

새싹교실

새싹교실

Week4

- 복습

- C Language (함수&포인터 Intro)

고주형

2019/05/07

Review

Review

Review 1

- 지역 변수
- 전역 변수

Review 2

- while
- for

Review 3

- 전처리기

 - #include <header파일>

 - #define 상수명(주로 대문자) 값

 - ...

C Programming

c l o a d u n n n n n n n n n n

오늘 할 것

- Function Intro
- How to use function
- Pointer Intro
- How to use Pointer
- Dereference
- Array Intro

Function Intro

프로그래밍을 계속 하다 보면?

- 반복되는 코드들 ...
- 계속 하는 실수들 ...
- 다른 사람의 코드를 이해...

Function Intro

함수를 쓰면?

- 한 번만 작성해 놓으면 나중에 다시 쓸 수 있다!

Ex) printf(), scanf(), 괄호가 들어가는 것들w

- 조금 더 직관적!

How to use Function

```
반환값의_자료형 함수이름(){  
    //명령어들 작성  
}
```

*용어

함수를 사용한다 == 함수를 호출(call)한다

How to use Function

반환값의_자료형? == 어떤 값을 return하냐?

void

int

float

double

char

...

How to use Function

Ex)

```
void main(){  
    print_hello(); //print_hello함수를 호출  
}  
void print_hello(){  
    printf("hello\n");  
}
```

How to use Function (return)

```
반환값의_자료형 함수이름(){  
    //명령어들 작성  
    return 반환값;  
}
```

How to use Function (return)

Ex)

```
void main(){  
    //add함수를 호출  
    printf("%d",add(1,2));  
}  
int add(int a, int b){  
    return a + b;  
}
```

연습

1. Int형인 100을 반환하는 함수를 만들어보자.
2. 1,2,3,4,5의 평균을 구하는 함수를 만들어보자.

Pointer Intro

변수는 어디에 생길까?

- 메모리의 어딘가에 생긴다.

메모리 어디에?

- n번째에

Pointer Intro

메모리 주소 연산자 &로 주소를 출력해보자.

```
int a = 10;
```

```
printf("%p",&a); //a의 주소를 출력한다.
```

Pointer Intro

메모리 주소 연산자 &로 주소를 출력해보자.

```
int a = 10;
```

```
//서식지정자 %p (pointer의 약자)
```

```
printf("%p",&a); //a의 주소를 출력한다.
```

```
//16진수로 출력해도 가능
```

```
printf("%x",&a); //a의 주소를 출력한다.
```

Pointer Intro

메모리 주소를 저장하는 변수 == 포인터 변수

포인터 변수는 자료형 뒤에 *을 사용해서 선언한다.
자료형* 포인터이름;

- * 포인터 변수를 줄여서 포인터라고도 부른다.
- *은 Asterisk라고 부른다.

How to use Pointer 참고

다양한 포인터 선언 방법

다 똑같은 의미! 마음에 드는 방법 쓰면 됨.

```
int* aPtr;
```

```
int * aPtr;
```

```
int *aPtr;
```

How to use Pointer 참고

포인터의 크기는 32비트인지 64비트인지에 따라 달라진다!

다른 자료형의 포인터 변수라도 크기는 모두 동일하다!

//확인해보자!!

```
printf("%d %d %d",sizeof(int*),sizeof(char*),sizeof(double*));
```

How to use Pointer

아무 변수의 주소를 포인터에 저장해보자!

```
//pointer to int
```

```
int a = 10;
```

```
int* aPtr = &a;
```

```
//pointer to float
```

```
float b;
```

```
float* bPtr = &b;
```

How to use Pointer

포인터가 아무 주소도 안 가리키고 싶다면?

널 포인터를 할당하면 된다.

```
int* Ptr1 = NULL;
```

```
float* Ptr2 = NULL;
```

Dereference(*)

역참조 연산자(*): 메모리 주소로 이동해서 값 가져오기!
*포인터

```
int a = 10;  
int* aPtr = &a;  
// 역참조!!  
*a = 11;  
printf("%d",a);
```


Dereference(*)

```
int a = 10;  
// pointer to int  
int* aPtr = &a;  
// pointer to int  
*a = 11;  
printf("%d",a);
```

* 정리

*의 사용법 3가지

- 곱하기 연산자:
- 포인터 변수:
- 역참조 연산자:

숫자 * 숫자
자료형 *
*포인터

pointer 정리

1. 포인터는 메모리의 주소를 저장하기 위해 사용한다.
2. 포인터 앞에 *을 사용하여 포인터가 가리키는 값에 접근할 수 있다.(역참조)

포인터 실습

main에 있는 두 변수의 값을 swap해주는 함수를 만들어 보자.

Array Intro

학교 전체 성적의 평균을 구하고 싶다.

```
int p1,p2,p3,p4, ..... , p100,  
..... , p200, ...;
```

ㅂ ㄷ ㅂ ㄷ

이럴 때 배열을 쓴다.

Array Intro

자료형 배열이름[크기]

- * 크기에는 무조건 상수가 들어가야한다.

Or

자료형 배열이름[크기] = {값1,값2,값3}

Or

자료형 배열이름[] = {값1,값2,값3}

- * 이 경우 알아서 크기를 채워줌.

Next Week

더블 포인터
포인터와 배열
동적 메모리 할당
Const Pointer

