

while and do while loops



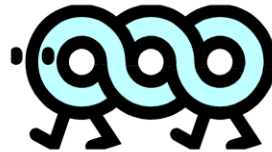
© A+ Computer Science - www.apluscompsci.com

The while loop

© A+ Computer Science - www.apluscompsci.com

While Loop Definition

A while loop is a block of code associated with a condition. As long as the condition is true, the loop will continue to run the block of code.



© A+ Computer Science - www.apluscompsci.com

the while loop

```
while( boolean condition placed here )  
{  
    do something 1;  
    do something 2;  
}
```



© A+ Computer Science - www.apluscompsci.com

As long as the condition is true, do something 1 and do something 2 will occur.

If the condition is false, do something 1 and do something 2 do not occur.

while loop

checks condition first

```
int run = 0;           //0 – start
while(run<5)           //1 - stop
{
    run = run + 1;      //2 - increment
    out.println(run);   //3 - code
}
```

OUTPUT

1
2
3
4
5

© A+ Computer Science - www.apluscompsci.com

As long as run is less than 5 (`run<5`), the loop will iterate. For each iteration, run is increased by 1 and run is displayed.

run begins with the value 0

Iteration 1 – `run = 0 + 1` `print(1)`

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

Iteration 3 – `run = 2 + 1` `print(3)`

Iteration 4 – `run = 3 + 1` `print(4)`

Iteration 5 – `run = 4 + 1` `print(5)`

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

while loop

```
int run = 7;           //0 - start
while(run<10)          //1 - stop
{
    out.println(run);  //2- code
    run++;             //3 - increment
}
```

What is the final value of run?

OUTPUT

7
8
9

© A+ Computer Science - www.apluscompsci.com

As long as run is less than 10 ($run < 10$), the loop iterates. For each iteration, run is displayed and then increased by 1.

The loop condition fails when run reaches the value 10 as 10 is not less than 10.

run begins with the value 7

Iteration 1 – print(7) $run = 7 + 1$

Iteration 2 – print(8) $run = 8 + 1$

Iteration 3 – print(9) $run = 9 + 1$

The loop condition fails when run reaches the value 10 as 10 is not less than 10.

Open whileone.java

© A+ Computer Science - www.apluscompsci.com

while loop 2

```
int run=25;
while(run>=10)
{
    out.println(run);
    out.println("loop");
    run=run-5;
}
```

OUTPUT

```
25
loop
20
loop
15
loop
10
loop
```

© A+ Computer Science - www.apluscompsci.com

As long as run is less than or equal to 10 ($run \leq 10$), the loop will iterate. For each iteration, run is displayed, loop is displayed, and run is decreased by 5.

run begins with the value 25

Iteration 1 – print(25) print(loop) run = 25-5

Iteration 2 – print(20) print(loop) run = 20-5

Iteration 3 – print(15) print(loop) run = 15-5

Iteration 4 – print(10) print(loop) run = 10-5

The loop condition fails when run reaches the value 5 as 5 is not greater than or equal to 10.

while loop 3

```
int run=10;
while(run<=25)
{
    out.println(run);
    out.println("loop");
    run=run+5;
}
```

What is the final value of run?

OUTPUT

```
10
loop
15
loop
20
loop
25
loop
```

© A+ Computer Science - www.apluscompsci.com

As long as run is less than or equal to 25 (`run<=25`), the loop will iterate. For each iteration, run is displayed, loop is displayed, and run is increased by 5.

run begins with the value 10

Iteration 1 – print(10) print(loop) run = 10+5

Iteration 2 – print(15) print(loop) run = 15+5

Iteration 3 – print(20) print(loop) run = 20+5

Iteration 4 – print(25) print(loop) run = 25+5

The loop condition fails when run reaches the value 30 as 30 is not less than or equal to 25.

tracing a while loop

```
int total=0,x=1;
while(x<6)
{
    total=total+x;
    x++;
}
out.println(total);
```

TRACE

x	total	output
	0	
1	1	
2	3	
3	6	
4	10	
5	15	
6		15

© A+ Computer Science - www.apluscompsci.com

As long as x is less than 6 ($x < 6$), the loop will iterate. For each iteration, total is increased by the value of x and x is increased by 1.

