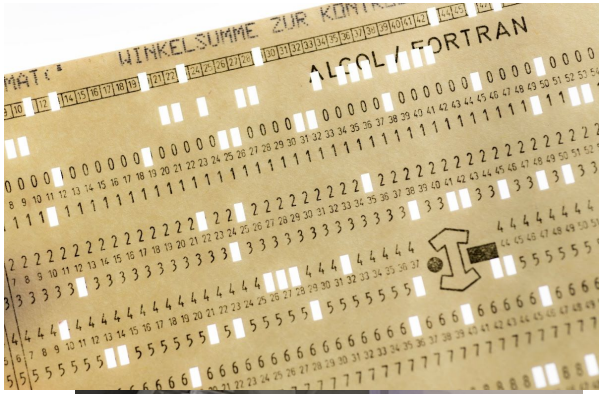
Roblox coding with Lua

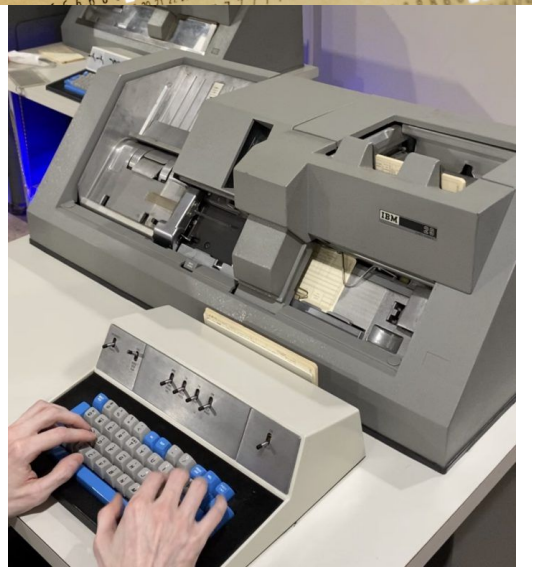
Lecture 1

A hacker's guide to them codez by @murrekatt.

What is Programming?

1. *Programming, coding, software engineering*
2. **Writing text (instructions) to do something**
    a. software instructs hardware
    b. software instructs other software
3. “Text” to be translated into machine code
4. Many different programming languages - why?
5. **Great for problem solving-thinking training!**



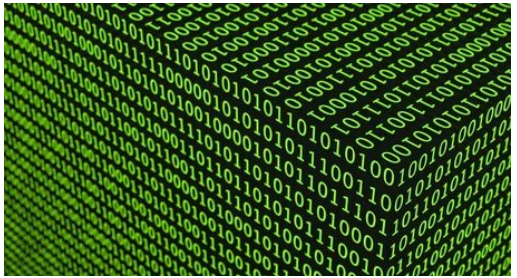


How a computer runs code

1. Software is compiled code
    a. Various intermediate steps
2. Compiled code is what the CPU understands
    a. Hardware architecture matters!
3. Zeros and Ones — binary — on or off elec
   in small circuits (chips)
4. Computer loads compiled code into memory
   to run it

```
return (function ()

    local function tdump(t)
        local search = pairs;
        local sf = string.format;
        local sr = string.rep;
        local perf_type = type;
        local ts = tostring;
        local function dmp(t, l, k)
            if perf_type(t) == "table" then
                local msg = sf("%s%s:", sr(" ", l*2), ts(k));
                print(msg);
                for k, v in search(t) do
                    dmp(v, l+1, k);
                end
            else
                local msg = sf("%s%s:%s", sr(" ", l*2), ts(k), ts(t));
                print(msg);
            end
        end
        dmp(t, 1, "root");
    end

    return {
        tdump = tdump;
    };
end)(...)
```

```
.text:00008768  e51b2038        ldr r2, [fp, #-56]
.text:0000876c  e51b301c        ldr r3, [fp, #-28]
.text:00008770  e300378f        mov r3, #1935
.text:00008774  e50b303c        str r3, [fp, #-60]
.text:00008778  e51b3020        ldr r3, [fp, #-32]
.text:0000877c  e2832001        add r2, r3, #1
```



