Luau Learning Order

Why is learning to script so hard? Well, it's because you need to learn scripting in a specific order! If you're learning in the wrong order it feels impossible. To make this simple, I've put together this PDF that puts everything in order that you will need to become an advanced developer. The list is designed for you to google: roblox [list item]

If you don't want to look for tutorials on everything in this list, check out The Basics Of Roblox Luau; It covers everything in the beginner concepts section and ends with you making your first Roblox game.

Use Code LUAULEARNORDER for 10% off!

Happy coding! *-Seal*

Beginner Concepts

1. Getting Started
-> Variables
-> Types: strings, numbers, booleans, nil
-> Lexical Conventions: naming rules, case sensitivity, comments

2. Functions
-> Defining and calling functions
-> Arguments (parameters)
-> Returning values

3. Scope
-> What is Scope
-> Local vs global variables
-> How scope affects function variables

4. Replication & Script Types
-> Replication / Filtering Enabled
-> Difference between Scripts and LocalScripts

5. Events
-> Bindable Events
-> Bindable Functions
-> Remote Events
-> Remote Functions
-> Connections *(:Connect)*

6. Conditionals
-> If statements
-> Else
-> Elseif

7. Loops

-> **While loops**
-> **For loops**
-> **Repeat...until**
-> **Break and Continue**
-> **Pairs and Ipairs**

8. Tables

-> **Arrays**
-> **Dictionaries**
-> **Using pairs and ipairs with tables**
-> **next()**
-> **ModuleScripts (as data containers and organization tools)**

Advanced Concepts

1. Table Library

-> **Table.insert()**
-> **Table.find()**
-> **Table.remove()**
-> **Table.clone()**
-> **Table.clear()**
-> **Table.sort()**

2. Coroutines

-> **Task.Spawn**
-> **Task.Defer**
-> **Threads**

3. Garbage Collection

-> **Garbage Collection**
-> **Strong Reference vs Weak Reference**

4. HTTP Service

-> **JSON Strings**
-> **Pcalls**

5. Networking

-> **Sanity Checks**
-> **Rate Limiting Requests**
-> **Serialization of Data Across Server and Client**
-> **Exploit Protection**

6. DataStores

-> **Writing Keys and Values**
-> **Saving Data**
-> **Data Limits**
-> **Ordered DataStores**

7. Useful Tools For Making Games:

-> Raycasting
-> Region3
-> Debounces
-> Type and Typeof
-> Assert
-> Command Bar
-> String Library
-> Math Library
-> Pathfinding Service

8. Object Oriented Programming

-> Meta tables
-> Classes
-> Functions vs Methods

9. Structuring Your Project

-> Using Module Scripts
-> Using a Framework

10. Libraries

-> Promise
-> Maid
-> Signal
-> TestEZ
-> Fastcast
-> Zone+

Congratulations

Once you complete this list you're game developer ready! If you've made it to the bottom of this list and you still need help, send me a message on the contact form on my website: definitelynotseal.com/contact-page