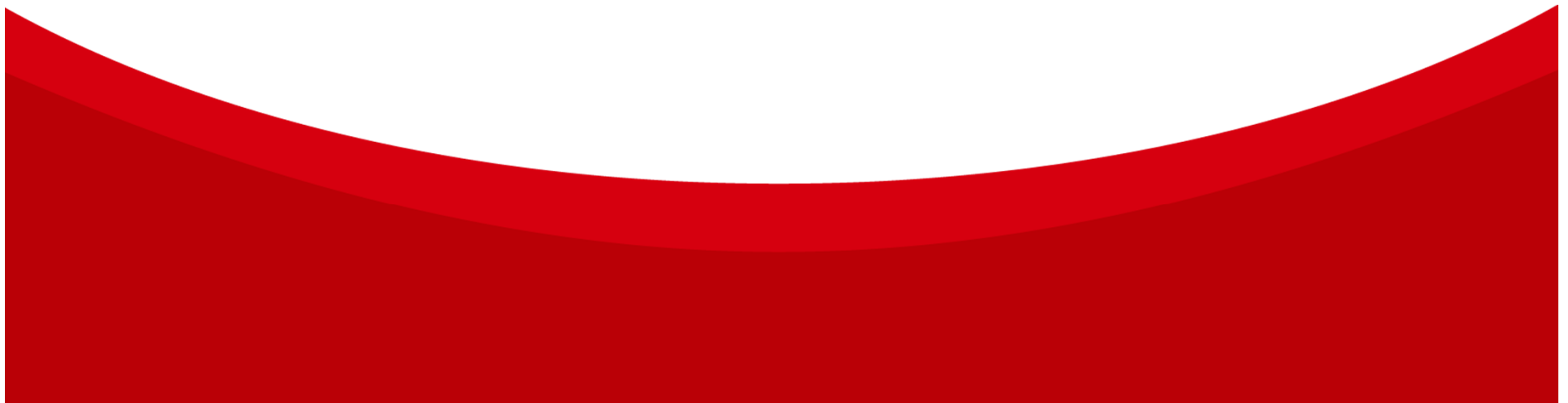


Chapter 2

입력



01

Question

변수를 선언한 후 값을 대입하여 다음과 같이 출력하는 프로그램을 작성하시오.

출력 예

a = 10

b = A

Chapter 2

입력

소스

```
#include <stdio.h>

int main()
{
    int a;
    char b;

    a = 10;
    b = 'A';

    printf("a = %d \n", a);
    printf("b = %c \n", b);

    return 0;
}
```

설명

- `int a;`
정수를 저장하기 위한 변수를 선언하고 변수의 이름을 `a`로 정한다. 변수란 “변경이 가능한 수”를 의미하며 위와 같이 선언을 하면 메모리에 정수를 저장할 수 있는 공간을 할당하고 그 메모리 공간의 이름을 `a`라고 정하게 된다. `int`는 선언하는 변수의 형식이 정수임을 나타낸다.
- `char b;`
문자를 저장하기 위한 변수를 선언하고 변수의 이름을 `b`로 정한다. `char`는 선언하는 변수의 형식이 문자임을 나타내며 1개의 문자만을 저장할 수 있다.
- `a = 10;`
변수에는 해당되는 형식의 값을 넣을 수 있는데 이를 “대입”이라고 하고, “변수 `a`에 10을 대입한다.”라고 말한다. 위에서 선언된 `a`라는 이름의 메모리 공간에 10이 저장된다.

- `b = 'A';`
b라는 이름의 메모리 공간에 문자 'A'가 저장된다. 문자의 경우는 그대로 쓰면 변수와 구분을 할 수 없기 때문에 반드시 ' '로 표시해야 함을 유의하자.
- `printf("a = %d \n", a);`
"%d"의 위치에 변수 a에 저장된 값인 10이 대입되어 출력된다.
- `printf("b = %c \n", b);`
"%c"의 위치에 변수 b에 저장된 값인 'A'가 대입되어 출력된다. "%c"는 문자를 표현 하기 위한 서식이다.
- 변수는 종류에 따라서 크게 숫자형 변수와 문자형 변수로 나뉘어진다. 또, 숫자형 변수는 숫자의 형태에 따라 정수형 변수와, 실수형 변수로 나뉘어진다.

