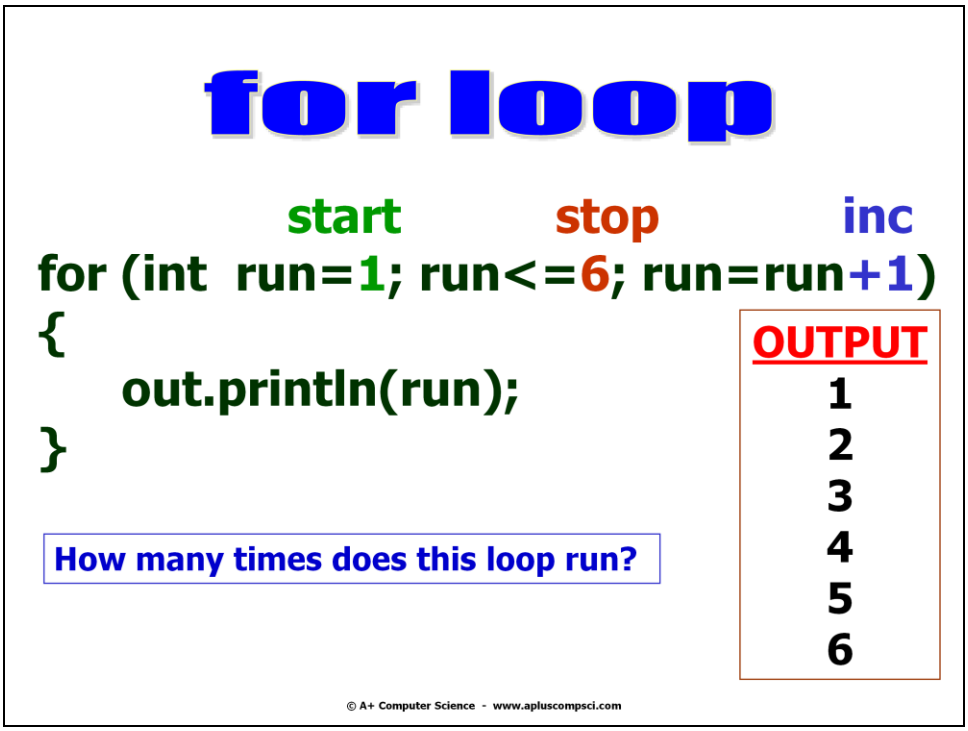
The start section of the loop happens only once. The start section sets up the loop.



The stop section sets the terms by which the loop can run. As long as run is less than or equal to 5, the loop will continue to run.



The step section of the loop either increases or decreases the value of run. In the example above, run is increased by 1 each iteration.



This loop starts run at 1 and increments run by one each iteration. The loop will continue to run as long as run is less than or equal to 6.

The loop will stop when the condition run<=6 fails.

