

C언어

2019.05.21 화

복습

프로그래밍 언어 / 형(type) / 함수의 구조와 사용
연산자 / 출력 / 입력

프로그래밍 언어

- Low-level
- High-level

프로그래밍 언어

- Low-level
 - 기계어, 어셈블리 언어
 - 기계는 이해할 수 있지만 사람은 이해하기 어려운 언어
- High-level
 - C, Python, Java 등 ..
 - 기계는 이해할 수 없지만 사람은 이해하기 쉬운 언어

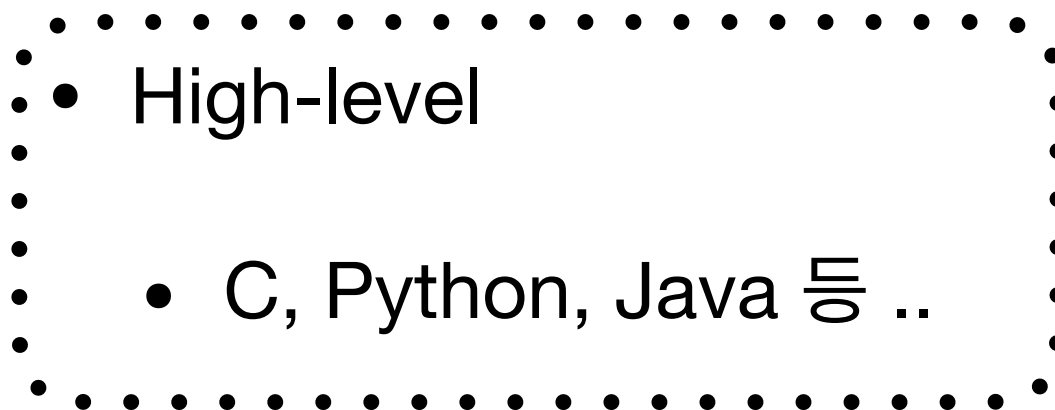
프로그래밍 언어

- Low-level

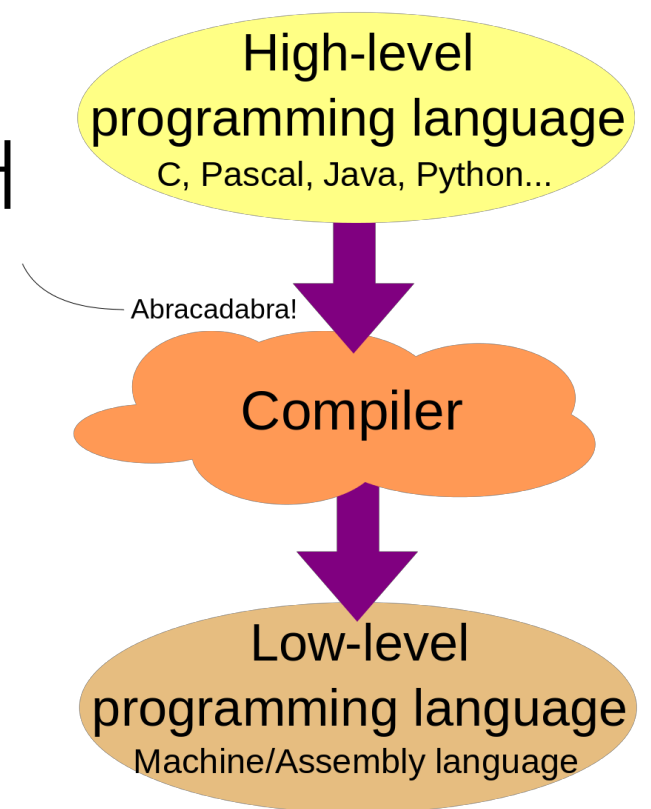
- 기계어, 어셈블리 언어

- 기계는 이해할 수 있지만 사람은 이해하기 어려운 언어

- 컴파일러



- 기계는 이해할 수 없지만 사람은 이해하기 쉬운 언어



형(type)

- 정수형

- $\text{char} < \text{short} < \text{int} < \text{long} < \text{long long}$
1 2 4 4 8

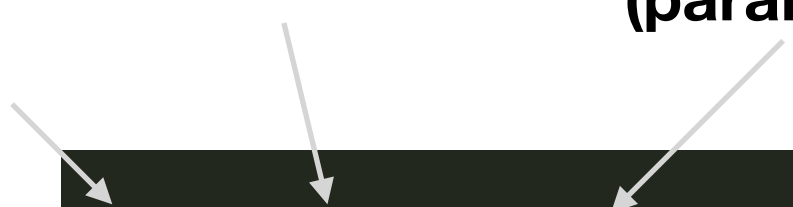
- 실수형

- $\text{float} < \text{double} < \text{long double}$
4 8 8 +

함수의 구조와 사용

반환 형 함수의 이름 인자
(parameter)

```
int main (void)
{
    printf("Hello world! \n");
    return 0;
}
```

The diagram consists of three labels with arrows pointing to specific parts of the C code. The label '반환 형' (Return type) has an arrow pointing to the 'int' keyword. The label '함수의 이름' (Function name) has an arrow pointing to the 'main' identifier. The label '인자 (parameter)' (Argument (parameter)) has an arrow pointing to the 'void' keyword in the parentheses.

