\*Cấu trúc GDScript cơ bản:

**extends Node** # Khai báo kế thừa. Phải kế thừa đúng tên lớp (tên Node) mà bạn sử dụng.

**var a = 0** # Biến toàn cục

**var b = 0** # Biến toàn cục

**func \_ready():** # Hàm

**var c = a + b** # Biến cục bộ

\*GDScript cơ bản

Các hàm chính:

**extends Node**

**func \_init():** #Thực thi một lần duy nhất khi khởi chạy chương trình.

**.     pass**

**func \_ready():** #Thực thi một lần duy nhất khi Node sẵn sàng hoạt động.

**.     pass**

**func \_process(delta):** #Hàm luôn thực thi lặp lại trong mỗi thời gian delta bằng nhau (không phùhợp cho các hoạt động vật lí)

**.     pass**

**func \_physics\_process(delta):** #Hàm thực thi lặp lại trong mỗi thời gian delta tăng dần (thích hợpcho hoạt động vật lí)

**.     pass**

+Lệnh Debug trên cửa sổ Console

**func \_ready():**

**.     var a = 4**

**.     print(a);**

**.     print("\nHello World")**

Output:

**4**

**Hello Word**

+Kiểu dữ liệu,biến ,hằng và cách khai báo.

**extends Node**

#var <tên biến> : [kiểu dữ liệu] (có thể "=" để gán giá trị trực tiếp mà không cần khai báo kiểu dữ kiệu (ép kiểu ngầm))

**var a = 4**

**var a: int = 4**

**var a**

**a = 4**

#các kiểu khai báo trên là tương tự như nhau. Ngoài ra ta còn có thể khai báo **var a:=4.**

**const a = 4** #khai báo hằng. Lưu ý phải gán giá trị cho hằng.

**var ab = Vecto2(0,0)** # Biến kiểu lớp

**var arr = [1, 2, 3]** #khai báo mảng

**var d = {2: 3, 4: "Hello"}** #khai báo từ điển

+Cấu trúc điều kiện IF…THEN…ELSE

func \_ready():

#     if(<điều kiện>):

#          [tập lệnh phía sau dấu tab phải thẳng hàng]

#     else:

#          [tập lệnh phía sau dấu tab phải thẳng hàng]

.     var a = false

.     var b = 0

.     if(a):

.          b = 1

.     else:

.          b = 2

+Cấu trúc Match

func \_ready():

.     var a = 0

.     Match(a):

.          1:

.               [tập lệnh phía sau dấu tab phải thẳng hàng]

.          2:

.               [tập lệnh phía sau dấu tab phải thẳng hàng]

+Vòng lặp FOR

func \_ready():

.     for i in 5:

.          [tập lệnh phía sau dấu tab phải thẳng hàng]

.     for i in range(5):

.          [tập lệnh phía sau dấu tab phải thẳng hàng]

#i chạy từ 0 đến 5-1

+Vòng lặp WHILE

func \_ready():

.     while(<điều kiện>):

.          [tập lệnh phía sau dấu tab phải thẳng hàng]

+Hướng đối tượng trong GDScript

Tạo lớp:

class Math: #tên lớp

.      var total #thuộc tính

.      static func \_sum(a: int, b: int) -> int: #phương thức

.          return a+b

Sử dụng lớp:

#var <tên biến file script> = preload("đường dẫn script")

var MathGD = preload("Math.gd")

func \_ready():

#     var <tên biến> = <tên biến script>.<tên lớp>.new()

.     var a = MathGD.Math.new()

.     print(a.sum(1,2))

Output: 3