

제 2강: 흐름 제어와 로직 (Control Flow & Logic)

1. 소개

단순한 입출력을 넘어, 상황에 따라 다르게 동작하거나 반복적인 작업을 처리하려면 '흐름 제어'가 필요합니다. 이번 강의에서는 조건문, 반복문, 그리고 함수에 대해 배웁니다.

2. 조건문 (If / Else)

가장 기본적인 로직입니다. "만약 ~라면 ... 해라"

```
local score = 80

if score >= 90 then
    print("A")
elseif score >= 70 then
    print("B")
else
    print("C")
end
```

꿀팁: Xilica에서는 1 을 true 로, 0 을 false 로 취급하는 경우가 많으니 비교할 때 주의하세요. `if InTable[1] == 1 then ... end`

3. 반복문 (Loops)

여러 핀을 한꺼번에 처리할 때 필수적입니다.

for 루프

```
-- 1부터 5까지 반복
for i = 1, 5 do
    print("핀 번호: " .. i)
    -- InTable[i] 처럼 활용 가능
end
```

ipairs (리스트 순회)

```
local commands = {"PWR", "VOL", "MUTE"}
for index, value in ipairs(commands) do
    print(index, value) -- 1 PWR, 2 VOL, 3 MUTE 순서로 출력
end
```

4. 함수 (Functions)

코드를 재사용하기 위해 묶어놓은 블록입니다.

```
-- 안전하게 숫자로 변환해주는 헬퍼 함수
local function SafeNum(val)
    local n = tonumber(val)
    if n == nil then return 0 end
    return n
end

local vol = SafeNum(InTable[1]) -- 사용하기
```

5. 실습 예제 (Exercise)

attach/course/exercises/ex02_logic_template.lua 파일을 열어 다음 미션을 수행하세요.

미션 목표:

1. 입력 1~5번까지 반복문(for)을 돌며 값을 읽으세요.
2. 각 값이 1(ON)이면 카운트를 1씩 올리세요.
3. 만약 켜진(1) 입력이 3개 이상이면 OutTable[1]에 "WARNING"을, 아니면 "OK"를 출력하세요.