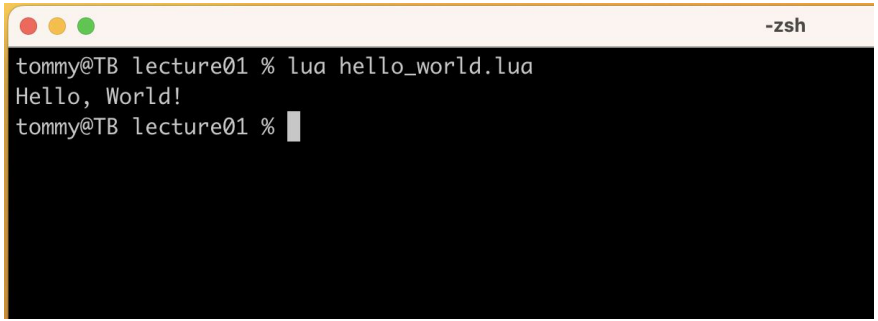
What is Lua?

1. **Simple and minimalistic language to be embedded**
2. Used by many platforms and other software
    a. Roblox, World of Warcraft, Angry Birds
    b. nginx, plugins, extension, ...
3. Not compiled to machine code, but **interpreted at runtime**
    a. Lua VM (virtual machine) used by Roblox
4. **A great language to get started with programming**

Lua in action

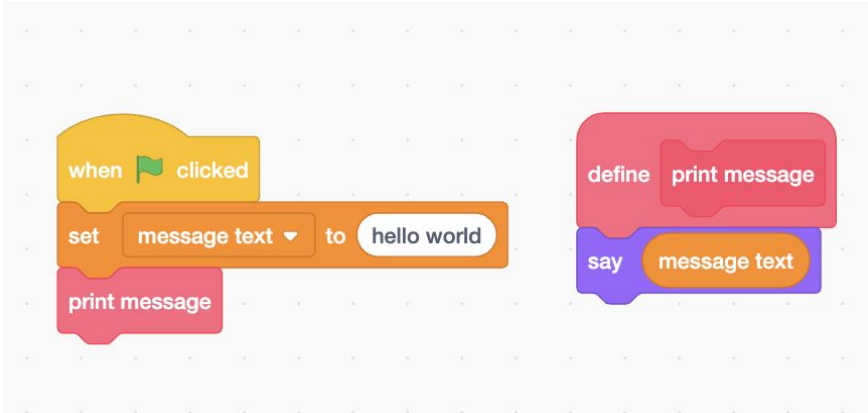
```
$ brew install lua

$ code hello_world.lua

$ lua hello_world.lua
```

```lua
local message_text = "Hello, World!"
local function print_message()
    print(message_text)
end
print_message()
```



Lua vs Scratch

```lua
local message_text = "Hello, World!"
local function print_message()
    print(message_text)
end
print_message()
```




```
-zsh
tommy@TB lecture01 % lua hello_world.lua
Hello, World!
tommy@TB lecture01 %
```

