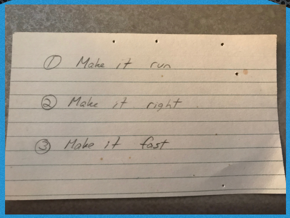
Virtuous Code Optimization

Paul Lefebvre
Xojo, Inc.

Donald Knuth

"Premature code optimization is the root of all evil"


① Make it run
② Make it right
③ Make it fast

Kent Beck

Make it Work

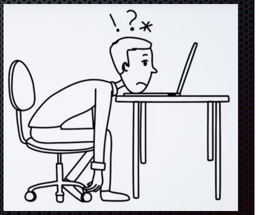
- Get something to happen
- Proof of concept
- It compiles!
- It runs!
- It does what you expect!
- Once
- ~~Ship it!~~

Make it Right

- Improve the code
- Error handling
- Edge cases
- Testing!

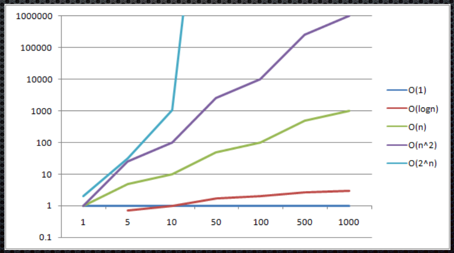
Make it Fast

- Relies on prior steps!
- May already be fast enough
- Ensures you have a baseline to compare with
- Risks
  - Can waste precious developer time
  - Can introduce bugs
  - Improvements may not be noticeable
- How to get started?



Big O Notation

- A way to describe an algorithm performance
- Popularized by Donald Knuth
- O(1), O(n), O(n^2), O(log n)



Profiler

- Help you find code with possible performance concerns
- Use directly from IDE
  - Project -> Profile Code
  - **StartProfiler**, **StopProfiler** commands
- Use in built apps
  - Creates Text File
  - Kem Tekinay's open-source viewer
    - github.com/ktekinay/Profile-Reader
  - docs.xojo.com/UserGuide:Code_Profiler

Demo

Tips

Switch Algorithms

- Remember Big O
- Find another algorithm that is faster
- Drastic difference with Sorting
  - Bubble Sort O(n^2)
  - Merge Sort O(n log n)

64-bit, Aggressive

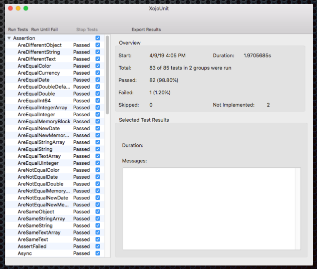
- Switch to 64-bit with its optimizing LLVM compiler
- Great for math-heavy code
- Only use Aggressive mode for final builds and performance testing
  - Avoid for general development as building is often slower

Inline Methods

- Methods that are called a lot can have a measurable performance hit
  - Due to compiler stack management
- Inline method to eliminate this
  - Copy method code within another method
  - Not as re-usable and risks bugs
  - Use judiciously

Unit Testing

- Important to ensure that optimizations have the same results
- Xojo Unit
  - Free and open-source
  - github.com/xojo/XojoUnit
- Displays timing for test methods



Assertion		✓
AreDifferentObject	Passed	✓
AreDifferentString	Passed	✓
AreDifferentText	Passed	✓
AreEqualColor	Passed	✓
AreEqualCurrency	Passed	✓
AreEqualDate	Passed	✓
AreEqualDoubleDefa...	Passed	✓
AreEqualDouble	Passed	✓
AreEqualInt64	Passed	✓
AreEqualIntegerArray	Passed	✓
AreEqualInteger	Passed	✓
AreEqualMemoryBlock	Passed	✓
AreEqualNewDate	Passed	✓
AreEqualNewMemor...	Passed	✓
AreEqualStringArray	Passed	✓
AreEqualString	Passed	✓
AreEqualTextArray	Passed	✓
AreEqualUInteger	Passed	✓
AreNotEqualColor	Passed	✓
AreNotEqualDate	Passed	✓
AreNotEqualDouble	Passed	✓
AreNotEqualMemory...	Passed	✓
AreNotEqualNewDate	Passed	✓
AreNotEqualNewMe...	Passed	✓
AreSameObject	Passed	✓
AreSameStringArray	Passed	✓
AreSameString	Passed	✓
AreSameTextArray	Passed	✓
AreSameText	Passed	✓
AssertFailed	Passed	✓
Async	Passed	✓

