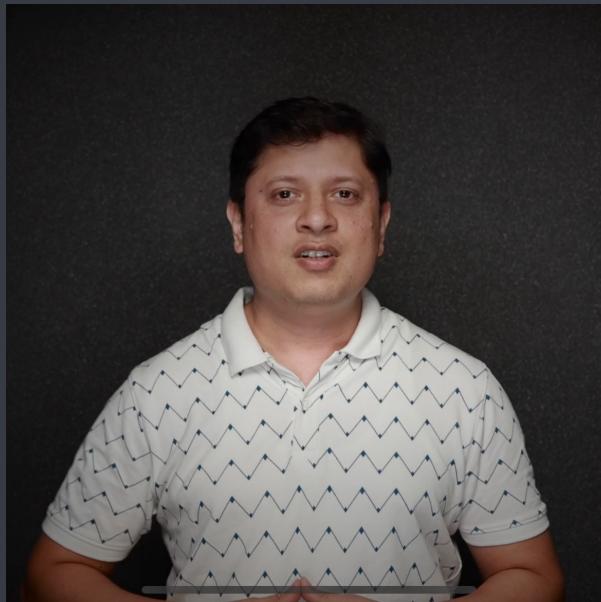


C Language

Function



Saurabh Shukla (MySirG)

Agenda

- ① Ways to define a function
- ② Header files and Library files

Ways to define a function

1. Takes Nothing, Returns Nothing
2. Take something, Returns Nothing
3. Takes Nothing, Return Something
4. Take something, Return Something

```
f1() { int a, b; f2() { printf("%d", a); } }
```

a b

```
#include <stdio.h>
void add();
int main()
{
    add();
    return 0;
}
void add()
{
    int a,b,c;
    printf("Enter two numbers");
    scanf("%d %d", &a, &b);
    c=a+b;
    printf("sum is %d", c);
}
```

T N R N

function declaration | function prototype

Return type
(Datatype of returned value)

function call

No return Keyword → RN

function definition

```
#include<stdio.h> TSRN
void add(int ,int );
int main()
{
    int x=3, y=4;
    add(x, y); Actual  
Arguments(parameters)
    return 0;
}
```

```
void add(int a, int b)
{
    int c; ↑  
formal Arguments
    c=a+b;
    printf("sum is %.d", c);
}
```

```
#include <stdio.h> T N RS  
int add();  
int main()  
{ int s;  
  
    s = add();  
    printf("sum is %d", s);  
    return 0;  
}  
  
int add()  
{  
    int a, b, c;  
    printf("Enter two numbers");  
    scanf("%d %d", &a, &b);  
    c = a + b;  
    return c;  
}
```

return
① value return
 ~~cont~~
② control return
 ~~cont~~

You can return
only one value

```
#include<stdio.h> TSRS
int add(int, int);
int main()
{
    int x=3, y=4, s;
    s = add(x, y); 
    printf("sum is %d", s);
    return 0;
}
```

```
}
```

```
int add (int a,int b)
{
    int c;
    c=a+b;
    return c;
}
```

Definition of predefined function

→ Library files

Declaration of predefined function

→ Header files

Ask Yourself

1. Function call vs Function Definition vs Function Declaration
2. Function saves memory
3. main() is a userdefined function.
4. Header file vs Library file
5. When to write void?
6. When to write return?
7. Return type of main()
8. call by value
9. Function Prototype = function declaration
10. Can we call main() function()? Yes

topmate.io/mysirg

first.c

```
#include "c:/---/second.c"  
main()  
{  
    add();  
}
```

Second.c

```
void add( )
```

```
{  
    ...  
}
```

```
printf("%d", add(3, 4));
```