## Stack vs Heap Pros and Cons

So how do the two structures fair against each other? The trick is to know where to appropriately use them. This lesson lists the pros and cons for both.

| Stack | Heap |
|---|---|
| very fast access | (relatively) slower access |
| local variables only | variables can be accessed globally |
| limit on stack size (OS-dependent) | no limit on memory size |
| variables cannot be resized | variables can be resized using `realloc()` |
| don’t have to explicitly de-allocate variables | you must manage memory (you’re in charge of allocating and freeing variables) |
| space is managed efficiently by CPU, memory will not become fragmented | no guaranteed efficient use of space, memory may become fragmented over time as blocks of memory are allocated, then freed |

With the theory out of the way, let’s take a look at how the Stack and Heap work in code.