run begins with the value 1

Iteration 1 – print run(1) run = 1 + 1

Iteration 2 – print run(2) run = 2 + 1

Iteration 3 – print run(3) run = 3 + 1

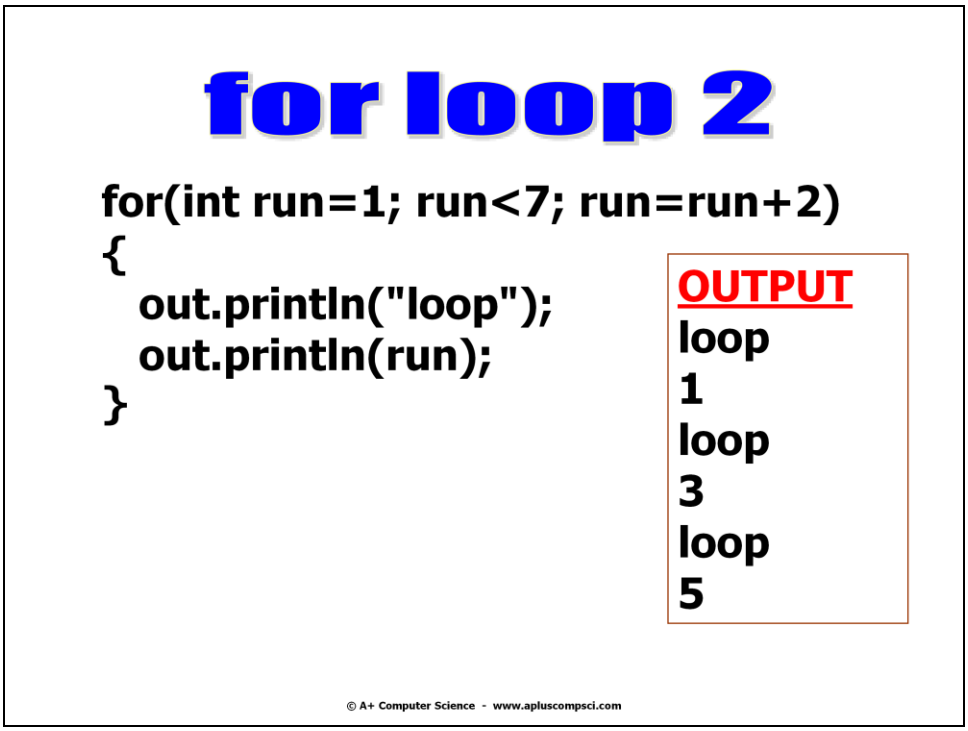
Iteration 4 – print run(4) run = 4 + 1

Iteration 5 – print run(5) run = 5 + 1

Iteration 6 – print run(6) run = 6 + 1

The loop condition fails when run reaches the value 7 as 7 is not less than or equal to 6.

Open forone.java



This loop starts run at 1 and increments run by two each iteration. The loop will continue to run as long as run is less than 7.

The loop will stop when the condition run<7 fails. The condition will fail when run equals 7.

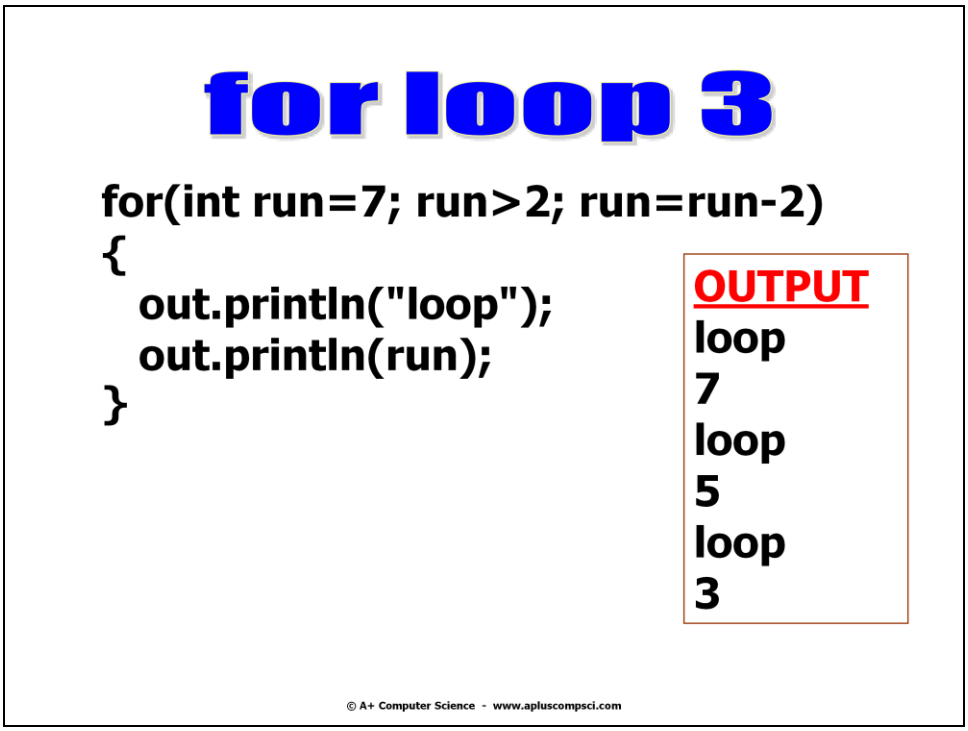
run begins with the value 1

Iteration 1 – print run(1) run = 1 + 2

Iteration 2 – print run(3) run = 3 + 2

Iteration 3 – print run(5) run = 5 + 2

The loop condition fails when run reaches the value 7 as 7 is not less than 7.