Actors vs Sequential

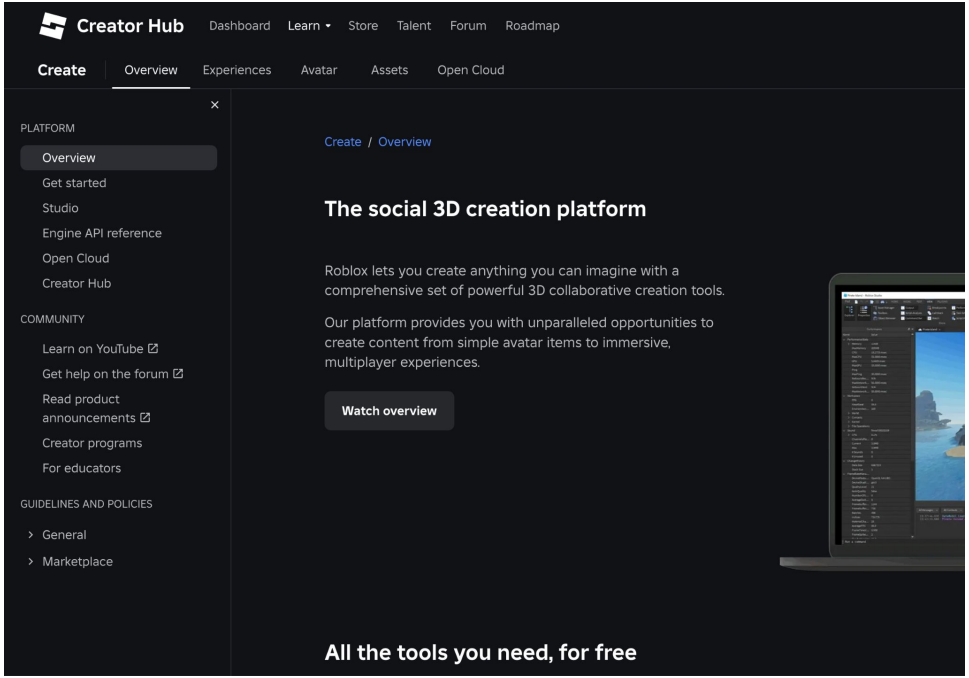
1. **Actors run own code and interact with other actors**
    a. Objects (Roblox)
    b. Sprites (Scratch)
2. **...this is a decentralized and concurrent model**
    a. “Things happen at the same time”
3. **Sequential**
    a. “Things happen one after the other in a fixed order”

Roblox and Lua

1. Roblox runs Lua inside itself
    a. Lua VM
    b. Luau, a Lua derivative... (more on this later)
2. **Lua scripts attached to objects**
    a. Scratch code attached to sprites
3. Lua is very simple...
4. ...But Roblox provides many things ready to use