total begins with the value 0 and x begins with the value of 1

Iteration 1 – total = $0 + 1$ $x = 1 + 1$

Iteration 2 – total = $1 + 2$ $x = 2 + 1$

Iteration 3 – total = $3 + 3$ $x = 3 + 1$

Iteration 4 – total = $6 + 4$ $x = 4 + 1$

Iteration 5 – total = $10 + 5$ $x = 5 + 1$

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

**Open
whiletwo.java**

**Open
whilethree.java**

© A+ Computer Science - www.apluscompsci.com

Dissecting *Dissecting* **Numbers**

© A+ Computer Science - www.apluscompsci.com

Accessing Digits

```
int num = 9154;  
out.println( num % 10 );  
out.println( num / 10 );  
num /= 10;  
out.println( num % 10 );  
out.println( num / 10 );
```

OUTPUT

```
4  
915  
5  
91
```

© A+ Computer Science - www.apluscompsci.com

% is used to access the remainder of division.

In the case above, % 10 is being used on a base 10 number to access the right most digit.

/ 10 is being used to chop off the right most digit.

Accessing Digits

How would you take apart the number 9154 digit by digit?

You would need a loop.

© A+ Computer Science - www.apluscompsci.com

In this example, the loop iterates as long as number is larger than 0.

Each time the loop iterates, the right most digit of the number is printed. % 10 is used to access the right most digit.

The number is also reduced by removing the right digit using / 10 and assigning that value back to number.

Accessing Digits

```
int number = 9154;  
while( number > 0 )  
{  
    out.println( number % 10 );  
    number = number / 10;  
}
```

OUTPUT

4
5
1
9

© A+ Computer Science - www.apluscompsci.com

In this example, the loop iterates as long as number is larger than 0.

Each time the loop iterates, the right most digit of the number is printed. % 10 is used to access the right most digit.

The number is also reduced by removing the right digit using / 10 and assigning that value back to number.

Open accessingdigits.java

© A+ Computer Science - www.apluscompsci.com

Summing Digits

set total to 0

**while num is greater than 0
 add right most digit to total
 remove right most digit**

print out the total

© A+ Computer Science - www.apluscompsci.com

Open **summingdigits.java** **Complete the code**

© A+ Computer Science - www.apluscompsci.com

Open
averagingdigits.java
Complete the code

© A+ Computer Science - www.apluscompsci.com

Common Common **Errors**

© A+ Computer Science - www.apluscompsci.com

COMMON ERRORS

```
int run=0;  
while(run<5)  
{  
    out.println(run);  
    < blank 1 >  
}
```



© A+ Computer Science - www.apluscompsci.com

This loop will run forever as there is no code to change the value of variable run. run begins with the value of 0 and run never changes. 0 is less than 5; thus the loop will never terminate as the condition will never fail.

Start work on the labs

© A+ Computer Science - www.apluscompsci.com

The do-while loop

© A+ Computer Science - www.apluscompsci.com

Do While Loop Definition

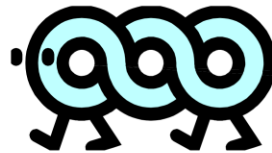
A do while loop is a group of statements that will run once before checking the loop condition. If the condition is true, the loop will run again. This run and check process will continue until the condition evaluates false.



© A+ Computer Science - www.apluscompsci.com

the do while loop

```
do{  
    do something 1;  
    do something 2;  
}while( boolean condition placed here );
```



© A+ Computer Science - www.apluscompsci.com

do something 1 and do something 2 will occur before the loop condition is evaluated. If the loop condition is true, do something 1 and do something 2 will occur again.

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

do while loop

```
int run=0;           //0 - start
do
{
    run = run + 1;    //1 - increment
    out.println(run); //2 - code
} while(run<4);      //3 - stop
```

OUTPUT

1
2
3
4

© A+ Computer Science - www.apluscompsci.com

Before the loop condition is evaluated, run is increased by 1 and run is displayed.