This loop starts run at 7 and decrements run by two each iteration. The loop will continue to run as long as run is greater than 2.

The loop will stop when the condition run>2 fails. The condition will fail when run equals 1.

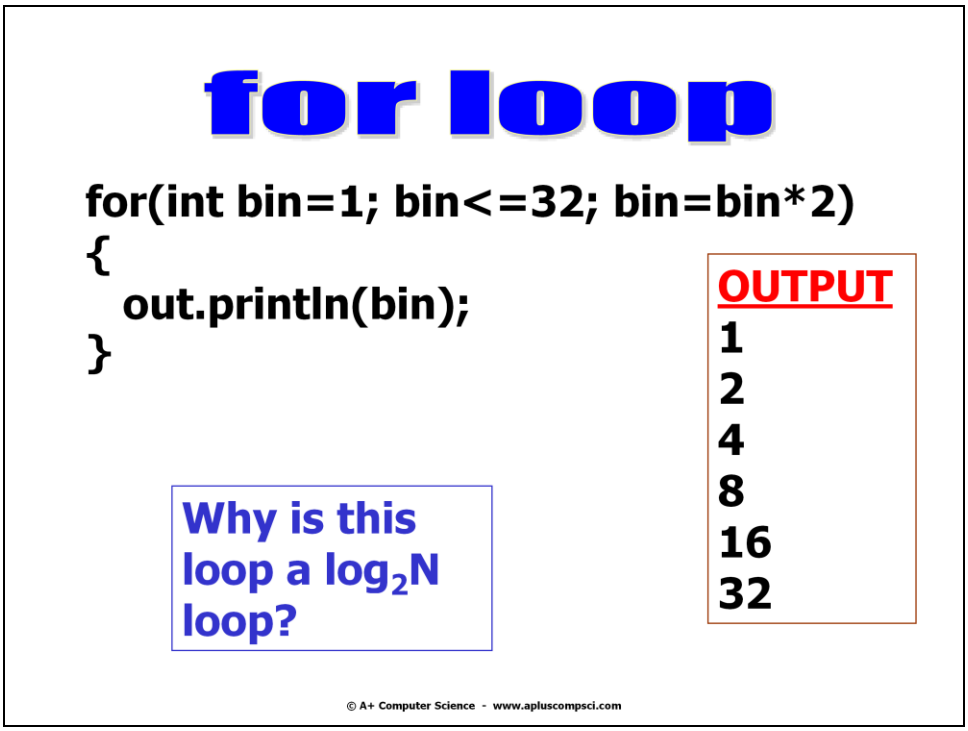
run begins with the value 7

Iteration 1 – print run(7) run = 7 - 2

Iteration 2 – print run(5) run = 5 - 2

Iteration 3 – print run(3) run = 3 - 2

The loop condition fails when run reaches the value 1 as 1 is not greater than 2.

