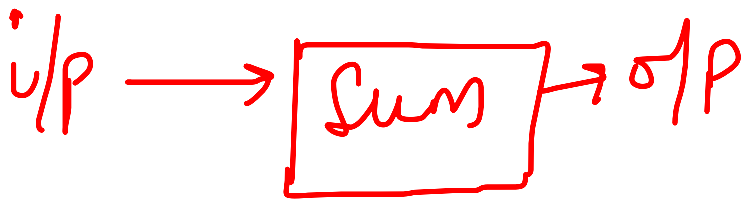
Functions → seperate code which can be called in program multiple times.

→ Reusability → Readable



Syntax

(output/return type → o/p; parameters → i/p)

```
public static output/return type func_name(parameters){
    // ________
}
```

calling a function

function_name(arguments)

(arguments ↳ i/p)

Note: public static will always be there before every function.

Some points

→ No. of parameters & arguments matters.
during define
during declare
during function calling.

→ Sequence of arguments matters.

→ i/p in parameters can be of different data type.

→ calling a function & storing a function both are different. sum(10,15) → call!
int ans = sum(10,15); → store

→ parameters can be empty, function will be void.

→ can't collect the function, if its void

→ naming - camel case.

2 4 6 8 10
10 12 14 16 18 20

HW_print all even number

```java
public static void main(String[] args) {

    Scanner s = new Scanner(System.in);
    int t = s.nextInt();

    for(int i = 1; i<=t;i++){
        int x = s.nextInt();
        int y = s.nextInt();
        even(x,y);
    }
}

public static void even(int x ,int y){
    for(int i = x; i <= y; i++){
        if(i % 2 == 0){
            System.out.print(i + " ");
        }
    }
    System.out.println();
}
```

t = 2

x = 1 y = 10

x = 10 y = 20

i = 1 ≤ 10 %.2 == 0

2 ≤ 10 %.2 == 0

3

⋮

10 ≤ 10 %.2 == 0

i = 10 ≤ 20 %.2 == 0

= 11 ≤ 20 %.2 == 0

⋮

20 ≤ 20 %.2 == 0