

기초프로그래밍

제3장 변수

Sangsoo Lim

CSAI

Dongguk University

차례

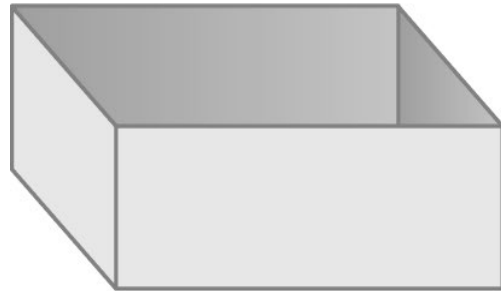
- 변수 선언하기
- 변수 선언 시 주의할 점

변수 선언하기

- 변수란?

- 데이터를 저장하는 임시 저장 공간
- 메모리 공간

임시 저장 공간



변수



카트

변수 선언하기

- 변수 a, b 만들어 보기

```
/* 3-1.c */
#include<stdio.h>
int main(void)
{
    int a;        // 변수 a (메모리공간 a)
    int b;        // 변수 b (메모리공간 b)

    return 0;
}
```

변수 선언하기

- 변수 선언 방법 - 변수의 종류
 - 정수형 변수: char형, short형, int형, long형
 - 실수형 변수: float형, double형, long double형

```
/* 3-2.c */
#include<stdio.h>
int main(void)
{
    int a;      // 정수형 변수 선언
    float b;    // 실수형 변수 선언

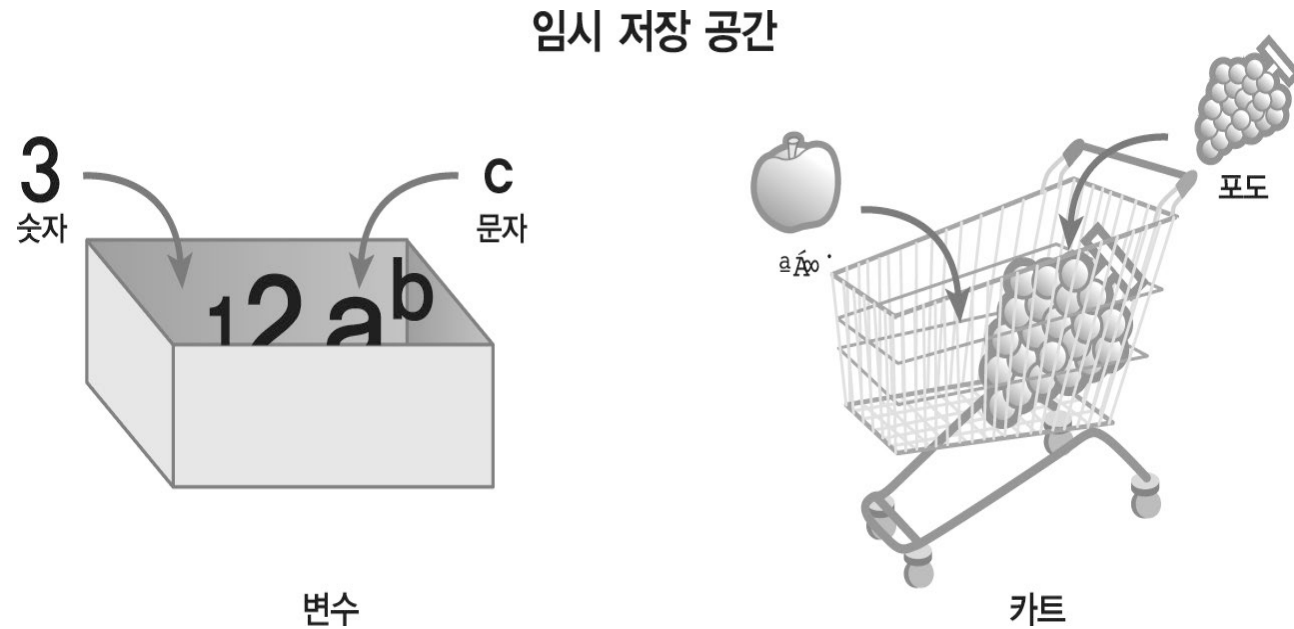
    return 0;
}
```

```
/* 3-3.c */
#include<stdio.h>
int main(void)
{
    int a, b;    // 정수형 변수 선언
    float c, d; // 실수형 변수 선언

    return 0;
}
```

변수 선언하기

- 데이터란?
 - 변수에 저장되는 값



변수 선언하기

- 예제 3-4.c 실습

```
/* 3-4.c */
#include<stdio.h>
int main(void)
{
    int a;
    int b;

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

    return 0;
}
```

