



## 4. 변수

### ■ 변수를 이용한 데이터의 저장

- 변수란? 변하는 숫자의 줄임말인데, 이러한 변하는 값(데이터)가 저장(메모리)되는 공간을 변수라고 한다.
- 변수를 사용하려면 반드시 미리 만들어야(선언해야) 한다.

<pre>#include&lt;stdio.h&gt; int main() {     ① int num;     ② num = 20;     printf("%d", num);     return 0; }</pre>	<p>① <code>int num;</code> (변수 선언 = 공간 할당)</p>  <p>② <code>num = 20;</code> (변수에 값 저장)</p> 
---	--

- 변수선언문 : `int num;`
- `int` : 정수 저장이 가능한 저장(메모리) 공간을 할당
- `num` : 그 메모리 공간의 이름을 `num`이라 함.

```
int num;           // num 이라는 이름의 변수 선언
num = 20;          // 변수 num에 20 저장
printf("%d", num); // 변수 num의 값 출력
```

- C언어에서 `=` 기호는 값의 대입을 뜻함. '대입 연산자'라고 함.

### ■ 변수의 다양한 선언 및 초기화 방법

- 선언된 변수에 처음으로 값을 저장하는 것을 가리켜 '초기화'라고 함.
- 초기화 이후에 저장된 값을 변경할 때에는 '대입' 또는 '대입연산'을 진행한다고 함.

```
int num;           // num 이라는 이름의 변수 선언
num = 12;          // 변수 num에 12로 초기화
num = 24;          // 변수 num에 24를 대입
```

- 선언과 동시에 초기화(변수 `num`이 메모리 공간에 할당되자마자 12로 초기화 됨)

```
int num = 12;      // num 이라는 이름의 변수 선언 및 값을 12로 초기화
```

- 둘 이상의 변수를 동시에 선언, 선언 및 초기화


```
int num1, num2;    // 두 개의 변수를 선언
int num3 = 30, num4 = 40; // 두 개의 변수를 선언 및 초기화
```

## ■ 변수의 여러 종류

종류	선언문
정수형 변수	int
실수형 변수	float
	double
문자형 변수	char

### ● 실습 예제

실행결과



// 예제(왼쪽 실행 결과를 보고 main 함수의 빈칸을 채워 보세요.)

```

#include<stdio.h>
int main()
{
    int num1 = 1;
     num2 = 1.0;
     c = 'a';
    

    return 0;
}

```

### ● 실습 예제

```

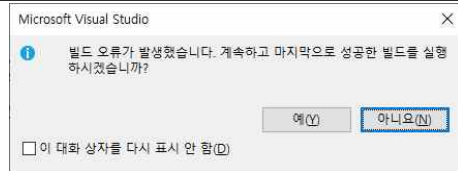
// 예제
#include <stdio.h>
int main()
{
    int num1, num2;
    int num3 = 30, num4 = 40;

    printf("num1: %d, num2: %d \n", num1, num2);
    num1 = 10;
    num2 = 20;

    printf("num1: %d, num2: %d \n", num1, num2);
    printf("num3: %d, num4: %d \n", num3, num4);
    return 0;
}

```

## ● 실행결과



error C4700: 초기화되지 않은 'num1' 지역 변수를 사용했습니다.

error C4700: 초기화되지 않은 'num2' 지역 변수를 사용했습니다.

- 에러 나는 이유 : 변수를 초기화하지 않으면 쓰레기 값(의미 없는 값)이 저장되기 때문

## ● 실습 예제(수정)

```
// 예제
#include <stdio.h>
int main()
{
    int num1 = 0, num2 = 0;
    int num3 = 30, num4 = 40;

    printf("num1: %d, num2: %d \n", num1, num2);
    num1 = 10;
    num2 = 20;

    printf("num1: %d, num2: %d \n", num1, num2);
    printf("num3: %d, num4: %d \n", num3, num4);
    return 0;
}
```

### 실행결과

Microsoft Visual Studio 디버그 콘솔

```
num1: 0, num2: 0
num1: 10, num2: 20
num3: 30, num4: 40
```

## ■ 변수선언 시 주의해야 할 사항

- 변수 이름을 지을 때 적용되는 규칙들 -용도에 맞는 유의미한 이름으로 짓는 것이 바람직하다.

