When to use the Heap ?

Since the Heap is a region of unregulated free space, its usage can be a bit complex. Let's discuss ways in which we can use it wisely.

We'll cover the following

When should you use the heap?

If you need to allocate a large block of memory (e.g. a large array, or a big struct), and you need to keep that variable around a long time (like a global), then you should allocate it on the heap.

If you need variables like arrays and structs that can change size dynamically (e.g. arrays that can grow or shrink as needed) then you will likely need to allocate them on the heap, and use dynamic memory allocation functions like `malloc()`, `calloc()`, `realloc()` and `free()` to manage that memory "by hand".

When should you use the stack?

If you are dealing with relatively small variables that only need to persist as long as the function using them is alive, then you should use the stack, it's easier and faster.

We will talk about dynamically allocated data structures after we talk about pointers, which is the topic for our next section.

Feel free to check out the links that follow. They contain a lot of insight into memory allocation.