LinkedIn article about variables allocation in stack and heap for both value and reference types:

link: `https://shorturl.at/d17Tn`

Compiled vs Interpreted Languages: Understanding C#'s Hybrid Approach

Compiled Languages: These translate the entire source code into machine code before running, producing an executable file that the CPU runs directly. Examples include C, C++, Rust, and Go.

- **Advantages:**
  - Faster execution since code is precompiled
  - Errors caught at compile time
  - Executables run independently without source code
  - Better optimization
- **Disadvantages:**
  - Compilation slows development
  - Platform-specific executables require recompilation
  - Changes need recompiling

Interpreted Languages: These execute code line-by-line at runtime without producing a separate executable. Examples include Python, traditional JavaScript, Ruby, and PHP.

- **Advantages:**
  - Immediate execution, no compile wait
  - Platform independence if interpreter is available
  - Easier debugging and flexibility
- **Disadvantages:**
  - Slower runtime execution
  - Errors found only when lines execute
  - Requires interpreter installed
  - Less optimization