

숫자형, 불리언, 문자열 데이터 타입에 대하여 상세하게 배웁니다.  
각 데이터 타입에 대한 함수의 사용법에 대하여 배웁니다.

## **CHAPTER 2. 데이터 타입 및 함수**

---

# 1. 숫자형 데이터 타입과 함수

숫자형 데이터 타입

숫자형 함수

# 숫자형

- 부동소수 및 실수
- 정수

숫자형



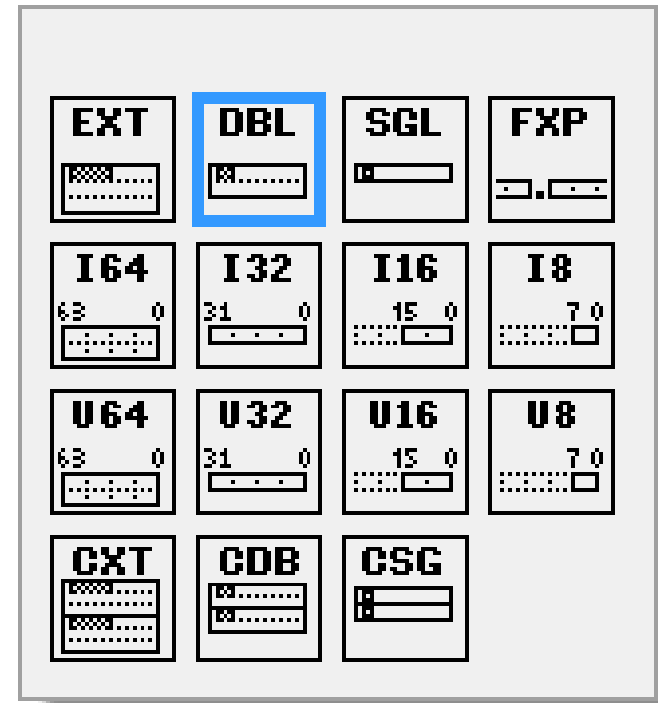
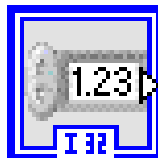
숫자형 2



