C 언어

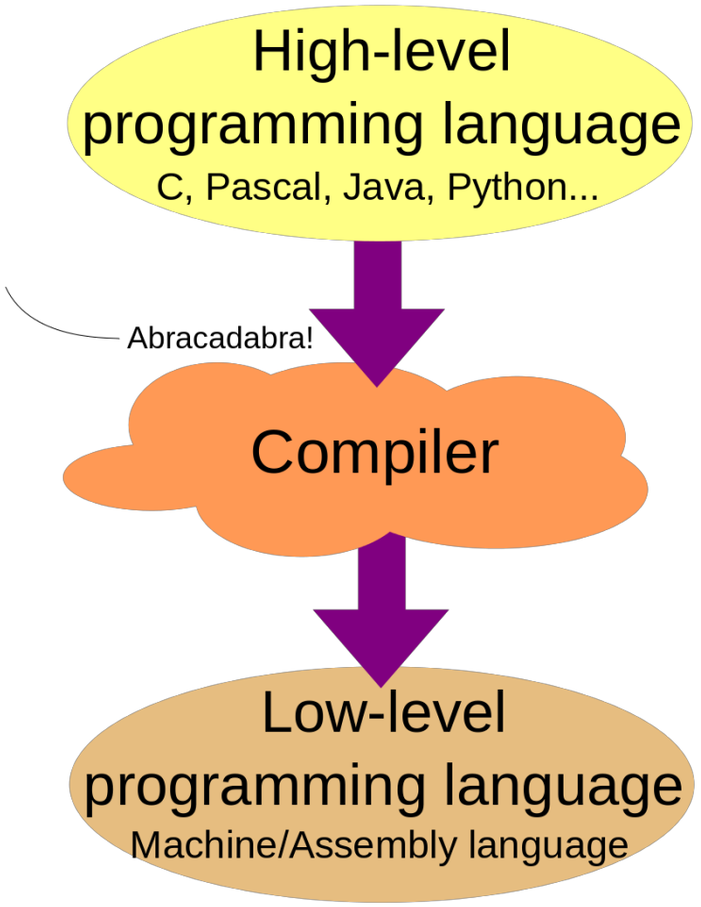
2019.05.17 금

프로그래밍 언어?

- 기계어
- 프로그래밍 언어
- 컴파일러

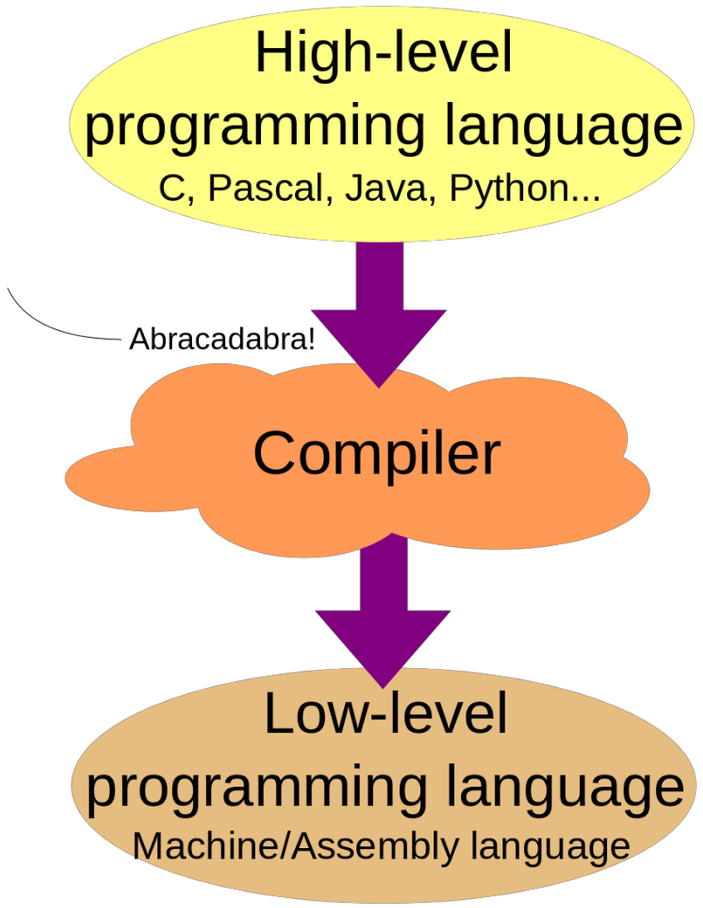
프로그래밍 언어?

