

# 목차

- 01. print()
- 02. 주석
- 03. 변수
- 04. 연산자
- 05. 테이블
- 06. true/false
- 07. 조건문
- 08. 반복문
- 09. 함수
- 10. 변수의 범위

[오늘부터 게임개발] 시리즈에서는 Lua라는 프로그래밍 언어를 사용해서 게임을 개발합니다. 이 부록에서는 그 Lua를 다루는 방법을 다루게 됩니다.

여러분이 다른 공부를 할 때처럼 모든 것을 암기할 필요는 없습니다. 간단하게 살펴본 다음에, 나중에 필요할 때 다시 찾아서 살펴보는 과정을 여러 번 겪으면 자연스럽게 여러분에게 상식으로 자리잡게 될 것입니다.

## 01 print()

### print()란?

게임 제작 과정에서 화면 상으로 확인할 수 없는 부분들을 확인하기 위해서 메시지를 띄울 수 있습니다. 예를 들면 상점에서 아이템을 구입하고, 이를 확인하기 위해 '사과 1개를 구입했습니다!' 라는 메시지를 출력할 수 있습니다.

이 메시지는 개발자를 위한 창인 콘솔창에 출력되기 때문에, 게임을 다운로드받아 플레이하는 유저에게는 보이지 않습니다.

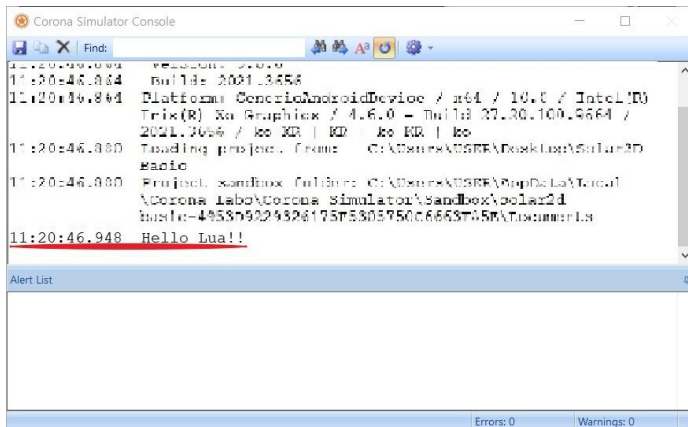
print()는 게임 플레이를 위한 기능이 아니라 개발자를 위한 기능입니다.

### print() 사용 방법

```
print( "Hello Lua!!" )
```

print( ) 안에 출력하기를 원하는 문장을 " (큰 따옴표) 혹은 ' (작은 따옴표)로 묶어 넣어주면 됩니다.

### 실행 결과



#### ▲ 콘솔창에 “Hello Lua!!”를 출력한 모습

콘솔창에 알 수 없는 영어들이 많지만, 우리가 출력한 문장만 찾아보도록 합니다. 앞으로는 콘솔창 캡처 화면 없이 출력 결과만 담았습니다.

## 02 주석

### 주석이란?

주석은 코드 중간에 메모를 작성하기 위해서 사용합니다. 복잡하고 긴 코드를 작성하게 될 때나 다른 친구와 같이 게임을 만들 때 코드 중간중간 주석을 사용하면 코드를 한눈에 알아보기 쉬워집니다.

또한 작성한 코드를 주석으로 만들어주면, 해당 코드는 프로그램 실행 시 작동하지 않습니다. 주석처리도 프로그램 실행과는 전혀 상관이 개발자의 편의성을 위한 기능입니다.

### 주석 사용 방법

- 1| -- 이건 주석입니다.
- 2| -- 여기에 쓰는 코드는 실행되지 않습니다.
- 3| print("hello Lua!!")
- 4| print("hello Solar2D!!")

앞에 --만 달아주면 그 뒷부분은 주석으로 처리됩니다.

### 실행 결과

Hello Solar2D!!

주석처리된 3| 코드는 실행되지 않아 "hello Lua!!"가 출력되지 않았고, 4| 코드만 실행이 되어 "hello Solar2D"만 출력이 되는 것을 확인할 수 있습니다.

## 03 변수

### 변수란?

프로그래밍에서 변수란 **변하는 어떤 값을 담을 수 있는 상자**라고 볼 수 있습니다. 게임을 진행하면서 계속 변화하는 어떤 값(Ex. 체력, 점수 등)을 변수에 담아뒀다가 사용하게 됩니다. 변수 안에 값이 계속 바뀌더라도 우리는 코드를 작성할 때 그 값을 담는 변수만 지칭하면 우리가 원하는 값을 사용할 수 있습니다.

