

# malloc in C: Dynamic Memory Allocation in C Explained

freeCodeCamp.org

## What is malloc() in C?

malloc() is a library function that allows C to allocate memory dynamically from the heap. The heap is an area of memory where something is stored.

malloc() is part of stdlib.h and to be able to use it you need to use `#include <stdlib.h>`.

## How to Use Malloc

malloc() allocates memory of a requested size and returns a pointer to the beginning of the allocated block. To hold this returned pointer, we must create a variable. The pointer should be of same type used in the malloc statement.

Here we'll make a pointer to a soon-to-be array of ints

```
int* arrayPtr;
```

Unlike other languages, C does not know the data type it is allocating memory for; it needs to be told. Luckily, C has a function called `sizeof()` that we can use.

```
arrayPtr = (int *)malloc(10 * sizeof(int));
```

This statement used malloc to set aside memory for an array of 10 integers. As sizes can change between computers, it's important to use the `sizeof()` function to calculate the size on the current computer.

Any memory allocated during the program's execution will need to be freed before the program closes. To **free** memory, we can use the `free()` function

```
free( arrayPtr );
```

This statement will deallocate the memory previously allocated. C does not come with a **garbage collector** like some other languages, such as Java. As a result, memory not properly freed will continue to be allocated after the program is closed.

## Before you go on...

## A Review

Malloc is used for dynamic memory allocation and is useful when you don't know the amount of memory needed during compile time.

Allocating memory allows objects to exist beyond the scope of the current block.

C passes by value instead of reference. Using malloc to assign memory, and then pass the pointer to another function, is more efficient than having the function recreate the structure.

## More info on C Programming: