|  |  |
| --- | --- |
| **What is the stack?** | It's a special region of your computer's memory that stores temporary variables created by each function (including the main() function). The stack is a "LIFO" (last in, first out) data structure, that is managed and optimized by the CPU quite closely. |
| **What is the heap?** | The heap is a region of your computer's memory that is not managed automatically for you, and is not as tightly managed by the CPU. It is a more free-floating region of memory (and is larger). |
| **Stack v Heap** | **Stack:**  very fast access  don't have to explicitly de-allocate variables  space is managed efficiently by CPU, memory will not become fragmented  local variables only  limit on stack size (OS-dependent)  variables cannot be resized  **Heap:**  variables can be accessed globally  no limit on memory size  (relatively) slower access  no guaranteed efficient use of space, memory may become fragmented over time as blocks of memory are allocated, then freed  you must manage memory (you're in charge of allocating and freeing variables)  variables can be resized using realloc() |

References:

* <https://gribblelab.org/CBootCamp/7_Memory_Stack_vs_Heap.html>