- 지금 꼭 알아두어야 할 자주 사용되는 변수의 종류는 다음과 같다.

종 류	선언문	메모리크기	사용범위 및 특징
정수형 변수	int	4byte	-2147483648 ~ +2147483647
실수형 변수	float	4byte	유효자리 약 6자리 (3.4E-38 ~ 3.4E+38)
	double	8byte	유효자리 약 15자리 (1.7E-308 ~ 1.7E+308)
문자형 변수	char	1byte	한 개의 문자만 저장

Chapter 2

입력

자료형	크기	범위	비고
char signed char	1 바이트, 8비트	-128~127	
unsigned char	1 바이트, 8비트	0~255	
short short int	2 바이트, 16비트	-32,768~32,767	int 생략 가능
unsigned short unsigned short int	2 바이트, 16비트	0~65,535	int 생략 가능
int signed int	4 바이트, 32비트	-2,147,483,648~ 2,147,483,647	
unsigned unsigned int	4 바이트, 32비트	0~4,294,967,295	int 생략 가능
long long int signed long signed long int	4 바이트, 32비트	-2,147,483,648~ 2,147,483,647	int 생략 가능
unsigned long unsigned long int	4 바이트, 32비트	0~4,294,967,295	int 생략 가능
long long long long int signed long long signed long long int	8 바이트, 64비트	-9,223,372,036,854,775,808~ 9,223,372,036,854,775,807	int 생략 가능
unsigned long long unsigned long long int	8 바이트, 64비트	0~18,446,744,073,709,551,615	int 생략 가능

- 변수이름은 프로그램을 작성하는 사람이 마음대로 정할 수 있지만 반드시 아래의 규칙을 지켜야 한다.
 - 1) 변수이름은 알파벳, 숫자, under bar('_')의 조합으로 선언할 수 있다. 공백을 포함한 기타의 특수문자('+', '-', '*', '/', '&'....)들은 사용할 수 없다.
 - 2) 첫 글자에는 숫자를 사용할 수 없다. (알파벳 또는 '_' 문자만 가능하다.)
 - 3) 대문자와 소문자는 다른 문자로 인식되므로 구분하여 사용한다.
 - 4) keyword("int", "void", "if", "return"과 같이 파란색으로 표시되는 단어)는 변수이름으로 사용할 수 없다.

※ 변수의 이름을 정할 때에는 어떤 용도로 사용할 것인지에 따라 알아보기 쉽도록 의미있게 정하는 것이 좋다.

예) 합계를 저장할 변수 : sum, total, hap 등
 평균을 저장할 변수 : avg(average의 약자), pyung 등

Chapter 2

입력

자가진단 1

정수형 변수를 선언하고 -100을 대입하여 출력하는 프로그램을 작성하라.

출력 예

-100

02

Question

두 개의 변수를 선언하여 각각 10과 20을 대입한 후 아래와 같이 숫자를 바꾸어 출력하는 프로그램을 작성하시오.

출력 예 20 10

Chapter 2

입력

소스

```
#include <stdio.h>

int main()
{
    int a;
    int b;

    a = 10;
    b = 20;
    printf("%d %d \n", b, a);

    return 0;
}
```

설명

- `int a; int b;`

정수형 변수 `a`와 `b`를 선언한다.

같은 종류의 변수를 2개 이상 선언할 경우 다음과 같이 선언한다.

```
int a;  
int b;      또는   int a, b, c;  
int c;
```

어떤 방법으로 선언을 해도 된다.

- `a = 10; b = 20;`

정수형 변수 `a`와 `b`에 각각 10과 20을 대입한다.

- `printf("%d %d \n", b, a);`

서식문자를 이용하여 변수 `b`와 `a`의 값을 출력한다. `b`가 먼저 출력됨을 주의하자.

