

Function

Getting something back from a function

Let's go back to the function `max` where we passed information about what integers the function shall work with

```
max (13, 14) ;
```

Printing things out to the screen or a file is not of any use to your program.

If the max function only printed to screen you could not do anything with the result of the function work.

Instead, we need the function to give us a value back that we can store in a variable.

Let's say we want to store the maximum value of two integers 13 and 14 in a integer variable called `max_value`.

Return value

Actually, it is trivial.

```
int max_value = max (13, 14);
```

Return value

A value, of some kind, returned by a function that the caller can store.

Return value vs. printf

Very few functions print something at all. C functions dealing with date, time, file, network etc very rarely print something.

Can a function both print and return?

Well, they generally don't But one exception to this is a function you've used tons of times....

Printf - the bastard?

`printf` actually prints to the screen (actually a stream). Its purpose is to print so that's nothing odd. But it also returns an integer value, the numbers of characters printed (useful when you need to make sure everything you wanted to print actually was printed).

Piece of advice

When writing new function you will most likely combine printf statements and a return value. But when you're done you better remove the print statements.

Return value

If the caller should be able to store your result

`printf` (and friends)

If you want things printed to the screen