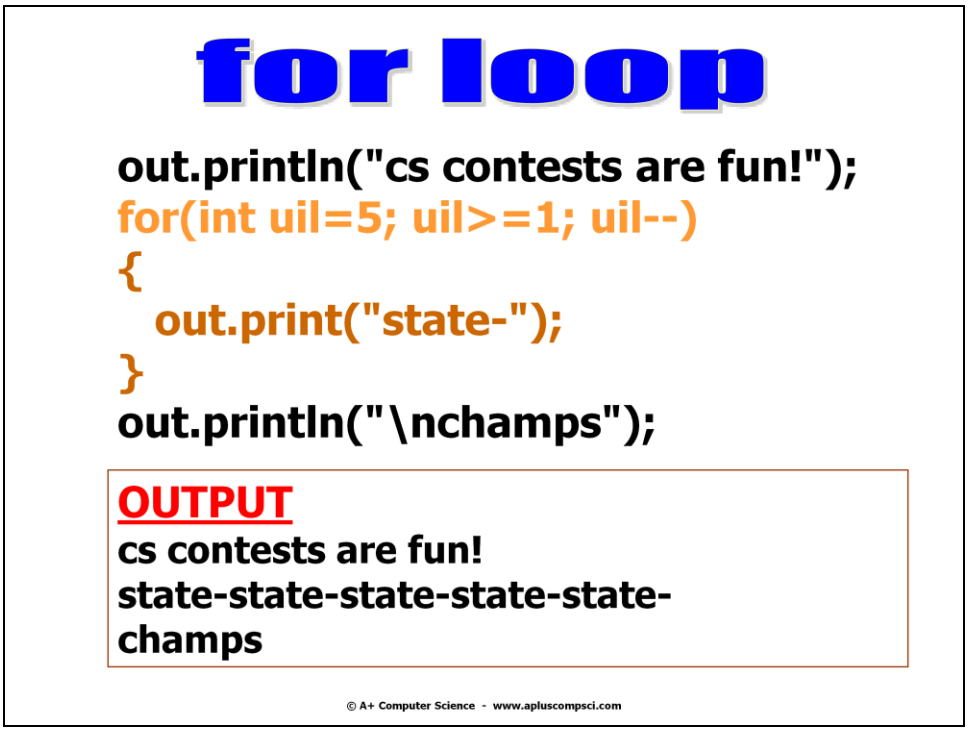


This loop starts bin at 1 and multiplies bin by 2 each iteration. The loop will continue to run as long as bin is less than or equal to 32.

The loop will stop when the condition bin<=32 fails. The condition will fail when run equals 64.

Open fortwo.java

Open forthree.java



This loop starts uil at 5 and decrements uil by 1. The loop will continue to run as long as uil is greater than or equal to 1.

The loop will stop when the condition uil>=1 fails. The condition will fail when uil equals 0.

Open foruil.java

Open graphicsrunner.java randomcoloredboxes.java

sum / total

Many times you will use a loop to total up a run of values.

total = total + run;

total is totaling up all values of run.

sum / total

```
int total = 0;
for(int run=1; run<6; run++)
{
   total=total+run;
}
out.println(total);
```


total starts at zero.

For each iteration of the for loop, the current value of run is added to total. Runs values would be 1,2,3,4,5,6

The loop fails when run reaches 6.

