

## loops in java

print no's from 1 to 10.

```
print (1);
print (2);
print (3);
print (4);
...
```

```
print (10);
```

print no's from 1 to 1000 ;

```
print (1);
...
print (1000);
```

loops are used to repeat a block of code.

→ avoids code duplication

### Types of loops

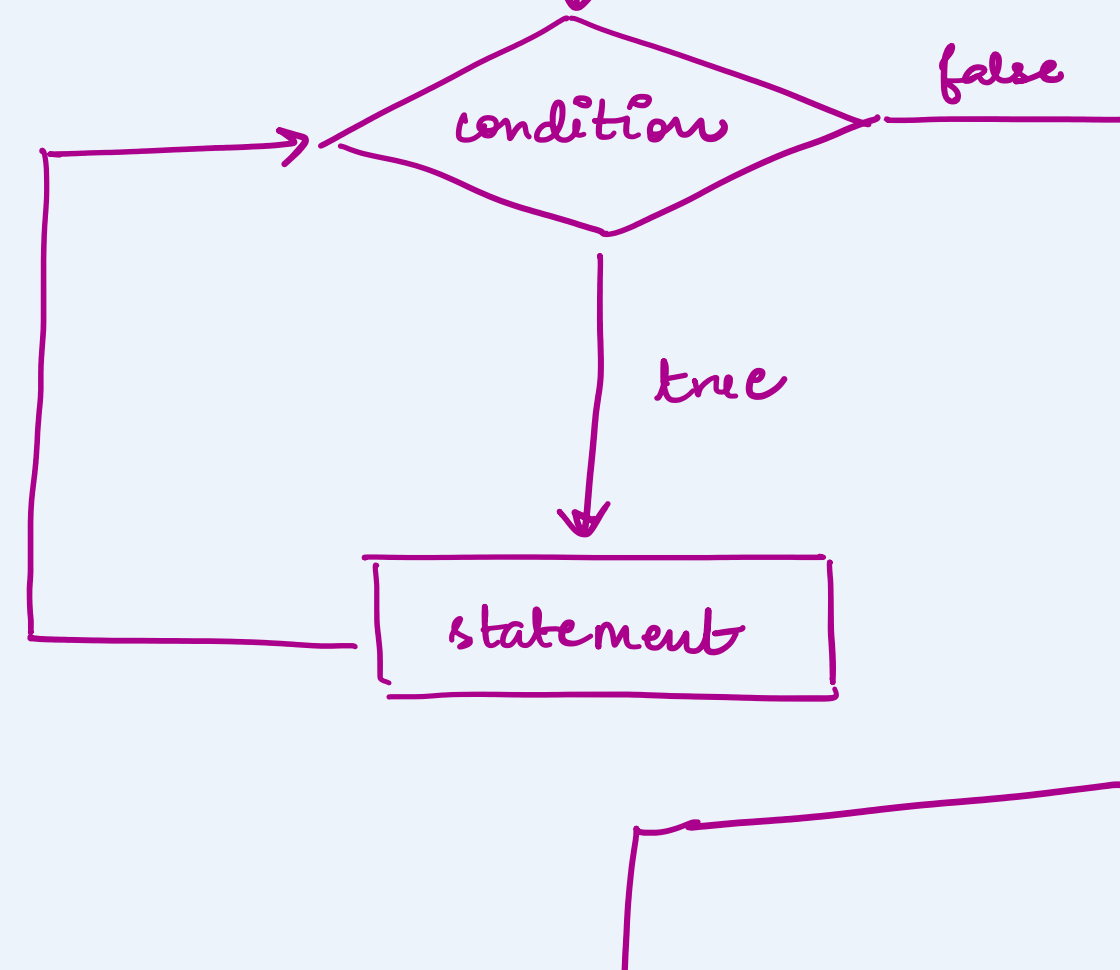
- ① for loop
- ② while loop
- ③ do while loop

### while loop

Syntax :

```
while (condition) {
    // code to be executed
}
```

Flow chart :



$i = 1, 2, 3, 4, \dots, 10$

```
int i = 1; // initialisation
while (i <= 10) { // condition
    print(i);
    i++; // update expression
}
```

1  
2  
3  
4  
...  
10

### Infinite while loop

```
while (true) {
    print("Infinite");
}
```

### for loop

Syntax :

```
for (initialExpression; testExpression; updateExpression) {
    // code to be executed.
}
```

