Arrays vs. vector

Why use an array if `vector` is so much more convenient? Why bother learning about arrays at all? Here's why.

- `vector` **elements** are always ==**allocated on the heap**==

- Arrays may be ==**allocated on the stack**==, ==**static area**==, ==**or the heap**==. This avoids performance issues that arise with dynamic memory.

- Arrays often have ==**higher performance**== and ==**take less memory**==.

- Arrays are generally used for low-level **systems programming** (operating systems)

- Array performance is ==**deterministic**==; for this reason, arrays are normally used for **embedded programs** that must run for long periods of time.

In short, arrays are usually faster and may take less memory than dynamic library types like `string` and `vector`. Using arrays in C++ is programming ==**as the CPU sees it**==.