숫자형



숫자형

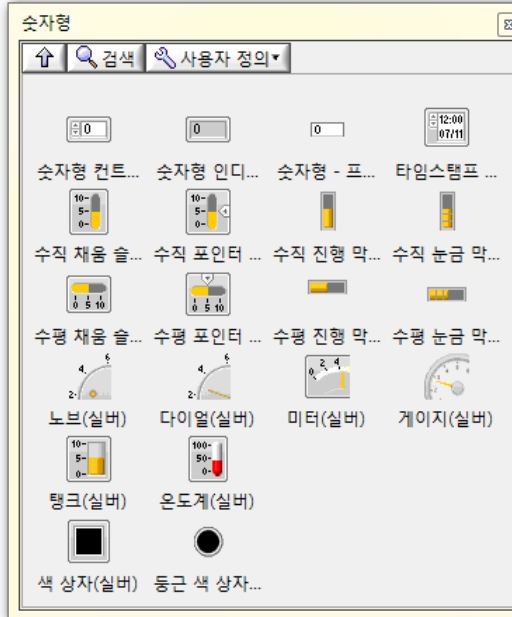


# 숫자형 컨트롤 팔레트

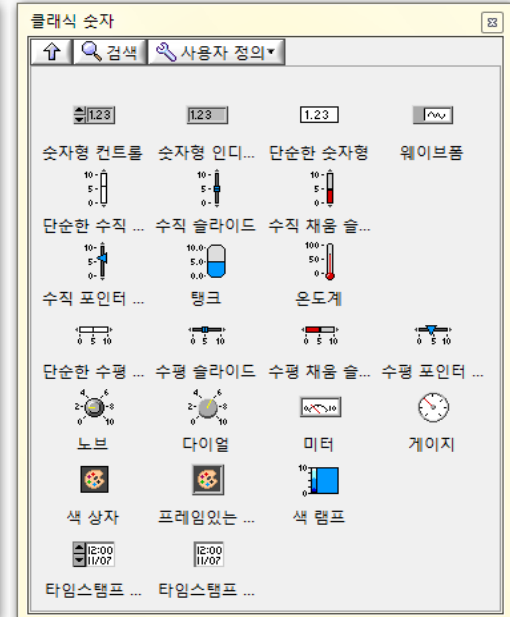
일반



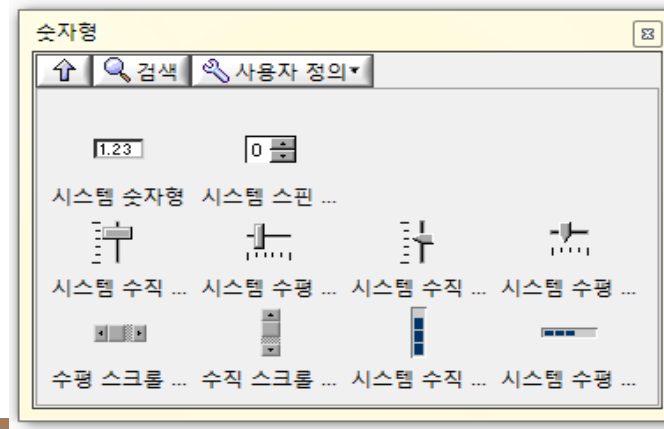
실버



클래식



시스템

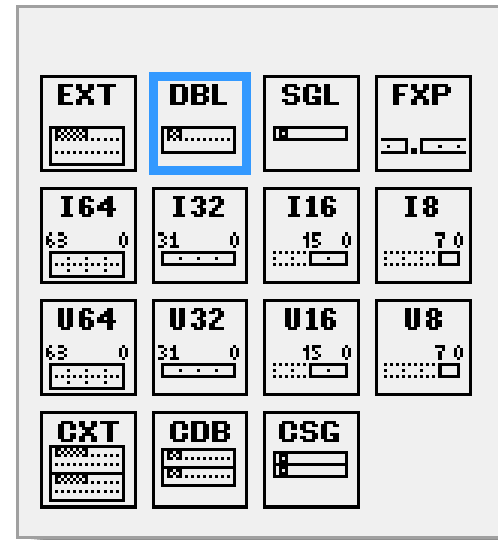


# 숫자형 데이터 타입

- 숫자형은 수학 연산을 수행하기 위한 목적의 데이터 타입.
- 머신은 0과 1의 **Binary** 데이터를 인식함.
- 사람은 자연수, 정수, 실수, 복소수를 사용함.
  - **자연수**는 컴퓨터에서 **Unsigned Integer**로 표현 (0과 양의 정수)
  - **정수**(Integer)는 이름 그대로 정수로 표현
  - **실수**는 고정소수점(Fixed Point)와 부동소수점(Floating Point)
  - **복소수**는 두 개의 실수 데이터로 구성되고, 각각 실수와 허수를 표현