함수의 구조와 사용

```
int main (void)
{
    printf("Hello world! \n");
    return 0;
}
```

함수의 몸체

함수의 구조

함수의 구조와 사용

```
int main (void)
{
    printf("Hello world! \n");
    return 0;
}
```

반환 값

함수의 구조와 사용



함수의 구조와 사용

```
int add (int a1, int a2)
{
    int sum = a1 + a2;
    return sum;
}
```

```
add(3, 5);
```

호출 및 인자 전달

연산자

산술 연산자

+	-	*	/	%	=
덧셈	뺄셈	곱셈	나눗셈	Modulo	대입

논리 연산자

&&		!
AND	OR	부정

출력, 입력

- 출력
 - printf
- 입력
 - scanf

출력

```
#include <stdio.h>

int main(void)
{
    int num1 = 10;
    int num2 = 12;
    int result1, result2, result3;

    result1 = (num1==10 && num2==12);
    result2 = (num1<12 || num2>12);
    result3 = (!num1);

    printf("result1: %d \n", result1);
    printf("result2: %d \n", result2);
    printf("result3: %d \n", result3);
}
```

입력

```
#include <stdio.h>

int main(void)
{
    int result;
    int num1, num2, num3;

    printf("3개의 정수 입력: ");
    scanf("%d %d %d", &num1, &num2, &num3);

    result = num1+num2+num3;
    printf("sum = %d", num1+num2+num3);
}
```

실습

1. 두 숫자를 입력받아 결과를 출력해주는 프로그램을 작성해보자
2. 1번에서 작성한 코드를 함수로 구현해보자

본 수업

서식 문자

서식 문자	출력 대상 (자료형, type)	출력 형태
%d	char, short, int	부호 있는 10진수 정수
%ld	long	
%lld	Long long	
%u		
%o	Unsigned int	부호 없는 10진수 정수
%x, %X		
%f	float, double	10진수 방식의 부동소수점 실수
%Lf	Long double	
%c	char, short, int	값에 대응하는 문자
%s	char*	문자열

예시

```
#include <stdio.h>

int main(void)
{
    long long num = 100;
    printf("%lld", num);

    char a = 'a';
    printf("%c", a);
}
```

문자의 표현

- 숫자만 읽을 수 있는 컴퓨터에게 문자를 읽게 하는 방법은?

문자의 표현

- 숫자만 읽을 수 있는 컴퓨터에게 문자를 읽게 하는 방법은?
- 문자를 숫자로 표현하게 한다

ASCII code

ASCII code?

- **ASCII** American Standard Code for Information Interchange, 미국 정보보안 표준 부호 **code**

아스키 코드	아스키 코드 값
A	65
B	66
C	67
~	126

제어 문자			공백 문자			구두점			숫자			알파벳		
10진	16진	문자	10진	16진	문자	10진	16진	문자	10진	16진	문자	10진	16진	문자
0	0x00	NUL	32	0x20	SP	64	0x40	@	96	0x60				
1	0x01	SOH	33	0x21	!	65	0x41	A	97	0x61	a			
2	0x02	STX	34	0x22	"	66	0x42	B	98	0x62	b			
3	0x03	ETX	35	0x23	#	67	0x43	C	99	0x63	c			
4	0x04	EOT	36	0x24	\$	68	0x44	D	100	0x64	d			
5	0x05	ENQ	37	0x25	%	69	0x45	E	101	0x65	e			
6	0x06	ACK	38	0x26	&	70	0x46	F	102	0x66	f			
7	0x07	BEL	39	0x27	'	71	0x47	G	103	0x67	g			
8	0x08	BS	40	0x28	(72	0x48	H	104	0x68	h			
9	0x09	HT	41	0x29)	73	0x49	I	105	0x69	i			
10	0x0A	LF	42	0x2A	*	74	0x4A	J	106	0x6A	j			
11	0x0B	VT	43	0x2B	+	75	0x4B	K	107	0x6B	k			
12	0x0C	FF	44	0x2C	,	76	0x4C	L	108	0x6C	l			
13	0x0D	CR	45	0x2D	-	77	0x4D	M	109	0x6D	m			
14	0x0E	SO	46	0x2E	.	78	0x4E	N	110	0x6E	n			
15	0x0F	SI	47	0x2F	/	79	0x4F	O	111	0x6F	o			
16	0x10	DLE	48	0x30	0	80	0x50	P	112	0x70	p			
17	0x11	DC1	49	0x31	1	81	0x51	Q	113	0x71	q			
18	0x12	DC2	50	0x32	2	82	0x52	R	114	0x72	r			
19	0x13	DC3	51	0x33	3	83	0x53	S	115	0x73	s			
20	0x14	DC4	52	0x34	4	84	0x54	T	116	0x74	t			
21	0x15	NAK	53	0x35	5	85	0x55	U	117	0x75	u			
22	0x16	SYN	54	0x36	6	86	0x56	V	118	0x76	v			
23	0x17	ETB	55	0x37	7	87	0x57	W	119	0x77	w			

