

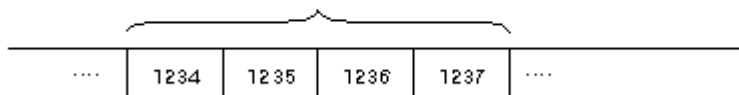
변수

“변수란 값을 저장할 수 있는 메모리 공간에 붙은 이름, 혹은 메모리 공간 자체를 가리켜 변수라고 한다.”

변수는 말뜻 그대로 풀어 보자면 변할 수 있는 수, 즉 고정되어 있지 않은 수라는 뜻이며 1이나 45 또는 3.14같은 상수의 반대 개념이다. 이름은 변수이지만 반드시 수치값만 저장되는 것은 아니며 문자열이나 포인터같은 좀 더 복잡한 값도 저장될 수 있다. 변수에서 말하는 수(數)를 좀 더 일반적으로 표현하면 데이터(data)이며 한국말로 알아듣기 쉽게 설명하면 값이다.

이야기를 바꿔 컴퓨터는 연산 대상 메모리의 위치를 구분하기 위해 숫자로 된 번지(Address)라는 개념을 사용한다. 메모리를 구성하는 각 바이트들은 0부터 시작해서 약 40억까지의 고유한 번지를 가지고 있으며 이 번지를 대상으로 값을 읽거나 쓴다. 예를 들어 “1234번지에서 1237번지까지의 4바이트에 56을 기록한다”는 식으로 동작한다.

이 메모리 영역을 Num이라고 선언한다.



이러한 방식은 사람의 입장에서 보면 굉장히 다루기 힘든 형태로 되어 있다. 예를 들어 어떤 값이 0x183c7eda번지에 저장되어 있다면 이런 값을 10개만 사용해도 사람의 머리로는 각 값을 구분할 수 없다.

그래서 좀 더 기억하기 쉬운 변수를 사용한다. 숫자로 된 번지에 별도의 이름을 붙여 놓은 것이 바로 변수이다. 사람은 숫자보다는 이름을 더 잘 기억하며 이름에 설명적인 의미를 부여할 수 있기 때문에 여러 개를 쓰더라도 구분하기 어렵지 않다.

$[0x3e98234a] = [0x3eff81a0] + [0x3d00aef6]$

이런 숫자, 그것도 16진수의 나열보다는 salary=pay+bonus가 사람에게 훨씬 더 쉬우며 변수의 이름이 대상을 잘 표현할수록 더욱 쉬워진다. 흔히 변수는 값을 저장하는 상자에 비유되곤 한다. 상자에 물건을 넣어 두듯이 변수를 하나 만들어 놓으면 여기에 값을 저장하거나 다시 꺼낼 수 있다.

변수는 C언어의 구성 요소 중에 명칭으로 분류된다. 그래서 명칭을 작성하는 일반적인 규칙대로 자유롭게 작성할 수 있다. 키워드는 쓸 수 없고 영문, 숫자, 밑줄 문자로만 구성되어야 하며 대소문자를 구분한다. 이런 문법적인 제약 외에도 기억하기 쉽게 하기 위해 변수를 사용하는 만큼 변수명을 작성하는 일반적인 몇 가지 법칙이 있다.

① 의미를 잘 설명할 수 있는 이름을 주는 것이 좋다. 의미 없는 이름을 쓰는 것도 물론 가능한 하지만 변수의 수가 많아지면 변수끼리 구분하기 어려워진다.

② 변수명의 길이에겐 제약이 없지만 3~10자 내외로 작성하는 것이 좋다. Num, TotalScore, NowStage 등이 좋은 예이다. 변수명을 길게 쓰면 의미를 분명하게 나타낼 수는 있지만 변수명을 기억하기 어려워지고 오타를 입력할 가능성이 많아져 바람직하지 않다.

③ 대소문자 구성을 일관되게 하는 것이 좋다. C언어는 대소문자를 구분하므로 Score와 score는 다른 변수이다. Score라고 해 놓고 쓸 때는 score라고 쓰면 이 변수는 선언되지 않은 것으로 에러 처리된다.

처음 실습을 할 때는 편의상 a, b, i같은 짧은 변수명을 즐겨 사용하지만 적절한 변수명을 붙이는 것은 아주 중요하다. 수백개의 변수들에 아무렇게나 이름을 붙여 놓으면 시간이 지난 후 자기 스스로도 코드를 파악하기 어려워지며 유지 보수하기에도 어려움이 생긴다.

Tip. 스타일 가이드 혹은 코드 컨벤션 등의 이름으로 검색하면 관습적으로 사용하는 여러 규칙들을 살펴볼 수 있다.