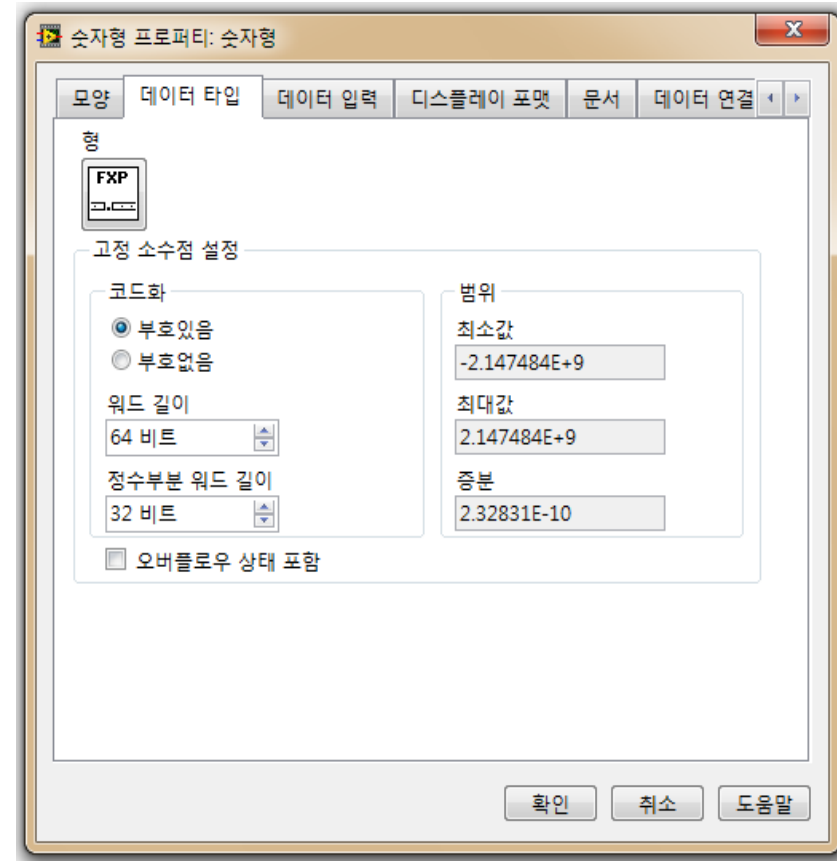
# 부동소수

- 부동소수는 ANSI/IEEE Standard 754-1985 규약 준수
  - 32bits 포맷의 단정도(SGL) 부동소수
  - 64bits 포맷의 배정도(DBL) 부동소수
  - 128bits 포맷의 확장형(EXT) 부동소수
- 부동소수(Floating)은 할당된 메모리 크기에서 소수점을 기준으로 정수 부분과 소수점 아래의 메모리 크기를 가변적으로 처리해주는 숫자 형임
  - 0.123456을 처리할 때와 12345.6을 처리할 때가 머신 입장에서는 확연히 다른데, 부동소수는 사람과 동일하게 소수점의 위치와 관계없이 숫자를 처리할 수 있습니다.
  - 숫자형 컨트롤이나 인디케이터의 기본 설정은 배정도(DBL)임.



# 고정소수

- 소수점을 기준으로 정수 부분과 소수점 아래, 그리고 부호를 명확하게 고정.
  - 부호 있음/없음
  - 전체 워드 길이
  - 정수부분의 워드 길이
- 최대 워드 길이는 **64비트**임.



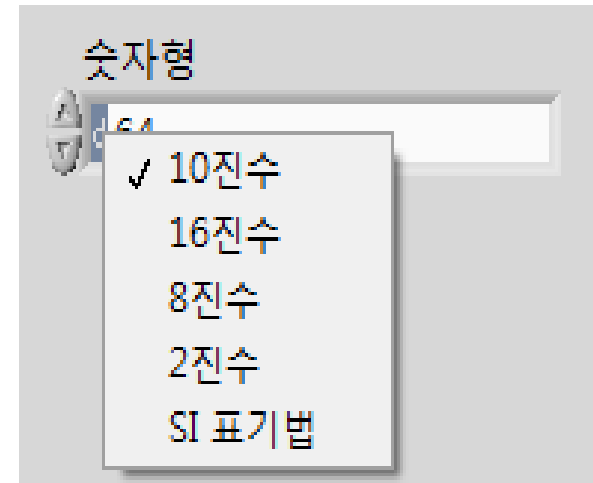
# 정수 Integer

- 보호 있는(signed)와 보호 없는(unsigned) 정수.
- 부호 있는 정수는 I8, I16, I32, I64로 네 가지
  - I8은 -128부터 127까지.
  - I16은 -32,768에서 32,787까지.
  - I32는 -2,147,483,648에서 2,147,483,647까지.
  - I64는 -9,223,372,036,854,775,808에서 9,223,372,036,854,775,807까지.