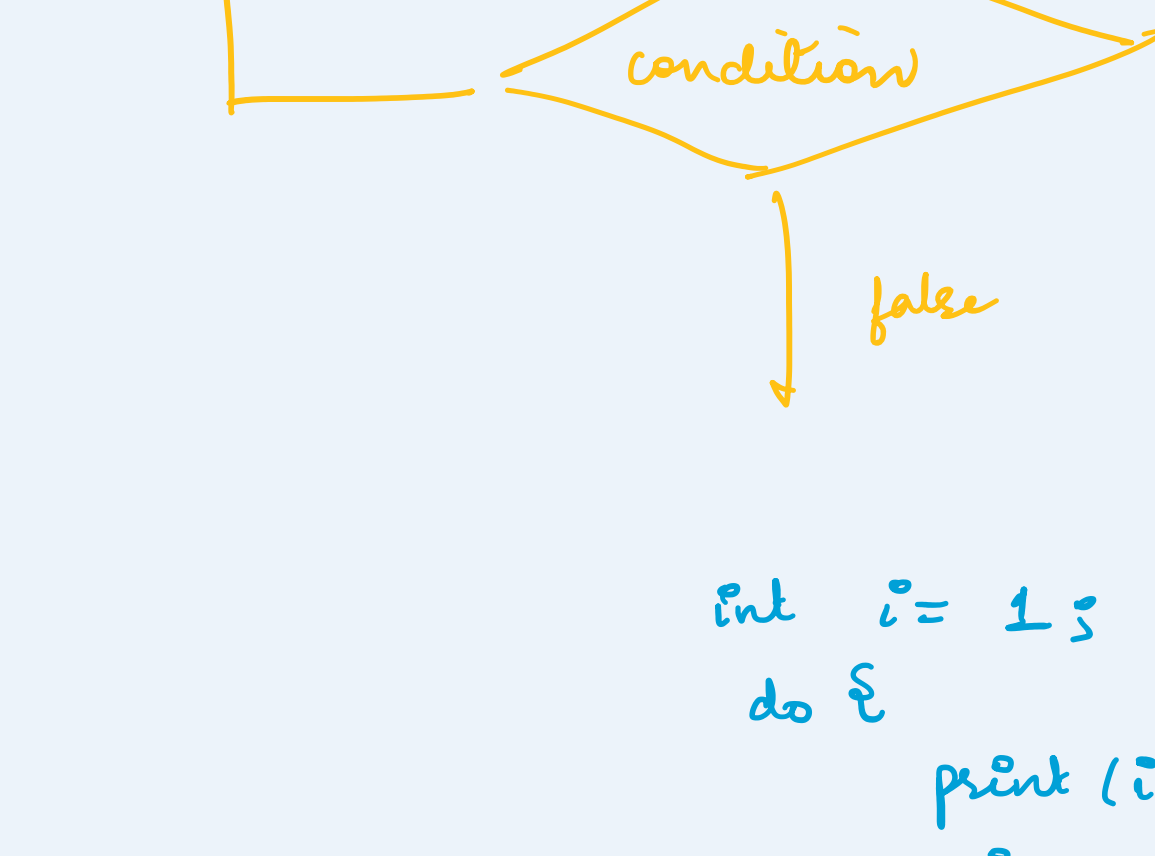
### do while loops

→ If the no. of iterations is not fixed and you must execute the loop atleast once, it is recommended to use do-while loop.

Syntax :

```
do {
    // code to be executed.
} while (condition);
```

Flow chart :



```
int i = 1;
do {
    print(i);
    i++;
} while (i <= 10);
```

Q →

calculate power  
 $a^b$

ans =  $1 \times a \times a \times \dots \times a$  (b times)