Iteration 1 – total = 1 + 1 total is 1

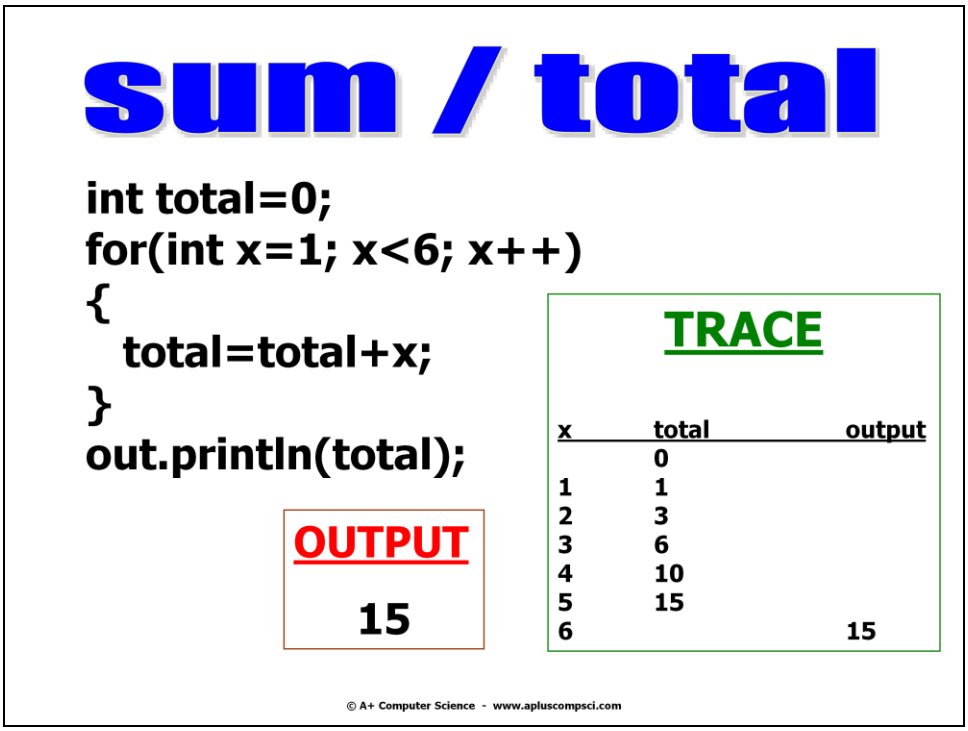
Iteration 2 – total = 2 + 2 total is 3

Iteration 3 – total = 3 + 3 total is 6

Iteration 4 – total = 6 + 4 total is 10

Iteration 5 – total = 10 + 5 total is 15

The loop condition fails when run reaches the value 6 as 6 is not less than 6.



total starts at zero.

For each iteration of the for loop, the current value of run is added to total. Runs values would be 1,2,3,4,5,6

The loop fails when run reaches 6.