- 기계어
  - 0과 1로 이루어진 언어
  - 기계가 이해할 수 있음
- 프로그래밍 언어
  - 비교적 사람이 이해할 수 있는 언어
  - 기계가 이해할 수 없음
- 컴파일러
  - 프로그래밍 언어 -> 기계어 로 바꾸어, 기계가 이해할 수 있도록 함
  - 이러한 작업 = ‘컴파일’



프로그래밍 언어?

- 기계어 - Low-level language
  - 0과 1로 이루어진 언어
  - 기계가 이해할 수 있음
- **프로그래밍 언어 - High-level language**
  - **비교적 사람이 이해할 수 있는 언어**
  - **기계가 이해할 수 없음**
- 컴파일러
  - 프로그래밍 언어 -> 기계어 로 바꾸어, 기계가 이해할 수 있도록 함
  - 이러한 작업 = ‘컴파일’



C 언어의 역사

- 1971년
- UNIX라는 운영체제의 개발을 위함
    - 기존의 언어는 Assembly 언어로 만들어져있었음
    - 하드웨어의 의존도가 높음
- Dennis Ritchie, Ken Thompson
- ALGOL 60(1960) - CPL(1963) - BCPL(1969) - B(1970)

변수

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

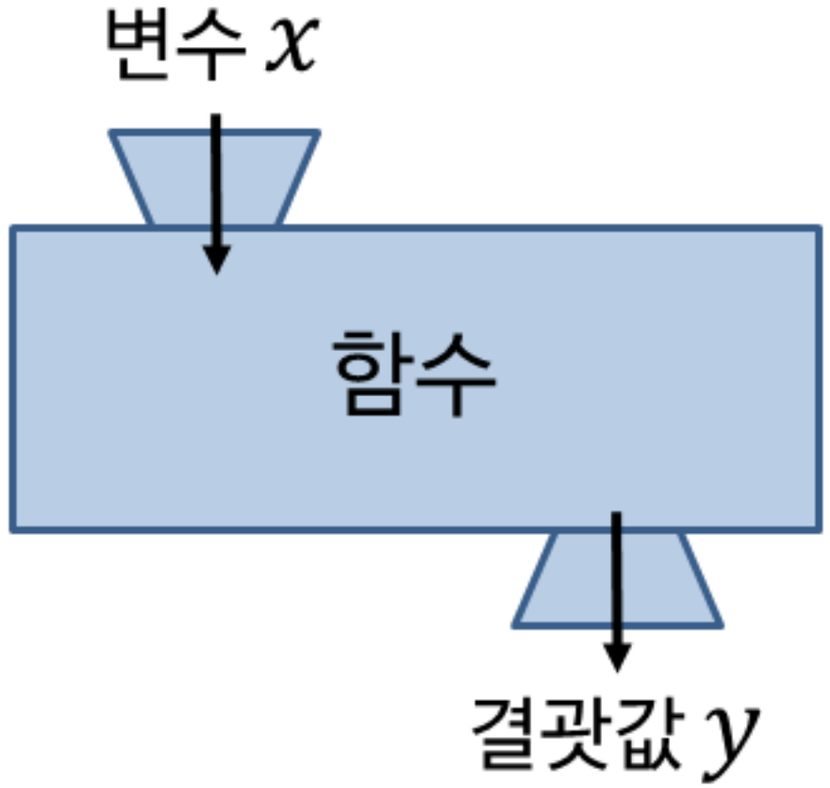
변수의 형(type)

정수형	char	1바이트	-128 이상 +127 이하	
	short	2바이트	-32,768 이상 +32,767 이하	
	int	4바이트	-2,147,483,648 이상 +2,147,483,647 이하	
	long	4바이트	-2,147,483,648 이상 +2,147,483,647 이하	
	long long	8바이트	-9,223,372,036,854,775,808 이상	
			+9,223,372,036,854,775,807 이하	
실수형	float	4바이트	±3.4 * 10^-37 이상	±3.4 * 10^-38 이하
	double	8바이트	±1.7 * 10^-307 이상	±1.7 * 10^-308 이하
	long double	8바이트 이상	double 이상의 표현	

(1 byte = 8 bit = 2^8)

함수

- 함수?
  - x를 넣었을 때, y로 변환 시켜주는 것



함수의 구성

반환 형

함수의 이름

인자
(parameter)