자가진단 2

정수형 변수 2개를 선언하여 `-1`과 `100`을 대입한 후 아래와 같이 출력하는 프로그램을 작성하라.

출력 예

-1

100

03

Question

두 개의 변수를 선언하여 각각 10과 20을 대입하여 그 합을 나타내는 식을 출력한 후, 각각의 값을 30과 40으로 변경하여 다시 식을 출력하는 프로그램을 작성하시오.

출력 예

$$10 + 20 = 30$$

$$30 + 40 = 70$$

Chapter 2

입력

소스

```
#include <stdio.h>

int main()
{
    int a, b;

    a = 10;
    b = 20;
    printf("%d + %d = %d \n", a, b, a+b);

    a = 30;
    b = 40;
    printf("%d + %d = %d \n", a, b, a+b);

    return 0;
}
```

설명

- `printf("%d + %d = %d \n", a, b, a+b);`

첫 번째 “%d” 부분에 a의 값, 두 번째 “%d” 부분에 b의 값 그리고 마지막 “%d” 부분에는 a와 b의 합이 출력된다. 똑같은 문장이 두 개이지만 각각의 문장이 실행될 때 a와 b의 값이 다르므로 출력되는 결과 역시 다르게 된다.

- 변수는 변경이 가능한 수이다. 처음에 변수 a와 b에 10과 20을 대입하면 변경하기 전까지는 그 값을 유지하다가, 새로운 수를 대입하면 그 전에 저장되어 있던 값은 사라지고 새로 대입한 값으로 저장된다.

자가진단 3

두 개의 정수형 변수를 선언하고 값을 대입하여 아래와 같이 출력되는 프로그램을 작성하라.

출력 예

$$55 - 10 = 45$$

$$2008 - 1999 = 9$$

04

Question

화면에 다음과 같이 출력하는 프로그램을 작성하시오.

출력 예 원주 = 5 * 2 * 3.140000 = 31.400000
 넓이 = 5 * 5 * 3.140000 = 78.500000

Chapter 2

입력

소스

```
#include <stdio.h>

int main()
{
    int r = 5;
    double pi = 3.14;

    printf("원주 = %d * 2 * %lf = %lf \n", r, pi, r * 2 * pi);
    printf("넓이 = %d * %d * %lf = %lf \n", r, r, pi, r * r * pi);
    return 0;
}
```

설명

- `int r = 5;`
정수형 변수를 선언하고 초기값으로 5를 대입한다.
다음과 같은 두 문장을 합한 것과 같다.
`int r;`
`r = 5;`
- `double pi = 3.14;`
실수형 변수를 선언하여 초기값으로 3.14를 대입한다. 3.14는 실수이므로 `double`이나 `float`로 선언해야 하며 만약 `int`로 선언을 하면 잘못된 결과가 나오게 된다.
- `printf("원주 = %d * 2 * %lf = %lf \n", r, pi, r * 2 * pi);`
실수형 자료를 "%lf" 형식으로 출력하면 소수 7자리에서 반올림하여 소수점 이하 6자리까지 출력된다. 정수와 실수의 혼합계산 결과는 실수가 된다.

자가진단 4

다음 두 값을 변수에 저장하고 곱셈 계산식을 출력하는 프로그램을 작성하라.
추의 무게 = 49 , 중력의 비율 = 0.2683

출력 예

$$49 * 0.268300 = 13.146700$$

05

Question

화면에 다음과 같이 출력하는 프로그램을 작성하시오.

출력 예

전체 7자리로 맞추고 소수 4자리까지 출력

$x = 1.2340$

$y = 10.3459$

소수 2자리까지 출력(반올림)

$x = 1.23$

$y = 10.35$

Chapter 2

입력

소스

```
#include <stdio.h>

int main()
{
    double x = 1.234;
    double y = 10.3459;

    printf("전체 7자리로 맞추고 소수 4자리까지 출력 \n");
    printf("x = %7.4lf \n", x);
    printf("y = %7.4lf \n", y);
    printf("\n");

    printf("소수 2자리까지 출력(반올림) \n");
    printf("x = %.2lf \n", x);
    printf("y = %.2lf \n", y);

    return 0;
}
```

설명

- “%7.4lf”

여기서 7은 소수점을 포함한 전체 자릿수, 4는 소수점 이하 자릿수를 나타낸다.