Iteration 1 – total = 1 + 1 total is 1

Iteration 2 – total = 2 + 2 total is 3

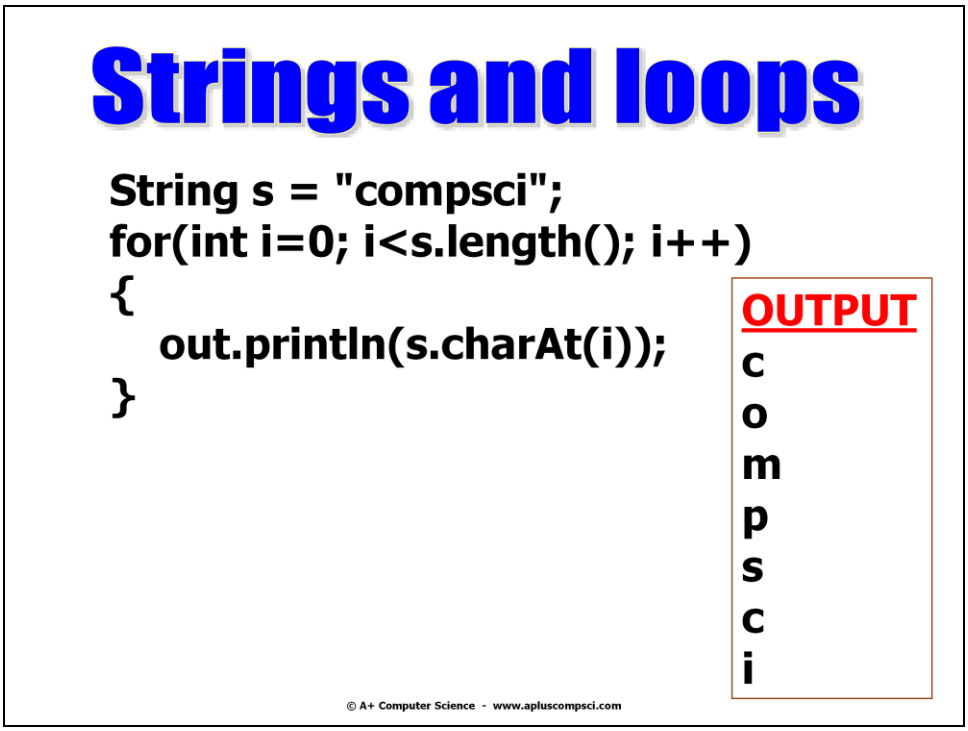
Iteration 3 – total = 3 + 3 total is 6

Iteration 4 – total = 6 + 4 total is 10

Iteration 5 – total = 10 + 5 total is 15

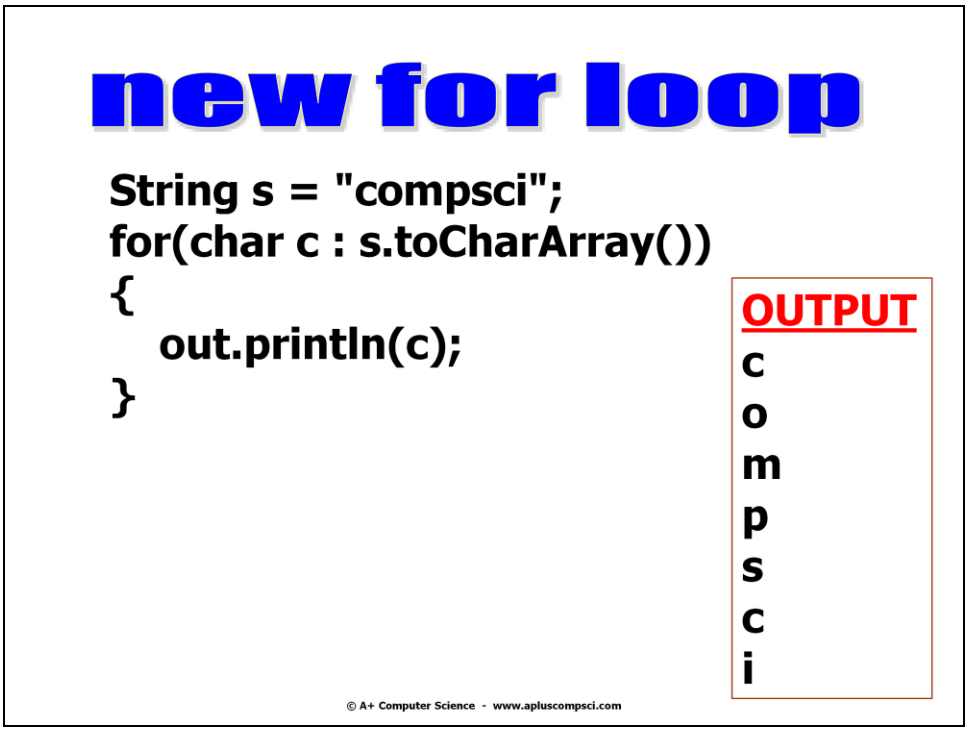
The loop condition fails when run reaches the value 6 as 6 is not less than 6.

Open fortotal.java



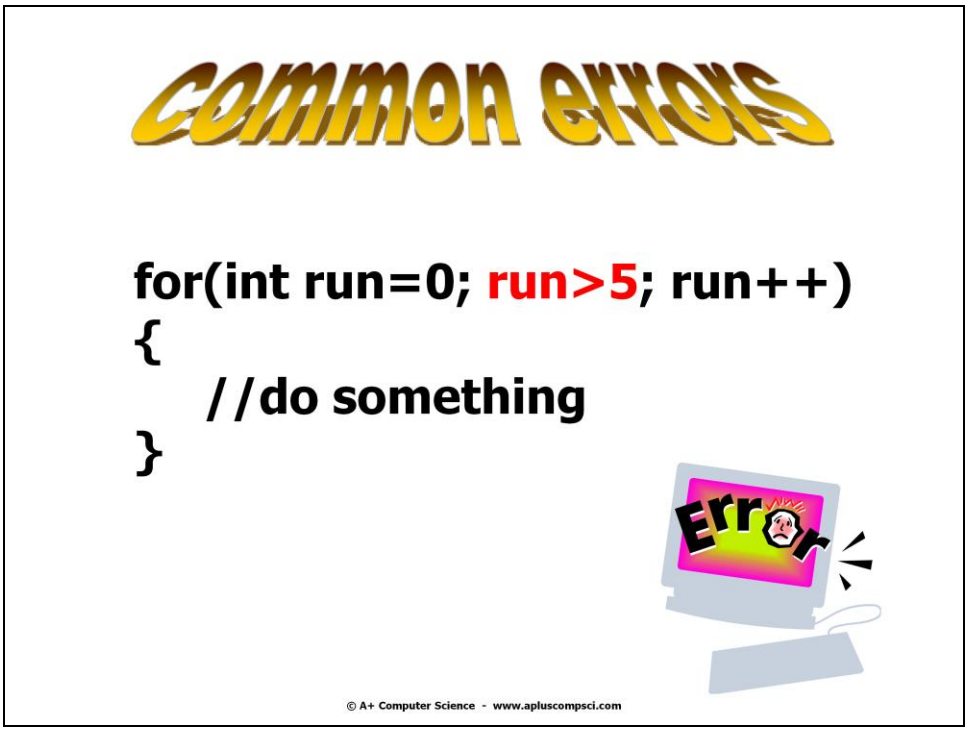
The for loop starts `i` at 0. As long as `i` is less than `s.length()`, the loop continues to run. For each iteration through the loop, `i` is incremented by 1.

