

C 요약본

jh

안내 말씀...

해당 파일은 C의 기능 파악용으로 만들어진 파일입니다. 기본기 습득을 하신 후 점검 용으로 보시면 됩니다.

1. 변수와 자료형

변수 = 하나의 데이터를 저장하는 자료구조

자료형 변수명 = 값; (변수의 선언 및 초기화)

캐스팅(변수 형 변환) = (자료형)변수명

자주 쓰이는 자료형			
자료형 이름	표현 데이터	크기(Byte)	초기값
int / long long	정수형 숫자	4 / 8	0
char	정수형 문자	1	/0
float / double	실수형 숫자	4 / 8	0.0
bool	논리형	-	false

연산자

연산자 종류	기호	항 개수	기능 설명
산술	+ , - , * , / , %	2	사칙연산, 나머지
비교	==, !=, >, <, >=, <=	2	같다, 다르다, 크기 비교
논리	&&(AND) , (OR) ,!(NOT)	2,(!)1	AND, OR ,NOT
관계	=	2	사칙연산과 붙여서 복합 관계연산자로 활용 (+=, -=, *=, /=, %=)
증감	++, -	1	변수 앞에 붙으면 해당 변수에 1 더한 후 연산 변수 뒤에 붙으면 연산 후 해당 변수의 값 1 증가

비트 연산

연산 힌트: 2의 제곱 수로 숫자를 쪼개서 형태 표현

$A \& B$ = 교 집합 (AND 연산)

$A | B$ = 합집합 (OR 연산)

$A \wedge B$ = 여집합 (합집합 - 교집합)

\ll, \gg 화살표 방향으로 비트 이동 (시프트 연산)

조건문 if

if(조건식)

```
{  
    조건식 결과가 참일 때 실행할 명령문;  
}
```

else if(조건식 2)

```
{  
    조건식2의 결과가 참일 때 실행할 명령문;  
}
```

else

```
{  
    조건식 결과가 거짓일 때 실행할 명령문;  
}
```

switch문

switch(변수명)

{

case 변수에 해당하는 값:

break; //작성 안할 시 case가 다음 case 찾으러 감

default :

case 에 해당하지 않는 기본 값 설정

}

while문

while(조건식)

```
{  
    조건식이 만족하는 동안 실행할 명령문;  
}
```


do while

조건식의 초기식이 초기값을 사용하는 경우 사용해주면 좋다.

do

{

조건식의 결과와 상관없이 1번은 실행, 그 이후 조건식의 결과에 따라 처리

}while(조건식);

while(1)

```
while(1)
```

```
{
```

조건식 만족 시 반복할 명령문, 조건식에 **1**을 적으면 무한 루프

```
}
```

for 문

```
for(초기식;조건식;증감식)
```

```
{
```

```
    실행할 명령문;
```

```
}
```

초기식 실행 -> 조건식 확인 -> 조건 만족 시 명령문 실행 -> 증감식 처리

함수

하나의 기능을 수행하기 위해 만든 명령문의 집합체

자료형 함수명(매개변수)

{

 자료형에 맞는 return 값;

}

매개변수(parameter): 함수 실행 시 넣어줄 데이터에 대한 표현

인수(argument): 함수 실행을 위한 매개변수에 대한 값 그 자체

구조체

변수를 묶어서 사용할 수 있는 사용자 정의 타입

struct 구조체명

{

변수 선언;(멤버 변수)

};

메인에서 사용 시 **struct** 구조체명 구조체변수명; <- 구조체 선언

struct 구조체명 구조체변수명 = {멤버 변수 순서대로 값 대입};

[구조체로 만든 값 사용]

구조체명.멤버변수명;

배열

같은 타입의 데이터를 묶어서 표현하는 자료구조(구조체, 포인터 배열 가능)

자료형 배열명[배열의 길이];

인덱스 = 배열의 값에 대한 순서를 표현한 숫자형 데이터(0부터 셈)

요소 = 배열 안에 있는 값

문자열

`char 변수명[길이];`

영문자 1개당 길이 1, 한글일 경우 2

`#include <string.h>`

`strcmp(문자열1, 문자열2)` 두 문자가 같으면 0 리턴

`strcpy(문자열1, 문자열2);` 문자열 내용 변경(복사)

포인터 1

타입* 포인터변수명; //포인터 선언

타입* 포인터변수명 = NULL; //NULL 포인터 (가리키는 값이 없다는 뜻)

타입* 포인터변수명 = &변수명;

*(Asterisk): 변수 앞이나 타입 뒤에 붙으면 참조 연산자의 뜻을 가짐.

주소값을 값으로 반환받는 기능 (포인터)

&(ampersand): 변수 앞에 붙어서 해당 변수의 주소값 표현

포인터 2

배열에서 포인터 연산 표현

`int iArray[5]`라 할 때,

`iArray[0] = iArray + 0;`

`int iArray2[0][2];`라 할 때,

배열 포인터 `(*ptr)[2];`

포인터 3

void 포인터

(자료형*)로 캐스팅해서 원하는 타입을 다 가리킬수 있는 만능 포인터

함수 포인터

자료형 (*함수포인터명) (매개변수) = NULL; //함수 포인터 선언

포인터 3

구조체에서 포인터 사용할 경우

```
struct 구조체명* 구조체변수명 = &구조체변수명;
```

구조체 포인터로 값 사용 시

```
구조체변수명 -> 멤버변수명;
```

```
(구조체변수명*).멤버변수명;
```

동적 할당

```
#include <stdlib.h>
```

1. 변수 할당

```
자료형* 변수명 = (자료형*)malloc(sizeof(자료형 크기));
```

2. 배열 할당

```
자료형* 변수명 = (자료형*)malloc(sizeof(자료형 크기)*배열의 길이);
```

```
free(변수명); <- 데이터 반납
```