

# 스크립트 가이드라인

## 기본 구조

```
function Start()

    --이 부분은 실행시 최초 한번만 호출됩니다.

end

function Update()

    --이 부분은 실행시 프레임마다 반복적으로 호출됩니다.

end
```

반드시 Start() 와 Update()를 작성할것

## 함수 기능

### 0. 기본 사용법

예제

```
local obj2 = Script:GetThisObject();
local trans = obj2:GetTransform();
function Start()

    print(obj2:GetName());
    obj2:LoadAnimPreset("man");

end

function Update()

    obj2:PlayAnim2D("idle",0);

    if (KeyInput_W() == true) then
        trans:MoveUp();
    end

    if (KeyInput_A() == true) then
        trans:MoveLeft();
        obj2:PlayAnim2D("run",1);
    end

    if (KeyInput_S() == true) then
```

```

        trans:MoveDown();
    end

    if (KeyInput_D() == true) then
        trans:MoveRight();
        obj2:PlayAnim2D("run",0);
    end

end

```

## 1. 전역함수

### KeyInput

```

bool KeyInput_W()
bool KeyInput_A()
bool KeyInput_S()
bool KeyInput_D()

```

- 키보드 입력상태를 반환합니다. (입력시 True를 반환)
- W,A,S,D의 입력을 지원합니다.

```

if (KeyInput_W() == true) then
    -- 내부 로직 구현
end

```

### DeltaTime

```

float GetDeltaTime()

```

- 해당 프레임의 DeltaTime을 반환합니다.

## 2. GameObjectManager

```

GameObject GameObjectManager:GetObject(String name)

```

- 이름을 입력받아 해당이름의 GameObject를 찾아서 반환합니다. 없으면 새 오브젝트를 생성합니다.

```

local obj = GameObjectManager:GetObject("hello");

```

## 3. GameObject

### 사용법

```

obj:[함수명] 의 형식으로 사용한다.
-- obj는 GameObject 타입의 인스턴스를 의미

```

## 예시

```
local obj = GameObjectManager:GetObject("hello");
obj:GetName();
```

## 함수

```
string GetName()
```

- 오브젝트의 이름을 가져옵니다.

```
void LoadAnimPreset(String name)
```

- 오브젝트에 미리 사전에 정의된 애니메이션 프리셋을 로드합니다.

```
void AddAnim2D(String name,int x,int y,int width, int height, int count, float holdTime)
```

```
-- name : 애니메이션 이름
-- x : 시작 x좌표
-- y : 시작 y좌표
-- width : 스프라이트 너비 (슬라이싱 px)
-- height : 스프라이트 높이 (슬라이싱 px)
-- count : 프레임의 수
-- holdTime : 단일 프레임당 시간 (ms)
```

- 오브젝트에 2D 애니메이션을 추가합니다.

```
void PlayAnim2D(String name)
```

```
-- name : 애니메이션 이름
```

- 해당 애니메이션을 재생합니다.

```
bool IsActive()
```

- 해당 오브젝트의 활성화 상태여부를 반환합니다.

```
void SetActive(bool value)
```

- 해당 오브젝트의 활성화 상태를 설정합니다.

```
Transform GetTransform()
```

- 해당 오브젝트의 Transform 을 반환합니다.

## 4. Transform

### 사용법

```
trans: [함수명]  
-- trans는 Transform 타입의 인스턴스를 의미
```

### 예시

```
local trans = obj2:GetTransform();  
trans:MoveUp();
```

### 함수 목록

#### Move

```
trans:MoveUp()  
trans:MoveDown()  
trans:MoveLeft()  
trans:MoveRight()
```

- 위치를 해당 방향으로 이동시킵니다.

```
local obj2 = Script:GetThisObject();  
local trans = obj2:GetTransform();  
  
function Update()  
  
    if (KeyInput_W() == true) then  
        trans:MoveUp();  
    end  
  
end
```

#### Translate

```
void Translate(float x, float y, float z)
```

- 현재 좌표에 입력된 좌표값을 더합니다.

#### Position

```
void SetPosition(float x, float y, float z)
```

- 현재 좌표를 입력된 좌표값으로 변경합니다.

## 5. AudioManager

### 사용법

```
AudioManager:[함수명]
```

### 예시

```
AudioManager:PlayAudio("walkSound")
```

### LoadAudio

```
void LoadAudio(string name, string path)
```

- 매개 변수: 저장할 이름 (str), 경로(str)
- 해당 경로에서 음원 파일을 가져와 name의 이름으로 등록합니다.

### PlayAudio

```
void PlayAudio(string name)
```

- 매개 변수: 오디오 이름(str)
- 등록된 name의 오디오를 재생합니다.

### StopAudio

```
void StopAudio(string name)
```

- 매개 변수: 오디오 이름(str)
- 현재 재생중인 name의 오디오의 재생을 중단합니다.

### DeleteAudio

```
void DeleteAudio(string name)
```

- 매개 변수: 오디오 이름(str)
- 등록된 name의 오디오를 오디오 목록에서 제거합니다.

### SetAudioVolume

```
void SetAudioVolume(string name,int volume)
```

- 매개 변수: 오디오 이름(str), 설정 볼륨(int [0~10])
- 등록된 name의 오디오의 소리 크기를 설정합니다.

### PauseAudio

```
void PauseAudio(string name)
```

- 매개 변수: 오디오 이름(str)
- 현재 재생중인 name의 오디오를 일시정지 시킵니다.

## ResumeAudio

```
void ResumeAudio(string name)
```

- 매개 변수: 오디오 이름(str)
- 재생이 중지된 name의 오디오의 재생을 재개합니다.