변수 선언하기

- 예제 3-5.c 실습

```
/* 3-5.c */
#include<stdio.h>
int main(void)
{
    int a=0;
    int b=1;

    printf("a의 값은 %d 입니다. \n", a);
    printf("b의 값은 %d 입니다. \n", b);

    return 0;
}
```


변수 선언하기

- 변수에 저장한 데이터는 변경될 수 있다 (변수는 임시공간)

```
/* 3-6.c */
#include<stdio.h>
int main(void)
{
    int a=0;
    int b=1;

    printf("a의 값은 %d 입니다. \n", a);
    printf("b의 값은 %d 입니다. \n", b);

    a = a+10;
    b = b+10;

    printf("변경된 a의 값은 %d 입니다. \n", a);
    printf("변경된 b의 값은 %d 입니다. \n", b);
    return 0;
}
```

변수 선언 시 주의할 점

- 변수 선언은 제일 앞쪽에 한다

```
/* 3-7.c */
#include<stdio.h>
int main(void)
{
    int a; // 정상
    int b; // 정상

    a=1;
    b=2;

    int c; // 에러 발생
    ...
}
```

변수 선언 시 주의할 점

- 변수의 이름은 의미 있게 짓는다

```
/* 3-8.c */
#include<stdio.h>
int main(void)
{
    int appleBox = 30;
    int grapeBox = 20;
    int total;

    total = appleBox + grapeBox;
    printf("총 %d 박스가 있습니다 \n", total);

    return 0;
}
```

변수 선언 시 주의할 점

- 변수의 이름을 지을 때 주의할 점

① 특수 기호, 공백 문자, 맨 처음 숫자를 사용하면 안된다

변수 이름이 올바른 경우	변수 이름이 잘못된 경우	잘못된 이유
<code>int Apple;</code>	<code>int ?apple;</code>	특수 문자 ? 사용
<code>int total;</code>	<code>int to tal;</code>	to 와 tal 사이에 공백 문자 사용
<code>int result2;</code>	<code>int 2result;</code>	맨 처음에 숫자 사용

② c언어에서 사용되는 키워드를 변수 이름에 사용하면 안된다

변수 이름이 잘못된 경우	잘못된 이유
<code>int int;</code>	키워드 int 사용
<code>int long;</code>	키워드 long 사용
<code>int short;</code>	키워드 short 사용

변수 선언 시 주의할 점

- 변수의 이름을 지을 때 주의할 점

③ C언어는 대소문자를 구분한다

대문자 변수 이름	소문자 변수 이름	설명
int Apple;	int apple;	같은 변수가 아닙니다.
int TOTAL;	int total;	같은 변수가 아닙니다.
int result;	int result;	같은 변수가 아닙니다.

④ ANSI(American National Standards Institute)에 표준화된 키워드들

auto	break	case	char	const	continue
default	do	double	else	enum	extern
float	for	goto	if	int	long
register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void
volatile	while

변수의 시작 주소와 &연산자

- 변수의 시작주소

- 변수 앞에 &를 붙이면 변수의 시작 주소를 알 수 있다

```
/* 3-9.c */
#include <stdio.h>
int main(void)
{
    int a=3;
    int b=4;
    printf("a의 값: %d \n", a);
    printf("b의 값: %d \n", b);

    printf("변수 a의 시작 주소: %X \n", &a);

    printf("변수 b의 시작 주소: %x \n", &b);
    return 0;
}
```

```
a의 값: 3
b의 값: 4
변수 a의 시작 주소: DD137E20
변수 b의 시작 주소: dd137e24
```

Summary

- 변수란 무엇인가?
- 데이터란 무엇인가?
- 변수에 저장된 데이터는 변경 가능
- 변수를 선언하는 방법
- 변수 선언 시 주의할 점과 변수의 시작 주소

1. 기본 변수 선언 및 초기화

- 예제 1-1: 여러 유형의 변수 선언

```
#include <stdio.h>

int main(void)
{
    char c = 'A';           // 문자형 변수
    short s = 100;          // short 정수형 변수
    int i = 10000;          // int 정수형 변수
    long l = 100000L;       // long 정수형 변수 (접미사 L)
    float f = 3.14f;        // float 실수형 변수 (접미사 f)
    double d = 3.141592;    // double 실수형 변수

    // 출력
    printf("char   : %c\n", c);
    printf("short  : %d\n", s);
    printf("int    : %d\n", i);
    printf("long   : %ld\n", l);
    printf("float  : %f\n", f);
    printf("double : %lf\n", d);

    return 0;
}
```


2. 변수의 초기화와 변경

- 예제 2-1: 변수값 변경

```
#include <stdio.h>

int main(void)
{
    int num = 0;    // 초기화
    printf("초기 num의 값은 %d 입니다.\n", num);

    num = 5;        // 새로운 값 대입
    printf("변경 후 num의 값은 %d 입니다.\n", num);

    num = num + 10; // 자기 자신과의 연산
    printf("또다시 변경 후 num의 값은 %d 입니다.\n", num);

    return 0;
}
```