①  $a = 3, b = 2$   
 $3^2 \rightarrow 9 \rightarrow 3 \times 3 \rightarrow 9$

②  $a = 4, b = 3$   
 $4^3 \rightarrow 64 \rightarrow 4 \times 4 \times 4$

### Keywords

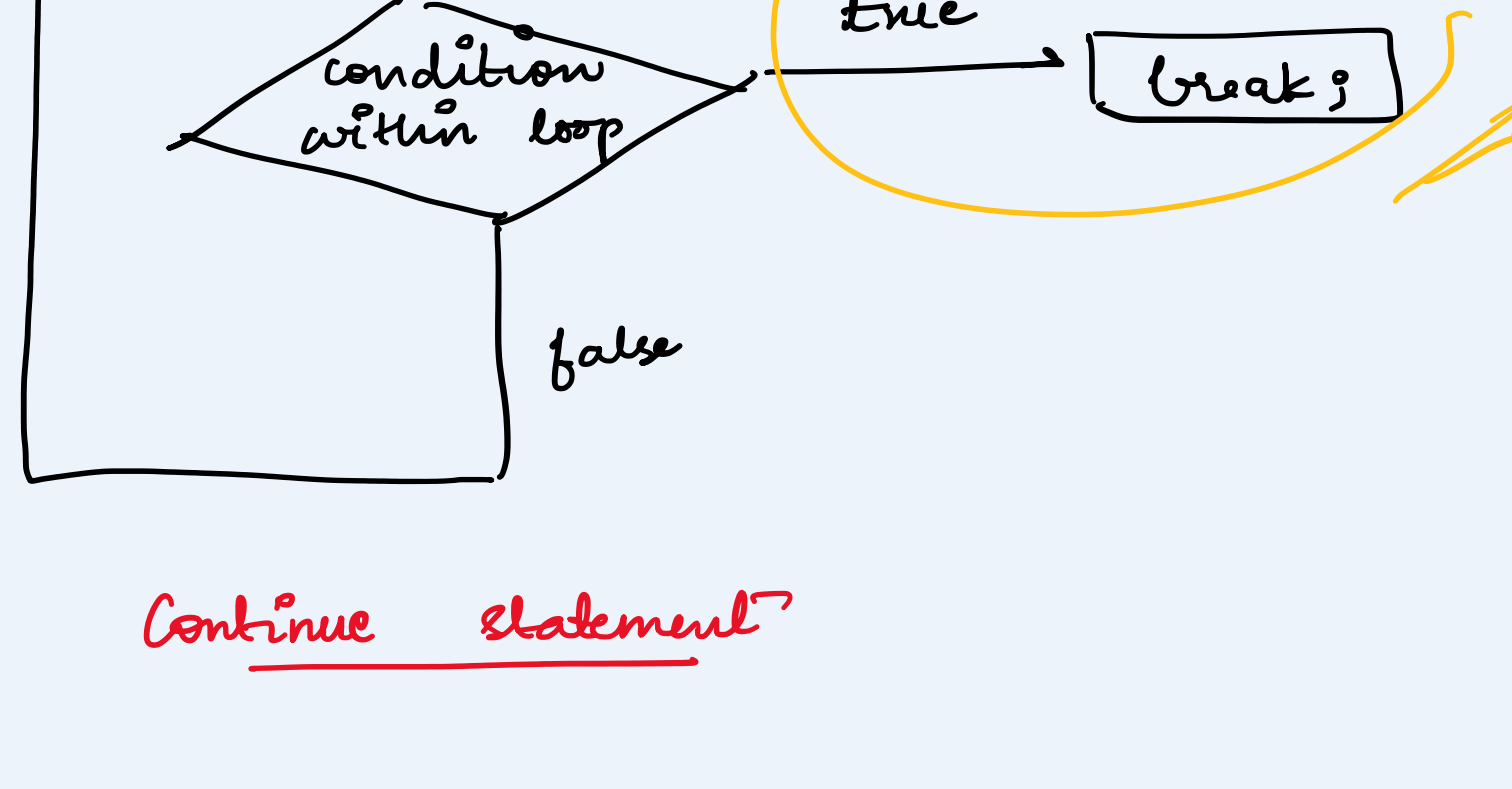
Java keywords are also known as reserved words.

→ These are predefined by java so it cannot be used as a variable or object name.

break, continue, byte, class, case, if, else, while, do, long, for

### Break statement

When a break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.



### Continue statement

used in loop control structure when you need to jump to the next iteration of the loop immediately.

Syntax: continue;

Q Write a program to check if a no is prime or not.

→ no → 7 → true  
 → no → 9 → false

Prime no →

$n$  factors: 1, no. itself  
 divisible by 2 →  $n-1$   
 $n-2, \dots, n-1$

Q

Print fibonacci numbers.  
 0, 1, 1, 2, 3, 5, 8, 13, ...  
 $f(n) = f(n-1) + f(n-2)$   
 first = 0, second = 1

Q

given a no. check if it is a armstrong no. or not.

$3^3 + 3^3 + 3^3 = 27 + 27 + 27 = 81$

$1^4 + 6^4 + 3^4 + 4^4 = 1 + 1296 + 81 + 256 = 1634$

### Two steps

① given a no, find the no. of digits

Take each digit one by one, power it with digits and add all values