첫째, 변수의 이름은 알파벳, 숫자, 언더바(\_)로 구성된다.

둘째, C언어는 대소문자를 구분한다. 따라서 변수 Num과 변수 num은 서로 다른 변수이다.

셋째, 변수의 이름은 숫자로 시작할 수 없고, 키워드도 변수의 이름으로 사용할 수 없다.

넷째, 이름 사이에 공백이 삽입될 수 없고 특수 문자를 사용할 수 없다.

## ■ C언어의 표준 키워드(keyword)

- 기능적 의미가 정해져서 C언어의 문법을 구성하는 단어들이
- 프로그래머가 다른 용도로 사용할 수 없도록 제한됨

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

## ■ 덧셈 프로그램 완성

- `int result;` 변수를 선언하고 덧셈의 결과를 저장

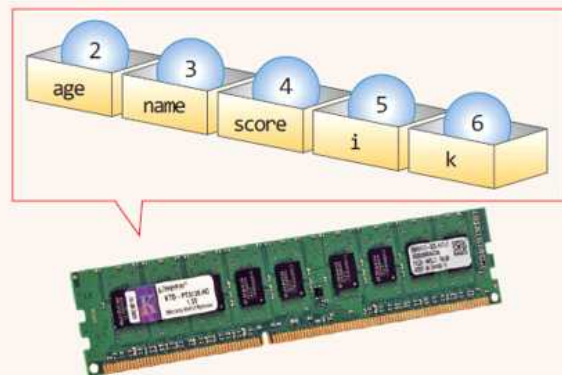
	<pre>// 왼쪽과 같이 실행결과를 위해서 아래 프로그램 main() 함수를 작성해보세요!! #include &lt;stdio.h&gt; int main() {     int num1 = 3;     int num2 = 4;      return 0; }</pre>
--	---

## ■ 참고로 읽어봅시다.



### 참고: 변수는 어디에 만들어 질까?

변수는 물리적으로 컴퓨터의 어디에 만들어지는 것일까? 변수는 메모리(memory)에 만들어 진다. 우리는 프로그램 안에서 변수를 만들고 변수에 이름을 부여한 다음, 변수 이름을 사용하여 메모리 공간을 사용하게 된다. 만약 변수를 이용하지 않으면 메모리의 주소를 가지고 데이터를 저장하여야 할 것이다. "300번지에 정수 20을 저장하라"와 같이 주소를 이용하여 메모리를 사용하는 것은 가능한 방법이지만 인간에게는 상당히 불편한 방법이다. 변수라는 개념을 사용할 수 있기 때문에 특별한 경우를 제외하고는 우리는 메모리를 주소로 접근할 필요가 없다.



### Tip: 좋은 변수 이름

변수의 이름을 짓는 것은 상당히 중요한 작업 중의 하나이므로 신중하고 시간을 투자해야 한다. 변수의 이름을 지을 때는 변수의 역할을 가장 잘 설명하는 이름을 지어야 한다. 좋은 변수 이름은 전체 프로그램을 읽기 쉽게 만든다. 하지만 반대로 즉흥적으로 지은 이름을 사용하게 되면 나중에 프로그램을 읽기가 아주 힘들어진다. 예를 들면 연도와 달, 일을 나타내는데 `i`, `j`, `k`라고 이름을 짓는 것보다 `year`, `month`, `date`라고 하면 프로그램이 읽기 쉬워질 것이다. 영어 단어만을 사용해야 하므로 한영 사전을 이용하여 한글을 영문으로 바꾸는 것도 좋은 아이디어이다.