3. 변수 사용 시 주의점(초기화, 범위, 네이밍)

- 예제 3-1: uninitialized(초기화되지 않은) 변수의 문제점

```
#include <stdio.h>

int main(void)
{
    int x;    // 초기화 안 함
    int y = 10;

    // x가 초기화되지 않았으므로, 예상치 못한 값이 출력될 수 있음
    printf("초기화되지 않은 x의 값: %d\n", x);
    printf("초기화된 y의 값: %d\n", y);

    // x에 값을 대입한 뒤 출력
    x = 5;
    printf("이제 초기화한 x의 값: %d\n", x);

    return 0;
}
```

3. 변수 사용 시 주의점(초기화, 범위, 네이밍)

- 예제 3-2: 올바른 변수 이름 짓기

```
#include <stdio.h>

int main(void)
{
    int numberOfApples = 20;    // Camel-case 사용 예시
    int apple_boxes = 3;       // Snake-case 사용 예시
    // int 2apples;             // 컴파일 오류 (맨 앞에 숫자 불가능)
    // int short;               // 컴파일 오류 (키워드 사용 불가능)
    // int apple@tree;          // 컴파일 오류 (@ 기호 불가능)

    printf("사과는 총 %d개, 박스는 %d박스 있습니다.\n", numberOfApples, apple_boxes);
    return 0;
}
```

4. 변수의 주소(& 연산자)

- 예제 4-1: 주소 연산자 &

```
#include <stdio.h>

int main(void)
{
    int a = 3;
    int b = 4;

    printf("a의 값: %d\n", a);
    printf("b의 값: %d\n", b);

    // 주소 출력 시, %p나 %x(16진수) 사용 (컴파일러마다 권장 형식 다를 수 있음)
    printf("변수 a의 시작 주소: %p\n", (void*)&a);
    printf("변수 b의 시작 주소: %p\n", (void*)&b);

    return 0;
}
```

4. 변수의 주소(& 연산자)

- 예제 4-2: 여러 변수의 주소 비교

```
#include <stdio.h>

int main(void)
{
    int x = 10;
    double y = 3.14;
    char ch = 'C';

    printf("x의 값 = %d,   x의 주소 = %p\n", x, (void*)&x);
    printf("y의 값 = %.2f,   y의 주소 = %p\n", y, (void*)&y);
    printf("ch의 값 = %c,   ch의 주소 = %p\n", ch, (void*)&ch);

    return 0;
}
```

1. 소량의 재고 관리

1. 프로그램 기능

- 사과(apple)의 재고 수와 바나나(banana)의 재고 수를 변수에 저장한다.
- 현재 총 과일 개수(사과 + 바나나)를 계산하여 출력한다.
- 이후 바나나가 일정 개수 입고되거나 사과가 판매되는 등의 상황을 변수에 반영하고, 다시 총 개수를 출력한다.

2. 예시 시나리오

- 초기 상태: 사과 10개, 바나나 8개
- 바나나가 5개 더 들어옴(+5)
- 사과가 2개 판매됨(-2)

현재 사과 개수: 10
현재 바나나 개수: 8
총 과일 개수: 18

[재고 변경 후]
현재 사과 개수: 8
현재 바나나 개수: 13
총 과일 개수: 21

2: 두 정수의 교환(Swap)

1. 프로그램 기능
 - int형 변수 x , y 에 임의의 값을 대입한다.
 - 임시 변수 $temp$ (또는 다른 이름의 변수)를 사용하여 x 와 y 의 값을 교환한다.
 - 교환 전과 교환 후의 x , y 값을 각각 출력한다.
2. 예시 시나리오
 - 초기값: $x = 5, y = 10$
 - 교환 후: $x = 10, y = 5$

초기 상태: $x=5, y=10$
교환 후 : $x=10, y=5$

3: 직사각형 넓이와 둘레 계산

1. 프로그램 기능

- 사각형의 가로 길이(length)와 세로 길이(width)를 실수형(double) 변수로 선언해 직접 값 대입(또는 scanf로 입력받아도 무방)한다.
- 직사각형의 넓이와 둘레를 계산하여 출력한다.
 - 넓이(Area): $\text{length} * \text{width}$
 - 둘레(Perimeter): $2.0 * (\text{length} + \text{width})$

2. 예시 시나리오

- $\text{length} = 12.5$, $\text{width} = 4.2$ 와 같이 실수값을 대입한다.
- 계산 결과를 소수점 둘째 자리 혹은 셋째 자리까지 표현해본다(서식 지정을 활용).

가로 길이: 12.50, 세로 길이: 4.20
직사각형의 넓이: 52.50
직사각형의 둘레: 33.40