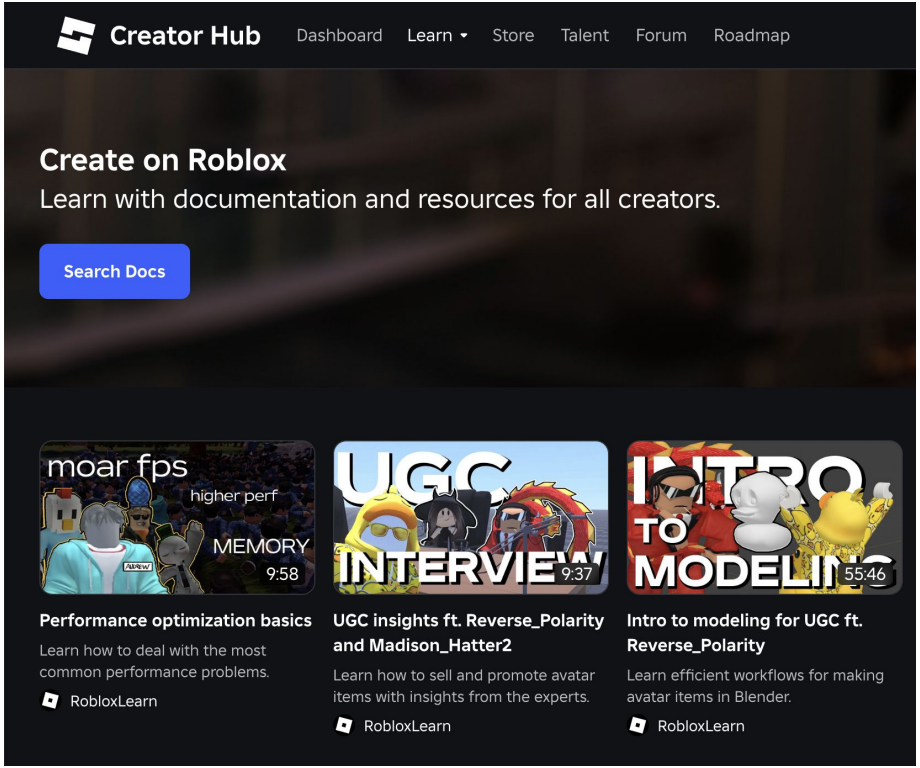
Roblox APIs

1. Roblox apps use Roblox API
2. https://create.roblox.com/docs/en-us/platform



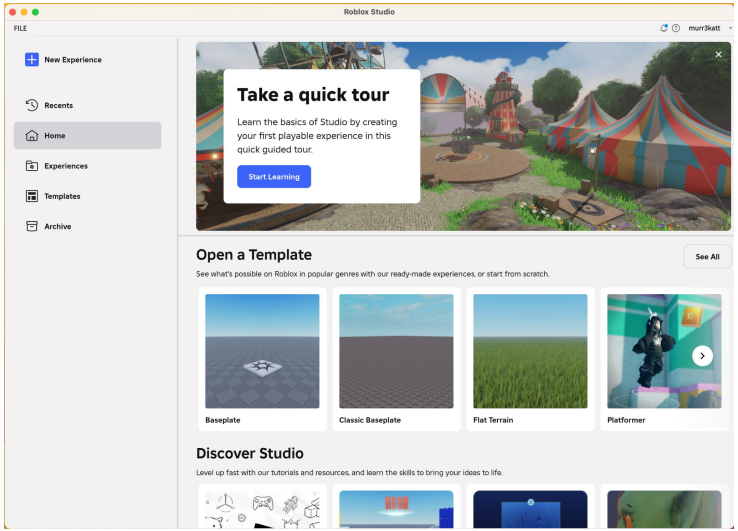
Roblox Docs

1. Great tutorials
2. https://create.roblox.com/docs



Roblox Studio in action

1. **Open Roblox Studio**
2. New experience
3. Add part and edit script:




```
local part = script.Parent

part.BrickColor = BrickColor.Red()

print("Hello, World!")
```

Why Lua matters

1. **Simple and powerful**
    a. Tables are Lua's magic data structure
2. Embeddable in other software
3. **Flexibility!**

Variables

1. Store and manipulate data (“memory”)
2. Dynamically typed

```
local part = script.Parent
local color = "Red"
local size = 5
print("Color:", color, "Size:", size)
part.BrickColor = BrickColor.new(color)
part.Size = Vector3.new(size, size, size)
```

Conditionals (if-then-else)

1. Controls flow...

```
local part = script.Parent
local playerCount = game.Players.NumPlayers
if playerCount > 0 then
    part.BrickColor = BrickColor.new("Green")
    print("Players in game! Sphere is green.")
else
    part.BrickColor = BrickColor.new("Red")
    print("No players. Sphere is red.")
end
```

For-loops

Repeat for all in a list...

```
local part = script.Parent
for i = 1, 5 do
    print("Loop iteration:", i)
    wait(1) -- Wait 1 second
    part.Transparency = i / 5 -- Fade sphere
end
```

While-Loops

Repeats until condition no longer true

```
local part = script.Parent
for i = 1, 5 do
    print("Loop iteration:", i)
    wait(1) -- Wait 1 second
    part.Transparency = i / 5 -- Fade sphere
end
```

Tables

Lua's core data structure ("memory")

```lua
local part = script.Parent
local colors = {"Red", "Blue", "Green"} -- Array-like table
local properties = {size = 5, isVisible = true} -- Dictionary-like table
for i, color in ipairs(colors) do
    print("Setting color:", color)
    part.BrickColor = BrickColor.new(color)
    wait(1)
end
part.Size = Vector3.new(properties.size, properties.size, properties.size)
print("Size set to:", properties.size)
```

Funcations

Reusable code blocks

```
local part = script.Parent
local function changeColor(newColor)
    part.BrickColor = BrickColor.new(newColor)
    print("Changed to:", newColor)
end
changeColor("Red")
wait(1)
changeColor("Blue")
```

Events

Handle interactions with other things and what happens

“How everything talks to each other”

```
local part = script.Parent
part.Touched:Connect(function(hit)
    part.BrickColor = BrickColor.new("Yellow")
    print("Sphere touched by:", hit.Parent.Name)
end)
```

Next steps and Q&A

1. **Lua coding lesson 2**
    a. Lua coding syntax and how it works
    b. Github (clone a repo from me)
2. Roblox platform how it works and how to use it
3. **Self-study**
    a. Roblox Docs
    b. Roblox APIs