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

함수의 구성

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

함수의 몸체

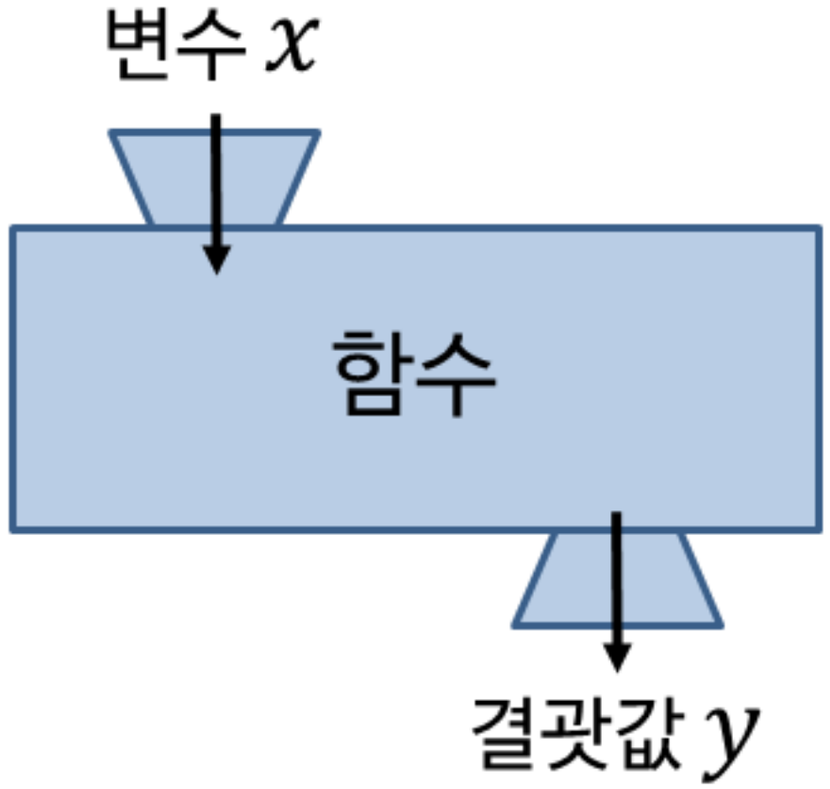
함수의 구성

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

반환 값

함수

- C에서 함수를 사용할 때
  - 함수의 정의
  - 함수의 호출
  - 인자의 전달



함수의 정의

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

반환 형 (반환 값) / 이름 / 인자 값 / 함수 몸체

함수의 호출, 인자의 전달

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

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

연산자 1 - 계산

- =
  - num = 20;
- +
  - num = a + b;
- -
  - num = a - b;
- *
  - num = a * b;
- /
  - num = a / b;
- %
  - num = a % b;
  - 1 = 10 % 3;

실습

```c
#include <stdio.h>

int main(void)
{
    int num1 = 9, num2 = 2;
    printf("%d + %d = %d", num1, num2, num1+num2);
    printf("%d - %d = %d", num1, num2, num1-num2);
    printf("%d X %d = %d", num1, num2, num1*num2);
    printf("%d / %d = %d", num1, num2, num1/num2);
}
```

연산자 짧게 쓰기 1

```c
#include <stdio.h>

int main(void)
{
    int num1 = 2, num2 = 4, num3 = 6;
    num1 += 3;
    num2 *= 4;
    num3 %= 5;
    printf("%d %d %d", num1, num2, num3);
}
```

연산자 짧게 쓰기 2

```c
#include <stdio.h>

int main(void)
{
    int num1 = 2, num2 = 4, num3 = 6;
    num1++;
    num2++;
    num3++;
    printf("%d %d %d", num1, num2, num3);
}
```

연산자 2 - 논리 연산자

- &&
  - A and B = ?
- ||
  - A or B = ?
- !
  - Not
  - !A = ?

실습

```c
#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);
}
```

사용자로부터 입력 받기

- 출력
  - printf
- 입력
  - scanf

scanf

```c
int a;

scanf("%d", &a);
```

10진수 정수 형태로 입력 받아서　　변수 a에 저장하기

실습

```c
#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);
}
```

연습문제

1. 프로그램 사용자로부터 두 개의 정수를 입력 받아서 두 수의 뺄셈과 곱셈의 결과를 출력하는 프로그램을 작성해보자

2. 하나의 정수를 입력 받아서, 그 수의 제곱의 결과를 출력하는 프로그램을 작성해보자. 예를 들어서 5가 입력되면 25가 출력되어야 한다