As long as run is less than 4 (`run<4`), the loop will continue to iterate.

run begins with the value 0

Iteration 1 – `run = 0 + 1` `print(1)`

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

Iteration 3 – `run = 2 + 1` `print(3)`

Iteration 4 – `run = 3 + 1` `print(4)`

The loop condition fails when run reaches the value 4 as 4 is not less than 4.

do while loop

```
int run=0;           //0 - start
do
{
    run++;           //1 - increment
    out.println(run); //2 - code
} while(run<4);      //3 - stop
```

OUTPUT

1
2
3
4

© A+ Computer Science - www.apluscompsci.com

Before the loop condition is evaluated, run is increased by 1 and run is displayed.

As long as run is less than 4 (`run<4`), the loop will continue to iterate.

run begins with the value 0

Iteration 1 – `run = 0 + 1` `print(1)`

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

Iteration 3 – `run = 2 + 1` `print(3)`

Iteration 4 – `run = 3 + 1` `print(4)`

The loop condition fails when run reaches the value 4 as 4 is not less than 4.

do while loop

```
int run= 4;           //0 - start
do
{
    out.println(run); //1 - code
    run--;            //2 - decrement
} while(run>0);       //3 - stop
```

OUTPUT

4
3
2
1

© A+ Computer Science - www.apluscompsci.com

Before the loop condition is evaluated, run is displayed and decreased by 1.

As long as run is greater than 0 (`run>0`), the loop will continue to iterate.

run begins with the value 4

Iteration 1 – print(4) run = 4 - 1

Iteration 2 – print(3) run = 3 - 1

Iteration 3 – print(2) run = 2 - 1

Iteration 4 – print(1) run = 1 - 1

The loop condition fails when run reaches the value 0 as 0 is not greater than 0.

**Open
dowhileone.java**

© A+ Computer Science - www.apluscompsci.com

do while loop 2

```
int run=25;  
do{  
    out.println(run);  
    out.println("loop");  
    run=run-5;  
} while(run>=10);
```

What is the final value of run?

OUTPUT

```
25  
loop  
20  
loop  
15  
loop  
10  
loop
```

© A+ Computer Science - www.apluscompsci.com

Before the loop condition is evaluated, run is displayed, loop is displayed, and run is decreased by 5.

As long as run is greater than or equal to 10 (`run>=10`), the loop will continue to iterate.

run begins with the value 25

Iteration 1 – print(25) print(loop) run = 25 - 5

Iteration 2 – print(20) print(loop) run = 20 - 5

Iteration 3 – print(15) print(loop) run = 15 - 5

Iteration 4 – print(10) print(loop) run = 10 - 5

The loop condition fails when run reaches the value 5 as 5 is not greater than or equal to 10.

do while loop 3

```
int run=10;  
do{  
    out.println(run);  
    out.println("loop");  
    run=run+5;  
} while(run<=25);
```

What is the final value of run?

OUTPUT

```
10  
loop  
15  
loop  
20  
loop  
25  
loop
```

© A+ Computer Science - www.apluscompsci.com

Before the loop condition is evaluated, run is displayed, loop is displayed, and run is increased by 5.

As long as run is less than or equal to 25 (`run<=25`), the loop will continue to iterate.

run begins with the value 10

Iteration 1 – print(10) print(loop) run = 10 + 5

Iteration 2 – print(15) print(loop) run = 15 + 5

Iteration 3 – print(20) print(loop) run = 20 + 5

Iteration 4 – print(25) print(loop) run = 25 + 5

The loop condition fails when run reaches the value 30 as 30 is not less than or equal to 25.

**Open
dowhiletwo.java**

**Open
dowhilethree.java**

© A+ Computer Science - www.apluscompsci.com

COMMON ERRORS

```
int run=0;  
do{  
    out.println(run);  
    < blank 1 >  
} while(run<5)
```



© A+ Computer Science - www.apluscompsci.com

This loop will run forever as there is no code to change the value of variable run. run begins with the value of 0 and run never changes. 0 is less than 5; thus the loop will never terminate as the condition will never fail.

Continue work on the labs

© A+ Computer Science - www.apluscompsci.com