The loop will print out each character in the String starting at 0 and going up through the String to `s.length()`.

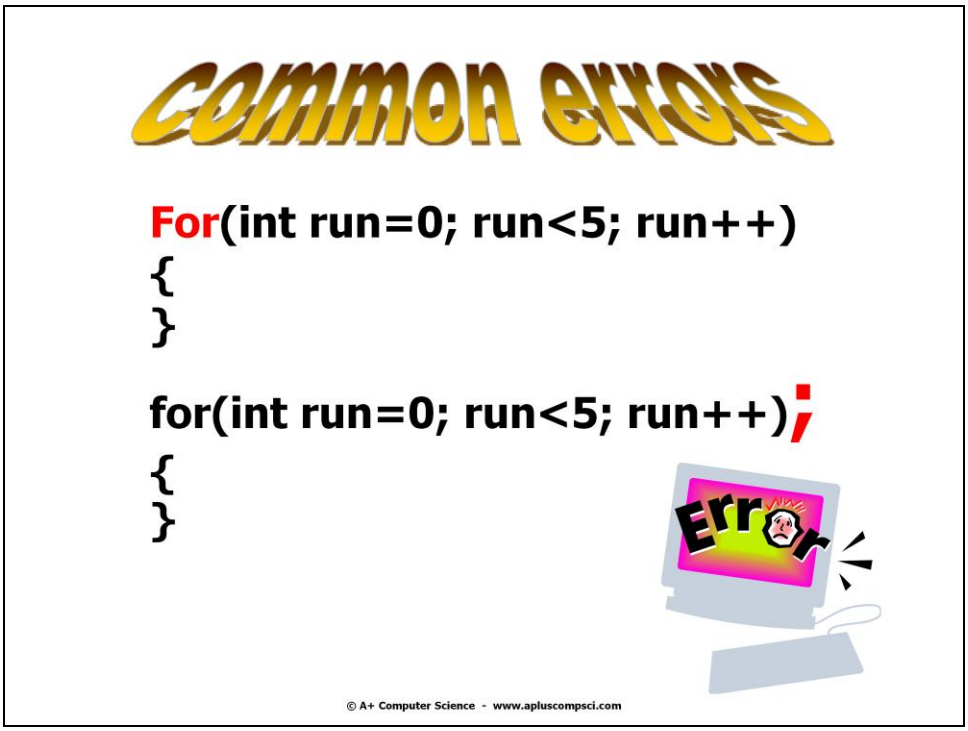


The new for loop iterates over the String starting at character c. Each character from the String is placed into c. Each letter is printed as it is encountered. The loop continues to run as long as there are more letters in the String.

Open forstrings.java newfor.java



The loop condition should always agree with the loops starting value.



For is always all lowercase. Never capitalize the f on for.

NEVER put a semi-colon before an OPEN BRACE.

{ and ; rule

Never put a ;
before an open { brace

;{ **illegal**

}; **legal**


Start work
on the labs
© A+ Computer Science - www.apluscompsci.com