실습 1

```
#include <stdio.h>

int main(void)
{
    char ch1 = 'A', ch2 = 65;
    int ch3 = 'Z', ch4 = 90;

    printf("%c %d \n", ch1, ch1);
    printf("%c %d \n", ch2, ch2);
    printf("%c %d \n", ch3, ch3);
    printf("%c %d \n", ch4, ch4);
}
```

실습 2

- 프로그램 사용자로부터 알파벳 문자 하나를 입력 받는다.
- 그리고 이에 해당하는 아스키 코드 값을 출력하는 프로그램을 작성해보자.
- 예를 들어서 프로그램 사용자가 문자 A를 입력하면 정수 65를 출력해야 한다.

변수

- 변수 = 변하는 수
- 상황에 따라 크기가 변하는 수
- 상황에 따라 값이 변하는 것

변수가 선언이 되면?

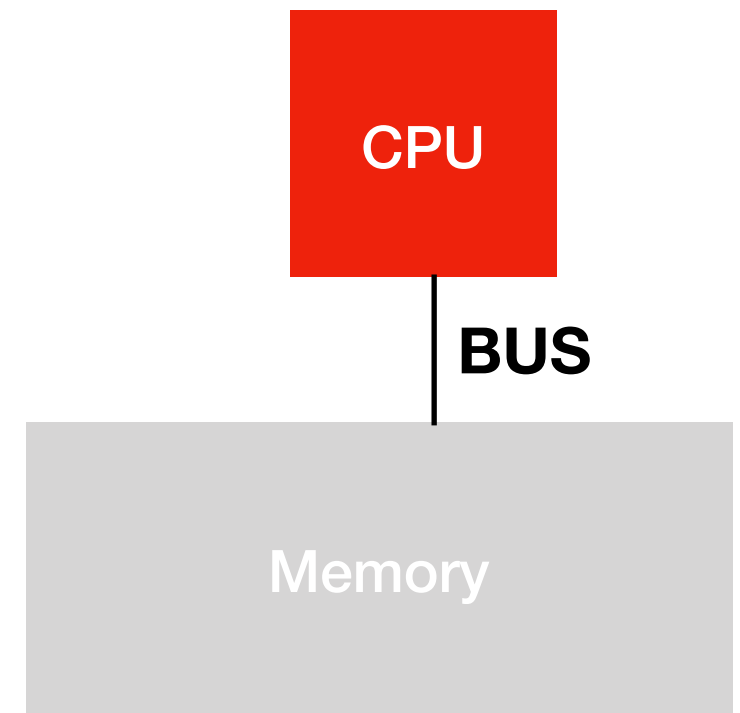
1. 사용자가 변수를 선언

2. 실행

1. CPU가 변수를 선언

1. 메모리에게 **변수를 할당할 공간을 줘! 라고 함**

2. 할당됨



상수

- 상수
 - 변경이 불가능한 데이터
- Literal_{그대로인} 상수 (리터럴 상수)

```
int num = 30 + 40;
```

 - 이름이 **없는** 상수
- Const_{constant, 변함없는} 상수 (콘스트 상수)

```
const int num = 70;
```

 - 이름이 **있는** 상수

반복문 - for 문의 구조

```
for (초기식; 조건식; 증감식) {  
    내용  
}
```

```
for (int num=0; num<3; num++) {  
    printf("Hello world!");  
}
```

예제

```
#include <stdio.h>

int main(void) {

    int total = 0;
    int i, num;

    printf("0 ~ num 까지의 덧셈을 할 때, num은? ");
    scanf("%d", &num);

    for(i = 0; i < num+1; i++)
        total+=i;

    printf("0 ~ %d 까지의 덧셈 결과: %d\n", num, total);

}
```

실습

1. 프로그램 사용자로부터 두 개의 정수를 입력받는다. 그리고 두 정수 사이의 수를 모두 더한 결과를 출력하는 프로그램을 작성한다.
예를 들어, 3과 5가 입력되면, 3+4+5의 12가 답이다.
2. 계승(factorial)을 계산하는 프로그램을 작성해보자.
 - $n! = 1 \times 2 \times 3 \times \dots \times n$
 - 예를 들어 $3! = 3 \times 2 \times 1 = 6$
3. 1번을 함수로 분리해서 구현해보자.