Overview

Start: 4/9/19 4:05 PM Duration: 1.9705685s
Total: 83 of 85 tests in 2 groups were run
Passed: 82 (98.80%)
Failed: 1 (1.20%)
Skipped: 0 Not Implemented: 2

Selected Test Results

Duration:

Messages:

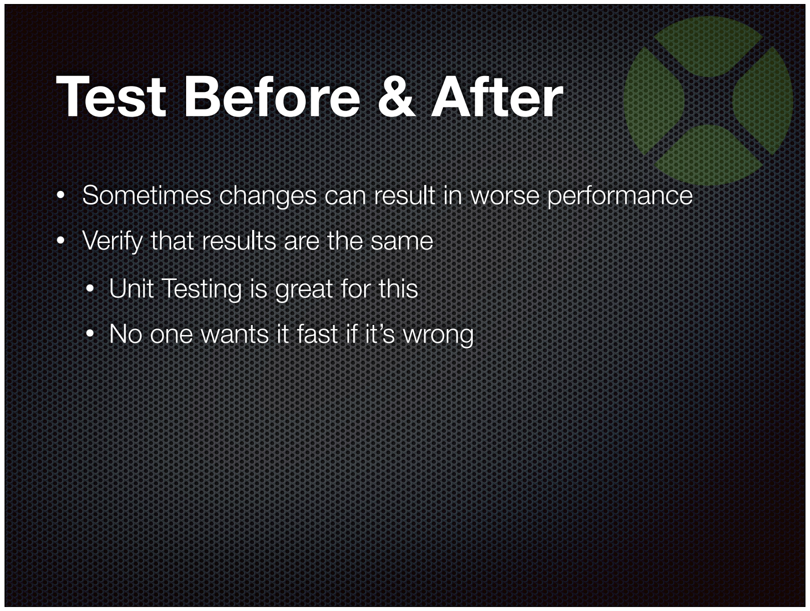
Reduce Loop Calculations

- Loops are primary source of performance problems
- Reduce calculations done within a loop
  - Identify "invariants" and set them outside of loop
    - Invariant: Value that does not change
  - Dim variables outside loops

Reduce Loop Calculations

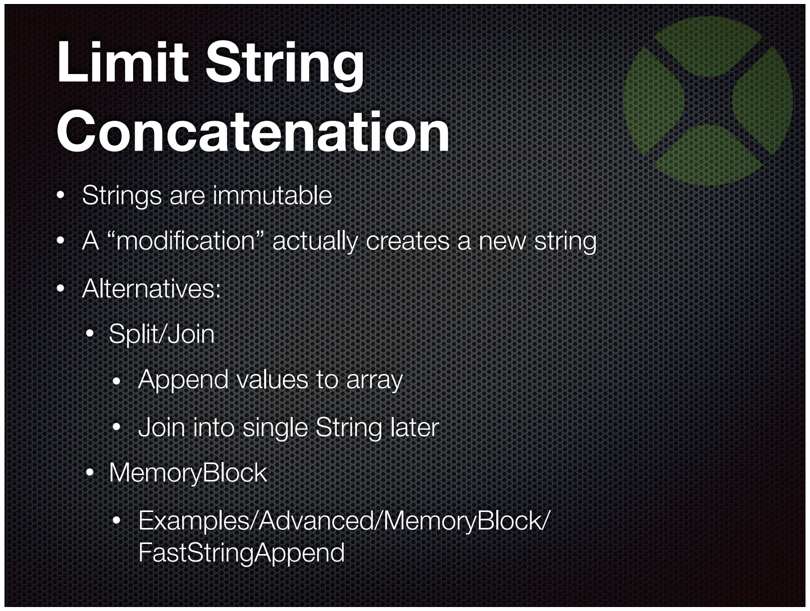
```
Do
  Dim specialValue As String
  specialValue = GetValue
  Dim value As Boolean
  value = DoOtherStuff(specialValue)
Loop Until value = True
```

```
Dim specialValue As String
specialValue = GetValue
Dim value As Boolean
Do
  value = DoOtherStuff(specialValue)
Loop Until value = True
```



