What is an actually good scripter™?

Since the beginning of Roblox, there have been many modelers, designers and the most important part of the whole community is… the scripter.

The scripter is making the game feel alive, he controls everything.
If the scripter doesn't deliver, the game doesn't either.

What tools are you working with, as a scripter?

Roblox uses Luau, a fast and very optimized version of Lua 5.1
Luau shouldn't be dealt with like Lua or should it?

Type annotation

Type annotation and type checking is one of the most useful aspects of Luau, why?
In Luau it's practically impossible to create errors during runtime when using type annotation.
Before you're even analyzing the code during runtime, it already yells at you, at edit time.

Why is this so powerful?

Not only are you avoiding lots of errors during runtime but you're actually improving performance.

When running code in native you're actually providing the compiler with useful information.
When using no type annotation the compiler might confuse a Vector3 with a table.
(Example Here)

That does not guarantee type annotation will magically speed up everything, you should invest resources in actually improving the code, not the type annotation.

Bad Lua habits

As a former BeamNG.drive mod developer, I practically had to rethink how to write "Lua" code, since Luau has more optimizations than you actually think.

1. Caching Libraries

Caching Libraries (like math lib) is a common use case in Lua to cache functions for faster execution. Luau does not need it and isn't recommended either.

2. Using global variables

Using global variables isn't recommended, and you should rather use local variables.

3. Assigning __index a function

Assigning __index as a function is not recommended, and you should use a table for __index instead.

4. Using loadstring(), setfenv() or getfenv()

These functions actually force de-optimization for Luau Environments.
Thus, they're not recommended.

5. Using pairs()

You might be familiar with the pairs() iterator function in Lua. What's bad about it then? — You can just remove the pairs() entirely, removing it saves a pairs() call.

Conclusion

Should you really know and learn all of it?

Short answer:

Nah, absolutely not.
It's an absolute waste of time, see the long answer.

Long answer:

When creating any game, you're always confronted with the one task that is the most essential in every game you will ever play… the **Gameplay**.

Should you focus and use half of your time optimizing your code for your 2 players?

Hell no.

Should you create a super optimized data store system using buffers, Base94 and extremely optimized functions?
Hell no.

So… what should you *really* focus on?

Gameplay – the Player Experience
No player will care if your code is super optimized and 30K lines long and so optimized it could run on a smart fridge.

They only care about the actual experience, the fun, and the joy of playing your game, not the 0.000001% that play it on the Smart Fridge.

Focus on the balancing, Movement System, Tutorial, etc.
What the Player really wants.