즉,

		.				
--	--	---	--	--	--	--

 와 같은 형태로 출력이 된다. 소수점 앞부분에 숫자가 없는 경우에는 공백을 출력하고 소수점 뒷부분은 0을 출력한다. 전체 글자수가 7을 넘어가면 무시하고 출력하며 소수점 뒷부분이 넘어가면 반올림하여 출력된다.

출력 예

1.234

	1	.	2	3	4	0
--	---	---	---	---	---	---

123

1	2	3	.	0	0	0	0
---	---	---	---	---	---	---	---

0.012356

	0	.	0	1	2	4
--	---	---	---	---	---	---

- “%.2lf”

전체 자릿수는 실제로 출력하고 소수점 이하는 소수 3자리에서 반올림하여 소수 2자리까지 출력한다.

자가진단 5

1야드(yd)는 91.44cm이고 1인치(in)는 2.54cm이다.
2.1야드와 10.5인치를 각각 cm로 변환하여 다음 형식에 맞추어 소수 첫째자리까지 출력하시오.

출력 예

2.1yd = 192.0cm

10.5in = 26.7cm

- *Hint*

```
printf("%4.1f yd = %5.1f cm\n", 변수, 식)
```

06

Question

나이를 키보드로 입력 받아서 다음과 같이 출력하시오

입력 예 당신의 나이는 몇 살입니까? 14

출력 예 당신의 나이는 14살이군요.

소스

```
#include <stdio.h>

int main()
{
    int age;

    printf("당신의 나이는 몇 살입니까? ");
    scanf("%d", &age);

    printf("당신의 나이는 %d살 이군요. \n", age);

    return 0;
}
```

설명

- `scanf("%d", &age);`

키보드를 통해 변수 `age`의 값을 입력받는다. `printf()`와 같이 자료형의 형식이 같아야 한다. 입력에서는 변수 앞에 반드시 `'&'` 문자를 붙여 주어야 하는데 해당 변수가 위치한 메모리에 입력 값을 넣으라는 의미이다.

- 혹시 위의 소스를 그대로 입력하고 실행을 했는데 “당신의 나이는 몇 살입니까?”라고만 출력되고 프로그램이 멈추었다고 고민하는 중이라면 지금 즉시 키보드로 14를 입력하고 Enter 키를 눌러보자.

- 위의 입력 예에서 이탤릭체로 나타낸 부분(“당신의 나이는 몇 살입니까?”)은 입력 받기 전에 먼저 출력을 해야 하는 문자열을 의미한다.

자가진단 6

키를 입력받아 출력하는 프로그램을 작성하라.

입력 예

height = 170

출력 예

Your height is 170cm.

- *hint*

"height = " 은 출력을 하는 부분으로 printf("height = "); 을 먼저 실행한 후 입력을 받아야 한다.

07

Question

두 개의 정수를 입력 받아 합과 곱을 출력하시오.

입력 예 두 수를 입력하시오. 3 5

출력 예 $3 + 5 = 8$

$3 * 5 = 15$

Chapter 2

입력

소스

```
#include <stdio.h>

int main()
{
    int a, b;

    printf("두 수를 입력하시오. ");
    scanf("%d %d", &a, &b);
    printf("%d + %d = %d \n", a, b, a+b);
    printf("%d * %d = %d \n", a, b, a*b);

    return 0;
}
```

설명

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

두 개 이상의 변수를 입력받을 때에도 `printf()`에서와 마찬가지로 변환지정 형식과 개수를 인수의 형식과 개수에 맞추어서 사용하면 된다.

자가진단 7

두 개의 정수를 입력 받아 곱과 몫을 출력하시오

입력 예

16 5

출력 예

16 * 5 = 80

16 / 5 = 3

08

Question

키와 몸무게 영문이름의 첫 자를 입력 받아 출력하는 프로그램을 작성하시오 (몸무게는 반올림하여 소수 첫째자리까지 출력한다.)

