1. Int num1; / int number1, number2; /

int numer1=10, number2=20, result; (declare and initialize) /

Number 1= 6; => 가장 최근 숫자로 업데이트 됨

1. Flow num2;
2. Double num3;
3. Char val; -> 숫자 넣는거 불가능 (one character)

=> 4 date types in C language.

* Type specifier : Ex)%d(intiger),

scanf("%lf", &number1);

**double 함수 사용하면, %lf 사용**.

%p=>값이 더 길게 나옴

const double pi = 3.124;

pi = 8.9;

short int x = 89; -> 2bytes

long int x = 89; -> 4btytes

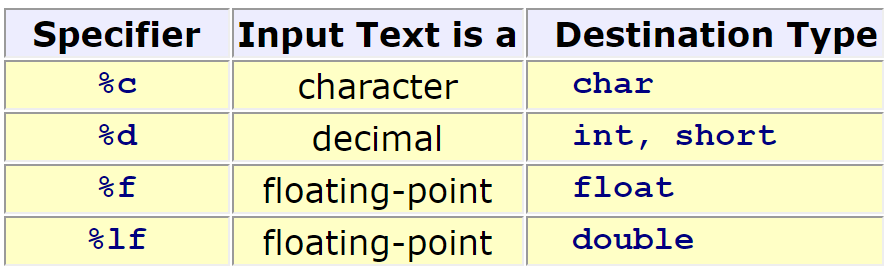
camel nofation=> ex-myLastName

const float vv = 6.76F;

double/int = 6.76. ;

**Format**

***format*** is a string literal that describes how to convert the text entered by the user into data stored in memory. ***format*** contains the conversion specifier for translating the input characters. Conversion specifiers begin with a % symbol and identify the type of the destination variable. The most common specifiers are listed below.



printf("NUM1: %lf NUM2: %lf Result is %.2lf\n", number1, number2, result );

printf("NUM1: %lf NUM2: %lf Result is %10.2lf\n", number1, number2, result );

NUM1: 12.000000 NUM2: 4.566660 Result is 16.57

NUM1: 12.000000 NUM2: 4.566660 Result is 16.57

printf("NUM1: %lf NUM2: %lf Result is XX%.2lf\n", number1, number2, result );

printf("NUM1: %lf NUM2: %lf Result is XX%-10.2lf\n", number1, number2, result );

NUM1: 12.000000 NUM2: 4.577770 Result is XX16.58

NUM1: 12.000000 NUM2: 4.577770 Result is XX16.58

int x = 8;

float y= 9.89;

float result = x \* y;

printf("RESULT = %f\n", result);

RESULT = 79.120003

int x = 8;

int y= 16;

int result = x / y;

printf("RESULT = %d\n", result);

RESULT = 0 => int는 정수만 가능

int x = 8;

int y= 16;

double result = x / (double)y;

printf("RESULT = %lf\n", result);

RESULT = 0.500000

double result =x / 16.0( 그냥 16은 안됨) ;