### Tip: 여러 단어로 된 변수 이름

여러 단어로 되어 있는 변수 이름을 나타내는 데 몇 가지의 방식이 존재한다. 먼저 가장 전통적인 방법은 `bank_account`처럼 중간에 밑줄 문자를 사용하는 것이다. 두 번째 방법은 `BankAccount`처럼 단어의 첫 번째 글자를 대문자로 하는 것이다. 어떤 방식도 사용해도 상관없고 다만 일관성있게 사용하면 된다. 이 책에서는 전통적인 C의 방법을 따라서 밑줄 문자를 사용하여 단어들을 분리하였다.

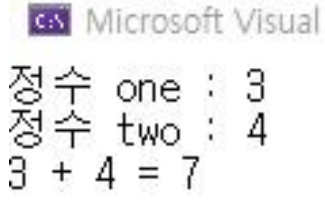
## 5. 키보드로부터 데이터 입력과 C언어의 키워드

### ■ 키보드로부터 정수입력을 위한 scanf() 함수의 호출

- scanf() 함수 : 키보드로부터 다양한 형태의 데이터를 입력받을 수 있음.

```
#define _CRT_SECURE_NO_WARNINGS // visual studio 만 필요! #pragma warning(disable:4996) 이것도 가능
#include <stdio.h>
int main()
{
    int num;
    scanf("%d", &num);
    printf("%d", num);
    return 0;
}
```

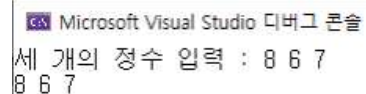
- 두 개의 정수를 입력받아 두 정수의 합을 출력하는 예제 작성

	<pre>// 왼쪽과 같이 실행결과를 위해서 아래 프로그램 main() 함수를 작성해보세요!! #define _CRT_SECURE_NO_WARNINGS // visual studio 만 필요! #include &lt;stdio.h&gt; int main() {     return 0; }</pre>
--	---

### ■ 입력의 형태를 다양하게 지정

```
#define _CRT_SECURE_NO_WARNINGS // visual studio 만 필요!
#include <stdio.h>
int main()
{
    int num1, num2, num3;
    printf("세 개의 정수 입력 : ");
    scanf("%d %d %d", &num1, &num2, &num3);
    printf("%d %d %d\n", num1, num2, num3);
    return 0;
}
```

#### 실행결과



## 문제 02-1 [scanf 함수의 활용]

아래의 문제들은 프로그램에 익숙해질 수 있도록 비교적 쉬운 문제들로 구성하였으니, 본인 스스로의 힘으로 해결할 수 있도록 노력해주세요~!


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

예시)  Microsoft Visual Studio 디버그 콘솔

```
정수 두 개를 입력하세요.(0 제외) : 756 38
756 * 38 = 28728
756 / 38 = 19
```

2. 프로그램 사용자로부터 세 개의 정수 num1, num2, num3를 순서대로 입력받은 후에, 다음 연산의 결과를 출력하는 프로그램을 작성해보자.

num1\*num2+num3

예시)  Microsoft Visual Studio 디버그 콘솔

```
세 개의 정수를 입력하세요.(0 제외) : 3 4 5
3 * 4 + 5 = 17
```


3. 하나의 정수를 입력받아서 그 수의 제곱의 결과를 출력하는 프로그램을 작성해보자.

예시)  Microsoft Visual Studio 디버그 콘솔

```
정수 한 개를 입력하세요.(0 제외) : 15
15의 제곱은 225
```

4. 입력받은 두 정수를 나누었을 때 얻게 되는 몫과 나머지를 출력하는 프로그램을 작성해보자.


예시)

```
 Microsoft Visual Studio 디버그 콘솔
두 개의 정수를 입력하세요.(0 제외) : 87 6
87 / 6 몫 : 14
87 % 6 나머지 : 3
```

5. 입력받은 세 개의 정수 num1, num2, num3를 대상으로 다음 연산의 결과를 출력하는 프로그램을 작성해보자.

(num1-num2)\*(num2+num3)\*(num3%num1)

예시)

```
 Microsoft Visual Studio 디버그 콘솔
세 개의 정수를 입력하세요.(0 제외) : 10 9 8
연산결과 : 136
```