입력 예

키를 입력하세요. 175

몸무게를 입력하세요. 76.36

이름을 입력하세요. K

출력 예

키 = 175

몸무게 = 76.4

이름 = K

Chapter 2

입력

소스

```
#include <stdio.h>

int main()
{
    int height;
    double weight;
    char name;

    printf("키를 입력하세요. ");
    scanf("%d", &height);

    printf("몸무게를 입력하세요. ");
    scanf("%lf", &weight);

    printf("이름을 입력하세요. ");
    scanf(" %c", &name);

    printf("키 = %d \n", height);
    printf("몸무게 = %.1f \n", weight);
    printf("이름 = %c \n", name);

    return 0;
}
```

설명

- `scanf("%lf", &weight);`

한 개의 실수를 입력받아 `weight`에 저장한다. `double`로 선언된 변수를 입력받기 위해서는 반드시 “%lf”를 지정해야 한다.

- `scanf(" %c", &name);`

한 개의 문자를 입력받아 `name`에 저장한다.

“%c” 앞에 공백을 넣어 준 것에 유의해야 한다. 일반적인 자료형의 경우에는 이전에 입력된 `[Enter ↵]` 나 공백문자, 탭문자(`[Tab]`) 등을 입력구분자로 처리하여 무시하지만 문자형 자료(`char`)의 경우에는 모든 문자를 입력으로 처리하게 된다. 만약 공백을 생략하게 되면 위에서 `weight`를 입력받은 후 누른 `[Enter ↵]`가 버퍼에 저장되어 있다가 위의 입력명령을 만나면 `name`에 저장되어 엉뚱한 결과가 출력된다.

문자형 자료에서 위와 같은 특수문자를 입력구분자로 처리하기 위하여 “%c” 앞에 공백을 넣어준 것이며 이렇게 하면 버퍼에 저장된 특수문자를 무시하고 새로운 문자를 입력으로 받게 되는 것이다.

- 문자형 자료(char)는 1byte로 구성되어 있어서 영문자나 숫자 등의 문자 한 개만을 입력받아 처리할 수 있다. 한글은 한 글자가 2byte(또는 3byte)이므로 입력받을 수 없다. 한글을 포함하여 단어나 문장과 같은 문자열을 입력받아 처리하는 방법은 뒤에서 배울 것이다.

자가진단 8

실수 2개와 한 개의 문자를 입력 받아 출력하되 실수는 소수 셋째자리에서 반올림하여 소수 둘째자리까지 출력하는 프로그램을 작성하시오

입력 예

12.2536

526.129535

A

출력 예

12.25

526.13

A

09

Question

float와 double형의 두 개의 실수를 입력 받아 반올림하여 소수 둘째자리까지 출력하는 프로그램을 작성하십시오.

입력 예

두 개의 실수를 입력하십시오.

2.9402

415.23968125

출력 예

x = 2.94

y = 415.24

Chapter 2

입력

소스

```
#include <stdio.h>

int main()
{
    float x;
    double y;
    printf("두 개의 실수를 입력하십시오.\n");
    scanf("%f %lf", &x, &y);
    printf("x = %.2f \n", x);
    printf("y = %.2lf \n", y);
    return 0;
}
```

설명

- `scanf("%f %lf", &x, &y);`

“%f”는 float형 실수, “%lf”는 double형 실수를 지정하는 것이다. float형은 전체의 길이가 6자리를 넘어가게 되면 정확한 결과를 기대할 수 없다.

123456789.1234와 같이 긴 실수를 입력해서 결과를 출력해 보면 float형은 정확한 결과가 출력되지 않음을 알 수 있다.

자가진단 9

두 개의 float형 실수와 한 개의 double형 실수를 입력 받아 소수 4자리에서 반올림하여 소수 3자리까지 출력하는 프로그램을 작성하시오.

입력 예

1.2568
95.21438
526.851364

출력 예

1.257
95.214
526.851

Thank You!!!

