

2022 C언어 101

```
#include <stdio.h> int main(){ return (); };
```

# Input		
getchar()	문자 하나 입력	char c; c = getchar();
gets()	문자열 입력	char c[10]; gets(c);
fgets()	gets 오퍼플로우 검사x fgets 자동 개행문자\n	
scanf()	서식 입력, 키보드 scanf는 띄어쓰기x fgets는 줄바꿈 전까지	scanf("%lf %c", &score, &grd);

# Output		
putchar()	문자 하나 출력	putchar('a')
puts()	문자열 출력	puts('hello')
fputs()		
printf()	서식 출력, 모니터	printf("%.1lf점 %c입니다.", 4.2, 'A');

# 배열		
~[2]={0, 0}	type 식별자[length], 초기화0	int ar1[3]={1, 2, 3} [1, 2, 3] int ar2[3]={1, 2} [1, 2, 0] int ar3[3]={0} [0, 0, 0]
~[]={1, 2}	초기화 개수에 따라 배열길이	int ar[]={1, 2, 3} [1, 2, 3] (3개)
배열, 첨자	주소 상수: ar, 변수명처럼 데이터를 할당할 수 없는 상수 포인터 연산: ar+1, 주소상수에 연산, 첫 번째 byte, int면 주소+4	첨자Index: ar[1] 0~, 음수x
다차원 배열	2차원배열 [2][3]={} n차원배열 [...][]={}	int ar[2][3] = {1, 2, 3, 4, 5, 6}; [1, 2, 3, 4, 5, 6] int ar[2][3] = {{1, 2}, {4, 5}}; [1, 2, 0, 4, 5, 0] int ar[2][3] = { {1, 2}, {4, 5, 6}, } [1, 2, 0, 4, 5, 6]

# 포인터 연산자 Pointer Operator // 데이터 참조: 식별자가 아닌 주소값으로 접근 // &, *					
포인터 변수 pointer variable	주소값 저장(크기값 동일)	<참조 데이터타입> *<식별자>; <포인터 변수>=&<데이터 변수>; *<포인터 변수>=<값>;	int *p; p = &data; *p= 20;	p	data변지 (&)
				*p	20 (찾아감)
				*p+2	22
배열, 포인터	배열이름은 주소상수 (주소값 재할당x)	<참조 데이터타입> *<식별자>; <포인터변수>=<배열이름>; int ar[3]={3, 6, 9};	int *pr; pr=ar;	pr[2]	= ar[2] = 9
				*pr+2	3+2 = 5
				*(pr+2)	ar[0+2] = 9

# C언어 표준 라이브러리 // #include <~.h>				
<stdio.h>	데이터 입출력 STanDard Input/Output	printf()	서식 출력, 모니터	
		scanf()	서식 입력, 키보드	
<math.h>	수학 함수	sqrt()	SQuare Root	제곱근
		pow()	POWer	제곱수
		abs()	ABSolute	절대값
<string.h>	문자열 처리	strlen()	LENGth	문자열 length
		strcpy()	CoPY	문자열 copy
		strcmp()	CoMPare	문자열 비교
<stdlib.h>	자료형 변환, 난수 STanDard LiBrary	atoi()	Alphabet TO Int	문자열 => 정수 타입
		atof()	Alphabet TO Float	문자열 => 실수 타입
		rand()	RANDom	난수 발생
<time.h>	시간처리	time, clock		

# Return	Call By Value	매개변수 복사(값이 변화해도 원래 값에 영향x)	int sum(a, b){ ~ }
	Call By Reference	매개변수를 카피가 아니라 포인터로 주소 넘김	int sum(a, *b){ ~ }
	값 없이 return; => function 데이터타입 void여야 함 ex) void add_value(int a, int b){ ~~ return; }		

# 구조체 struct	struct student { int no; int score; char grade; };	struct student kim = {111, 98, A}; struct student lee = {222, 89, B}; lee.score=86;	printf("%d %c \n", kim.score, kim.grade); printf("%d %c \n", lee.score, lee.grade);
--------------	--	---	--