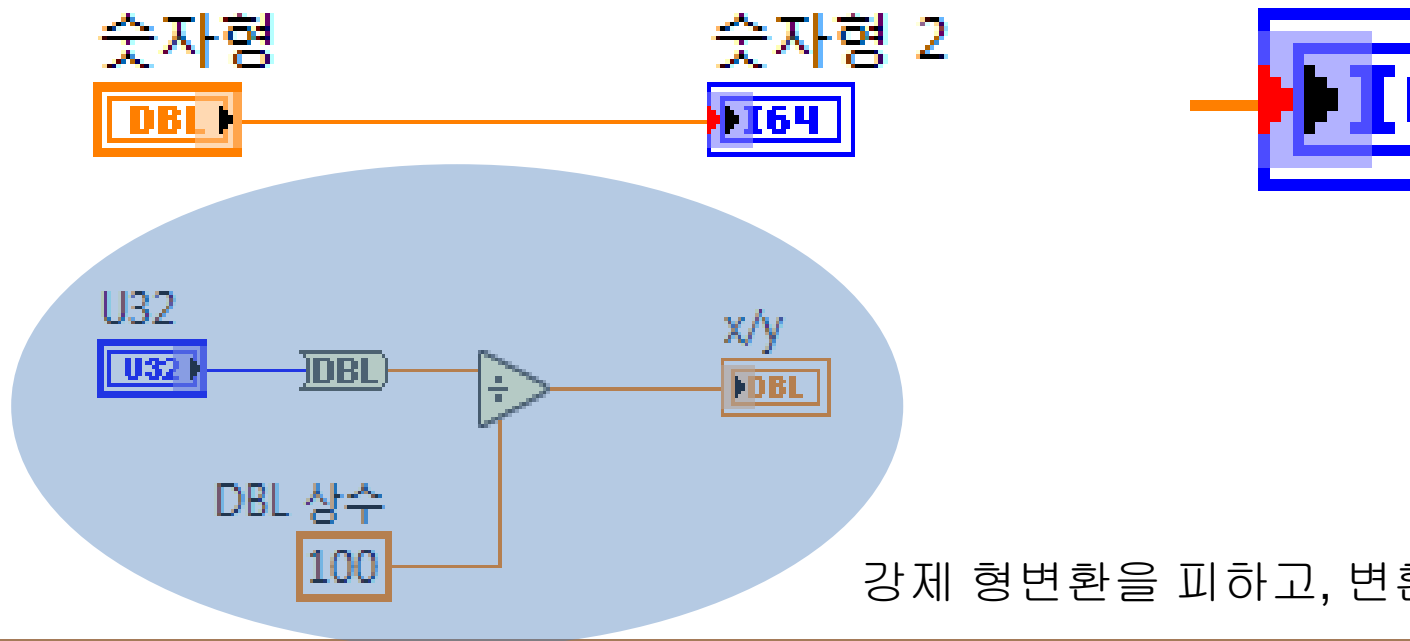
# 기수

- Integer 형
- 보이는 아이템 > 기수
  - 10진수, 16진수, 8진수 또는 2진수나 SI 표기법으로 변경
  - 문자 (d, x, o, b, p)로 표기



# 숫자형의 강제 형 변환

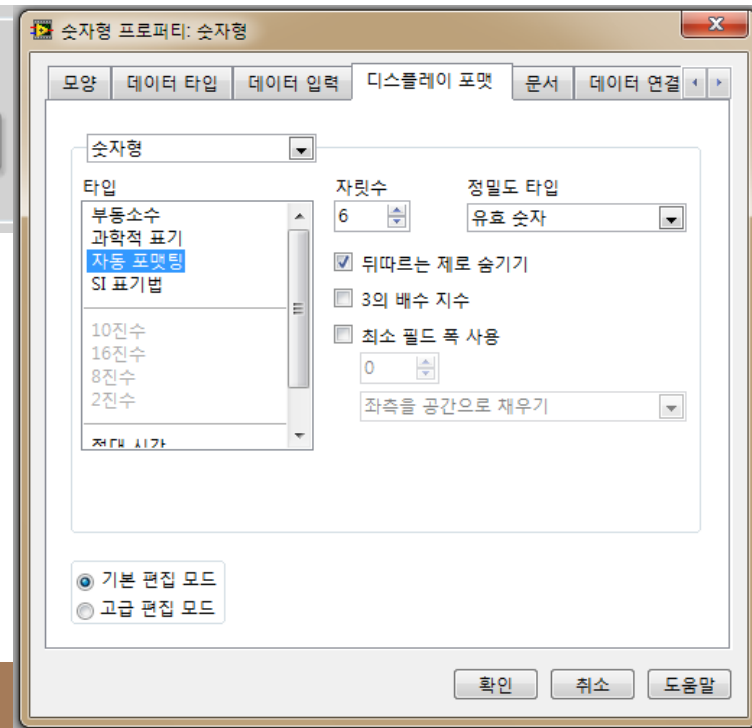
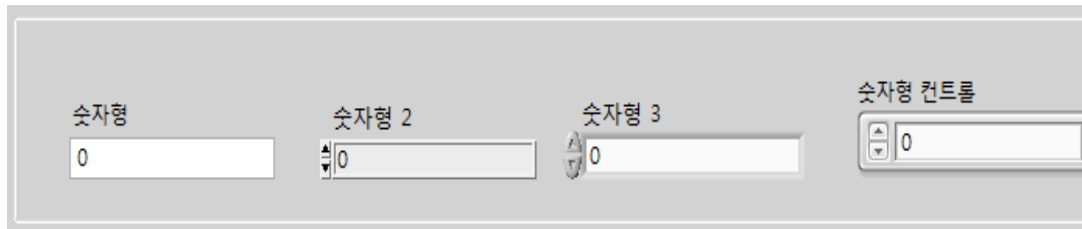
- LabVIEW Memory Manger
- 강제 형변환
- 강제 형변환 점



강제 형변환을 피하고, 변환함수를 사용한다.

