

DAY# 23.

Saturday, June 20, 2026 11:04 AM

Function in C programming.

⇒ A block of code that performs a specific task.

A named block of code

⇒ It can be called from different parts of program

Syntax →

```
returnType functionName (arguments  
parameters)  
{  
    // Block of code  
}
```

returnType can be any datatype including void
functionName any name allowed in C.
() parenthesis
↑ ↑
opening closing

arguments → 0 argument
 ⇒ or 1 more
passing N arg ⇒ returnType functionName ()
 {
 =
 }

pass 1 arg ⇒ returnType functionName (arg)
 {
 =
 }

More than 1 arg ⇒ returnType functionName (arg1, arg2, ...)

{ } ⇒ opening & closing braces / curly brackets.

How to understand / identify its function →

if should have returnType functionName ()
 {
 = return value
 }

CATEGORIES of function

Predefined function
⇒ function provided by C library
print () ⇒ predefined function
⇒ We can't change it

Userdefined function
⇒ We can create our own function
⇒ Create one function to print Hello five times
⇒ We can create n number of functions
⇒ Userdefined function's name should not be same as "predefined function".

1. Based on Arguments & return value.

⇒ No arguments & No return value

```
void myfunction ( )  
{  
=       
}
```

⇒ Arguments & No return value

```
void myfunction ( arg1, arg2, ... )  
{  
=       
}
```

⇒ Arguments & some return value

```
int myfunction ( arg1, arg2 )  
{  
=       
    return 10;  
}
```

⇒ No arguments & return value

```
int myfunction ( )  
{  
=       
}
```

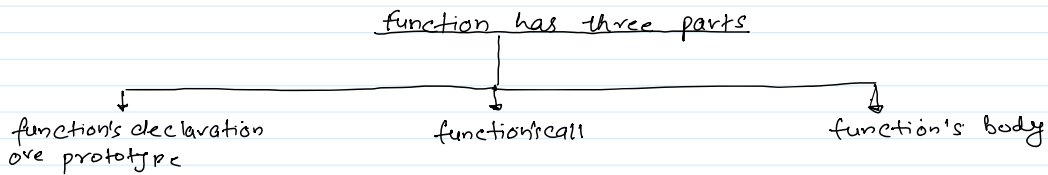
2. Inline function

↓
It's a keyword
inline void myfunction ()
{
=
}

3. Static function

↑
It's a keyword
static void functionname ()
{
=
}

return 10;



Syntax: →

Declaration - void functionname (Arguments); ✓

function call - functionname (arguments); ✓

function definition - void functionname (Arguments) {
 printf("Hello");
 printf("Hello");
 } ✓

Example:- // print Hello two times using a function

#include <stdio.h>

void printHelloTwotimes(_); * ⇒ function declaration.

⇒ int main()

{ printHelloTwotimes(); ⇒ function call

return 0;

}

⇒ void printHelloTwotimes() ⇒ Definition

{
 printf("Hello");
 printf("Hello");

}

==>

Why use function →

1. We can call same function n number of times from different places of program

2. print "Hello" twenty times.

printHelloTwotimes();
printHelloTwotimes();

⋮

// call function 3 times

==>

#include <stdio.h>

void printHelloTwotimes();

int main()

{ ~~printHelloTwotimes();~~

printHelloTwotimes();

printHelloTwotimes();

return 0;

}

```
→ return 0;  
}  
→ void printHelloTwoTimes() {  
→ printf("\n Hello");  
→ printf("\n Hello");  
→ printf("\n");  
→ }  
→ }
```

Hello
Hello
Hello
Hello