

# 2조 회의

2021년 5월 8일

## 참석자

서준수 ----[C 언어 강좌-12] 함수 (FUNCTION)

설기현-----[프로그래밍 입문] Arrays and pointers-1

## 회의 내용

- 함수
- 1. 함수 이름을 정하는 것은 변수 이름을 정하는 것만큼이나 굉장히 중요(흐름 파악 등)
- 2. 함수 선언 시에는 각 매개변수의 데이터 타입만 명시하고 이름을 생략할 수 있음.  
  
그러나 함수 정의에서는 매개변수의 데이터 타입과 이름을 모두 명시해야 함.
- 3. { } 로 감싸진 블록마다 별도의 스택(저장) 영역이 생김. 예) **main** 스택에서 매개변수로 넣었던 a, b 변수의 값이 있음. {함수}의 스택이 생성되고 스택 영역의 지역변수로 a와 b가 새롭게 할당됨. 함수에서 a, b 변수의 값을 아무리 변경하여도 **main** 스택 영역의 변수들의 값은 그대로.
- 4. **return** 값은 한 개밖에 선언할 수밖에 없을까? 여러 개가 필요할 때는 구조체나 포인터를 통해 해결할 수 있음.
- 4-1. 구조체: **return** 값을 여러 개의 데이터 타입으로 구성된 구조체로 선언하면 원하는 만큼 다양한 결과값을 묶어서 반환할 수 있음. (잘 모르겠다)
- 4-2. 포인터: 매개변수를 포인터 변수로 선언해야 함. **int\*** 타입의 변수 선언(예를 들어 **int\*a**), 함수를 호출할 때 변수 a의 주소를 넘김. 함수 실행 시 **\*a**와 같이 **\***을 이용해서 해당 주소의 실제 값을 변경하면 **main**의 스택에 있는 변수 a의 값이 변경됨. (애도 잘 모르겠다)

# Arrays

## 1. Arrays

- ex)
  - `int states[50];`
  - `int days[months]={31,30,31,30};`
  - `#define SIZE 4`
  - `int no_data[SIZE]; // uninitialized array`
- wrong example)
  - `int state 1[-4]; // size must be bigger than 0`
  - `int state 2[0]; // size must be bigger than 0`
  - `float state 3[1.2]; // size must be a integer`

## 2. Assigning Array values

- subnote  
 Can't assign one array to another as a unit  
 Can't use the list-in-braces form except when initializing

## 3. Multi Dimensional Arrays(an array of arrays)

- Type1---Two dimensional array  
 ex) `float rain [5][12]; // array of 5 arrays of 12 floats`
- Type 2---Three dimensional array  
 ex) `int box[10][20][30];`  
 To process this one, we need three nested loops.  
 This is a sequential procedure. As so on, four nested loops are needed for a four-dimensional array.

## Pointers (in order to deal with arrays more efficiently)

- Array name is the address of the first element of the array  
 ex) `int arr[10];`  
`arr==&arr[0];`  
 (This is the reason why we don't use the '&' operator for character strings when we call `scanf()` function.)

- Adding the \* operator )) let the value stored in the pointed-to object
- Adding 1)) make the value increase by the size, of the pointed-to type
- notation  
**`arr[n] == *(arr + n)`**---- refer to same value  
Imply: "Go to memory location arr , move over n units, and retrieve the value there."  
ex) `days[index] == *(days+index)`