

3.1 : C 프로그램 구조와 프로그램 실행

실습예제 3-1: 01comments.c

```
#include <stdio.h>

// 운영체계가 호출하는 함수, void로 매개변수 없음을 표시
int main(void)
{
    puts("3장 첫 c 프로그램!\n");

    printf("키워드: int void return 등\n");
    printf("식별자: main puts 등\n");

    return 0;
}
```

【생각해 보기】 출력은 어떤 형태일까요?

3장 첫 c 프로그램! 키워드: int void return 등 식별자: main puts 등	3장 첫 c 프로그램! 키워드: int void return 등 식별자: main puts 등
--	--

☞ 제어문자 출력

제어문자	의미	제어문자	의미
\n	줄 바꿈	\a	벨 소리
\t	탭 위치로 이동	\b	왼쪽으로 한 칸 이동
\r	맨 앞으로 이동		

【도전과제 - 1】 다음 프로그램 출력은?

#include <stdio.h>	예상 출력
<pre>int main(void) { printf("Goot\bd\tchance\n"); printf("Cow\rW\n"); return 0; }</pre>	

3.2 : 자료형(Data Type)과 변수(Variable) 선언

□ 변수(Variable)

- 프로그램은 문제 해결 **시나리오**이며, 프로그램은 문제 해결을 위해 **데이터**를 처리

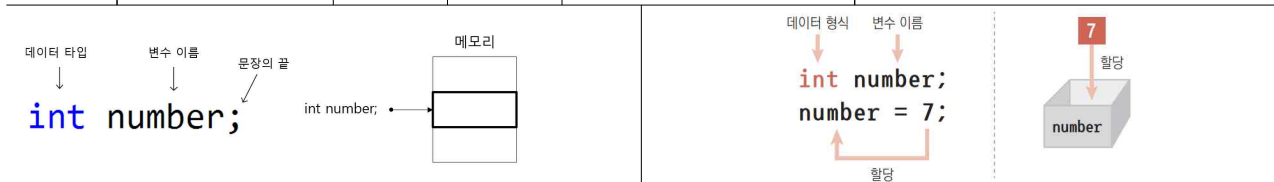
구분	정의	값 변경	메모리 역할	예시코드
변수	실행 중 값이 바뀔 수 있는 데이터 공간	가능	값을 저장할 공간 제공	int age = 20; age = 25;
	age = 25			20 -> 25 (내용을 언제든지 바꿀 수 있음)

□ 변수(Variable)는 왜 필요할까?

- 변수 : 프로그램에서 일시적으로 데이터를 저장하는 공간
- 데이터가 입력되면 어딘가에 저장해야 다음에 사용할 수 있음
- 변수의 저장공간은 **메인 메모리**에 만들어 짐
- 변수로 메모리의 주소를 사용할 수 있지만 기억하기 불편 - 변수 이름으로 사용
(예: 메모리 100번지에 20을 대입하라 / 변수 sum에 20을 대입하라)

□ 변수(Variable) 선언 및 활용

자료	설명	자료형	크기 (Byte)	선언 예시	사용 예시/출력 예시
정수	정수 저장 (소수점 없음)	int	4	int age = 20;	printf("%d", age); → 20 출력
실수	실수 저장 (소수점 포함)	double	8	double pi = 3.14;	printf("%.2lf", pi); → 3.14 출력
문자	문자 1개 저장	char	1	char grade = 'A';	printf("%c", grade); → A 출력
문자열	여러 문자 저장	char[]	가변	char name[] = "Tom";	printf("%s", name); → Tom 출력

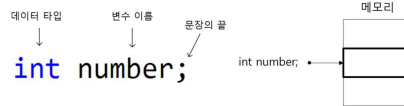


- 변수는 중괄호의 블록({})에 선언하여 선언한 위치부터 블록 끝까지 사용 가능
- 변수의 자료형이 같으면 동시에 여러 개 선언 가능 (예: int b, c)
- 대입 연산자는 왼쪽의 변수에 오른쪽의 값을 저장함 (예: a = 10;)
- 변수는 대입 연산자 왼쪽에서는 **저장공간**, 오른쪽에서는 **값**이 됨 (예: a = 10; b = a;)

※ l-value(left value) : 저장공간으로 사용하는 변수, r-value(right value) : 값으로 사용하는 변수

□ 변수(Variable) 이름 짓기

- 메모리 시작 주소의 이름 (앞으로 number라고 불러 주세요)



- 변수 이름 만드는 규칙

■ 알파벳 문자와 숫자, 밑줄 문자 _로 구성	sum	o
■ 첫 번째 문자는 반드시 알파벳 또는 밑줄 문자 _	_count	o
■ 대문자와 소문자를 구별	dongyang3	o
■ C 언어의 키워드와 똑같은 이름은 허용되지 않음	dong_yang	o
	2nd_try	x
	Dongyang#	x
	int	x

- 좋은 변수 이름 만들기 : 변수의 역할을 잘 설명하는 이름

- 밑줄 방식 : bank_account
- 단어의 첫 글자를 대문자 : BankAccount

□ 변수(Variable) 타입이 왜 필요할까?

변수 선언

자료형(타입, Type) 변수 이름

`int a;`

메모리 시작 주소

메모리

① 메모리에 int 타입 크기의 공간을 할당
② 그 공간의 메모리 시작 주소를 a로 부름

변수 값 할당

`a = 10;`

메모리 시작 주소

메모리

① a의 메모리 시작 주소부터 int 타입 공간에
② 값 10을 복사

변수 타입이 왜 필요할까?

- 변수 타입은 **메모리 크기**를 나타냄
 - 변수 이름으로 메모리 시작 주소를 알아냄
 - 메모리 크기를 알아야 메모리 주소에서 얼마만큼 읽을지 결정할 수 있음
- 컴퓨터는 변수 타입으로 **데이터 해석**
 - 메모리에 2진수 '0100 0001' 값은 int 경우 65로, char인 경우 'A'로 해석

변수의 시작위치 → 주소

메모리

길이 : 변수가 차지하는 메모리의 크기

```

/**
 * 소스 : 02variables.c
 * 버전: V1.0
 **/
#include <stdio.h>

int main(void)
{
    int year = 2022;    //선언과 동시에 변수 초기화
    int credits;

    credits = 15;       // 선언된 변수에 초기화

    printf("%d년도\n", year);
    printf("이수학점: %d학점\n", credits);

    return 0;
}

```

【생각해 보기】 실행 결과는 ?



```

/**
 * 소스 : 03addsub.c
 * 버전: V1.0
 */
#include <stdio.h>

int main(void)
{
    int data1 = 20, data2 = 13;

    // 대입 연산자의 왼쪽과 오른쪽에서의 변수의 의미 해석
    int diff = data1 - data2;
    int sum = data1 + data2;

    printf("data1: %d, data2: %d\n", data1, data2);
    printf("차 : %d, 합: %d\n", diff, sum);

    return 0;
}

```

【생각해 보기】 실행 결과는 ?