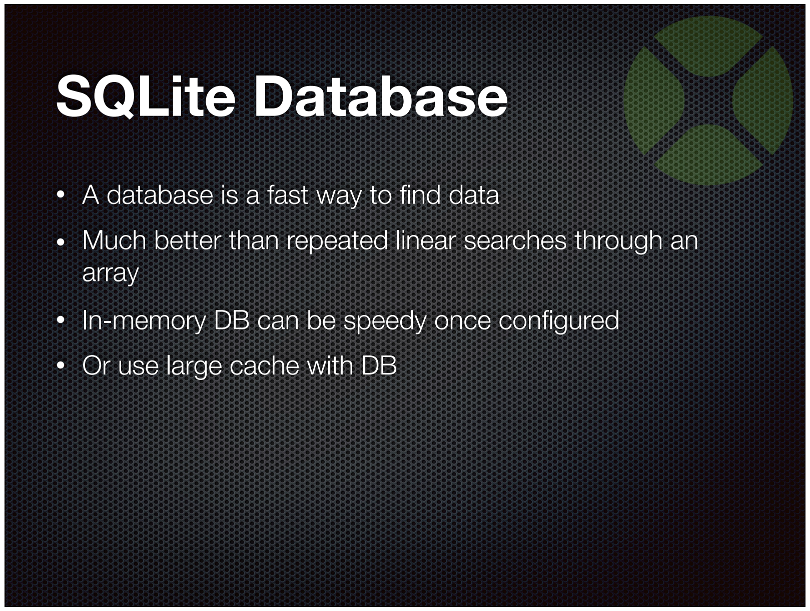
Test Before & After

- Sometimes changes can result in worse performance
- Verify that results are the same
  - Unit Testing is great for this
  - No one wants it fast if it's wrong



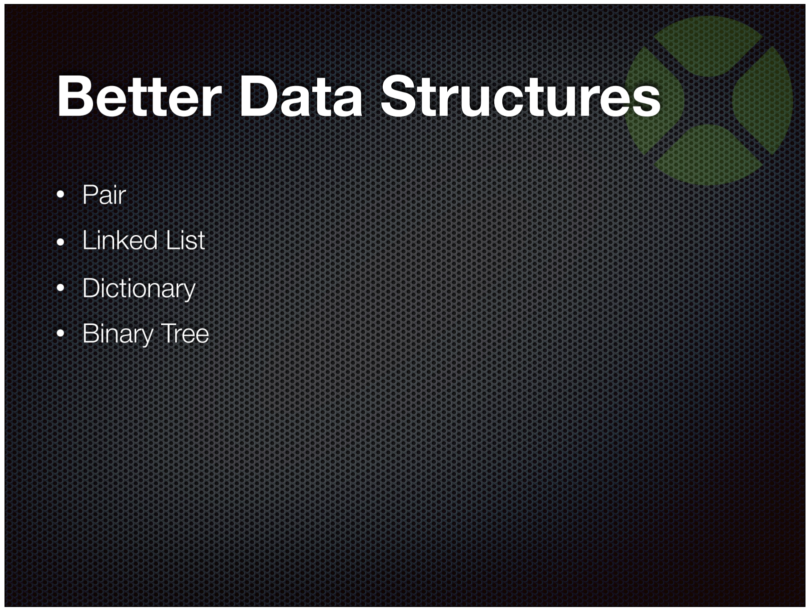
Limit String Concatenation

- Strings are immutable
- A "modification" actually creates a new string
- Alternatives:
  - Split/Join
    - Append values to array
    - Join into single String later
  - MemoryBlock
    - Examples/Advanced/MemoryBlock/FastStringAppend



SQLite Database

- A database is a fast way to find data
- Much better than repeated linear searches through an array
- In-memory DB can be speedy once configured
- Or use large cache with DB



Better Data Structures

- Pair
- Linked List
- Dictionary
- Binary Tree

Don’t Be Evil — Be Virtuous

- Optimize only after things are working
- Start with Profiler
- Unit Test to verify Results
- Apply tips as appropriate

Q & A

Paul Lefebvre

paul@xojo.com

Give us feedback on this session in the XDC app!