

FUNCTIONS:

fn is a block of code (a) set of instructions that can be called when needed.

2 types of fns. are there.

① Calling fn.

E.g.

Tha sum
+
calling fn.

Call 8 Kalyan
+
called fn.

② Called fn.

how to identify this is calling fn. (a) called fn.?

→ Calling fn. looks like the following :-

Name - of - the - fnk (arguments);

Name - of - the - fn();

E.g.

add();

add(5, 6);

add(a, b, c);

how 'called fn.' will look like?

return type - function_name()

{ Some block of code to perform task;

}

E.g. void main(), int main()

pre-defined funcⁿ: - already defined by the compiler
e.g.: printf, scanf, main(), void main(), int main()

user-defined funcⁿ:

These funcⁿ. can be created by user & for a specific task.

→ how called in source line
(general type)

return-type — function-name (parameter) ^{ex. (optional)}

{
 some block of code to perform task;
}

* how to create user-defined funcⁿ.

→ it called fn. is present & calling fn. may not be present then no problem.

BUT if calling fn. is present then CALLING fn. should be present.

* Inside main calling fn. &
before main calling fn.

* Creating a user-defined fn. 601
 #include <stdio.h>
 // all called fn. should be before main

```
void add()
```

```
{
```

```
    int a = 10, b = 20;
```

```
    printf("sum = %d", a + b);
```

```
}
```

```
void main()
```

```
{
```

// all calling fn. should be inside main.

```
    add();
```

```
}
```

OUTPUT:

30

Extra Typing

~~types of something.~~

~~(1) without argument & without return value~~

S.no	types of functions.	Called
①	Without argument	Without return value
②	Without argument	WITH return value
③	With argument	With OUT return value
④	With argument	WITH return value.