### 변수 사용 방법

```
1| local HP      -- 변수 선언
2| HP = 100      -- 변수에 값 대입
3| local MP = 100 -- 변수 선언과 동시에 값 대입
4| local bg = display.newImage("img/bg.png")
```

변수를 사용하려면 먼저 선언을 한 뒤에 사용해야 합니다. 1| 첫번째 줄 코드처럼 앞에 **local**을 붙이고 원하는 **변수의 이름**을 지정해주면 해당 이름으로 된 변수 상자가 생성이 되어, 뒤에서 변수를 사용할 수 있게 됩니다.

2| 코드처럼 변수를 선언한 뒤에 **=**를 사용해서 변수에 값을 넣어줄 수 있습니다. 여기서 **=**는 ‘오른쪽의 값을 왼쪽의 변수에 넣어라(값 대입)’라는 의미입니다.

3| 코드처럼 변수의 선언과 대입은 한 줄로 동시에 작성할 수 있습니다.

4| 코드는 본편에서 사용해봤던 `display` object를 생성해서 변수에 담아두는 코드입니다.

### 변수에 담을 수 있는 값의 종류

변수에 담을 수 있는 값의 종류로는 대표적으로 정수(-1, 0, 1 ..), 실수(소수)와 문자열(“hello Lua!”), **변수가 비어있음을 뜻하는 nil**이 있습니다.

부울(true/false)도 있지만, 이는 따로 06 true/false에서 다룹니다.

## [0] [4] 테이블

### 테이블이란?

테이블은 번호표(index)를 가지는 변수들의 집합입니다. 같은 계열의 데이터들을 좀 더 편리하게 저장하고 관리하기 위해서 사용합니다.

index는 1부터 시작하는 정수로 사용하면 되고, [0] [8] 반복문과 함께 사용하면 더 간편하게 코드를 작성할 수 있습니다.

### 테이블 사용하기

```
1| local score = {}    -- 테이블 변수 선언
2| score[ 1 ] = 95
3| score[ 2 ] = 25
```

테이블도 일종의 값이므로, 1| 코드처럼 테이블을 담을 변수를 먼저 선언해야 합니다. 그 다음 변수 이름 index 형식으로 여러 개의 변수를 사용할 때마다 선언할 필요없이 사용할 수 있습니다.

### +) 테이블 속에 테이블

테이블 안에 테이블을 생성해주면 index를 여러 개 붙여 다차원 테이블로 사용할 수 있습니다.

```
1| local table = {}
2| table[1] = {}      -- 테이블 속 테이블 생성
3| table[1][1] = 100  -- 다차원 테이블 사용
4| table[1][2] = 200
```

## 05 연산자

### 연산자란?

연산자를 이용해 값을 변화시킬 수 있습니다. 연산자에는 우리가 익숙한 수학 연산자(+ - \* / 등)와 관계 연산자, 논리 연산자 그 외에 몇 가지 연산자가 있습니다. 여기서는 간단한 수학 연산자와 기타 몇 가지 연산자만 살펴봅니다.

### 연산자의 종류

#### 수학 연산자

더하기	+	$20 + 8 == 28$
빼기	-	$20 - 8 == 12$
곱하기	*	$20 * 8 == 160$
나누기	/	$20 / 8 == 2.5$
나머지	%	$20 \% 8 == 4$
제곱	^	$2^4 == 16$

#### 기타 연산자

..	문자열을 합쳐주는 연산자
#	테이블의 길이를 알려주는 연산자

# 담당 디자이너분께

표는 작년 버전 ppt에 넣은 표를 이미지로 따온겁니다..!

저번처럼 교재 디자인이랑 어울리게 적당히 넣어주시면 감사하겠습니다.

### 연산자 사용하기

```
1| local N = {}  
2| N[1] = 10  
3| N[2] = 8  
4|  
5| print( "N[1] + N[2] = " .. N[1] + N[2] )  
6| print( "N[1] - N[2] = " .. N[1] - N[2] )  
7| print( "N[1] * N[2] = " .. N[1] * N[2] )  
8| print( "N[1] / N[2] = " .. N[1] / N[2] )  
9| print( "N[1] % N[2] = " .. N[1] % N[2] )  
10| print( "테이블의 길이 = " .. #N )
```

### 실행 결과

```
N1 + N2 = 28  
N1 - N2 = 12  
N1 * N2 = 160  
N1 / N2 = 2.5  
N1 % N2 = 4  
테이블의 길이 = 2
```

연산자를 사용해본 예시 코드입니다.

문자열과 값을   연산자를 이용하여 한 줄로 출력했고, 테이블의 길이를 알려면

10| 코드처럼   연산자를 테이블 변수 앞에 붙여주면 알 수 있습니다.