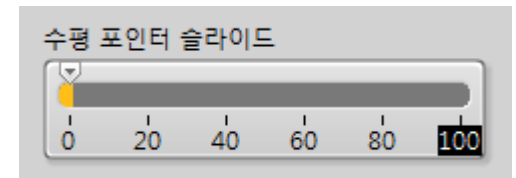
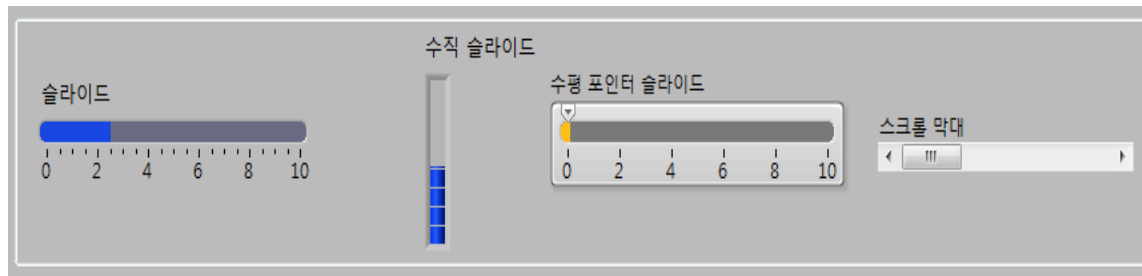
# 숫자형 컨트롤과 인디케이터

- 수평 방향으로 크기 조절할 수 있음.
- 디스플레이 포맷



# 슬라이드

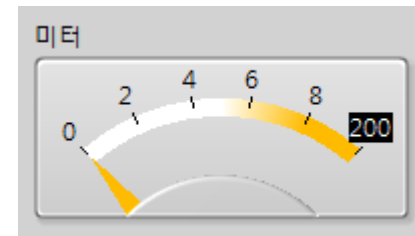
- 스케일이 있는 숫자형 객체
- 수직 슬라이드, 수평 슬라이드, 탱크, 온도계 등
- 스케일은 슬라이드의 눈금, 마크, 선형 스케일, 로그 스케일 등을 의미



- 디지털 디스플레이 보이기

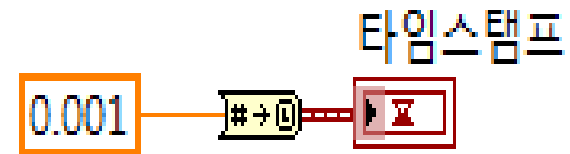
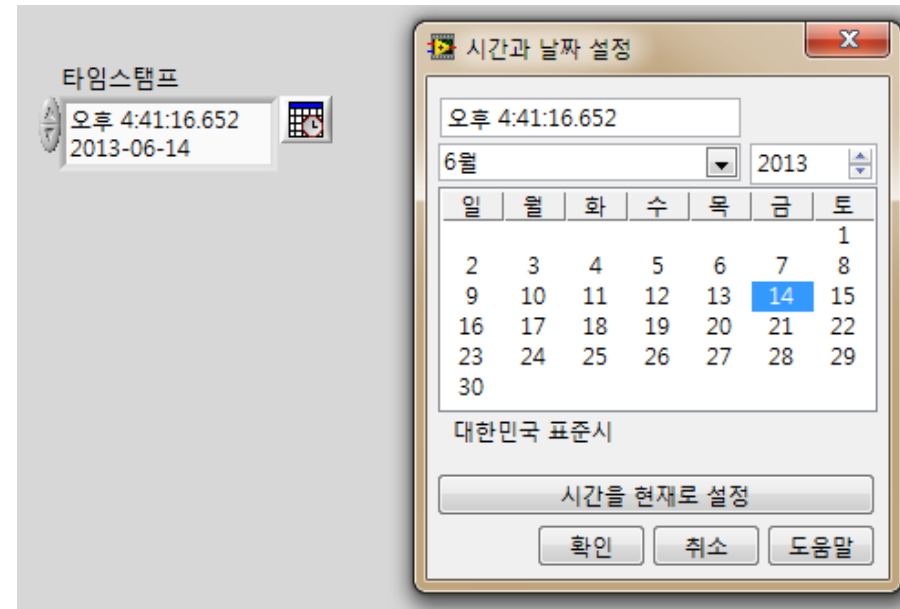
# 회전식 컨트롤과 인티케이터

- 회전식은 스케일이 있는 숫자형 객체.
- 노브, 다이얼, 미터, 게이지 등.



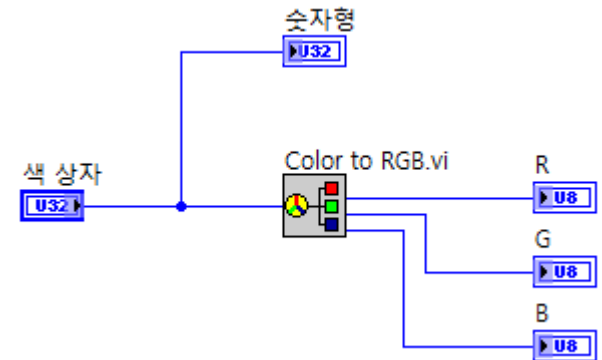
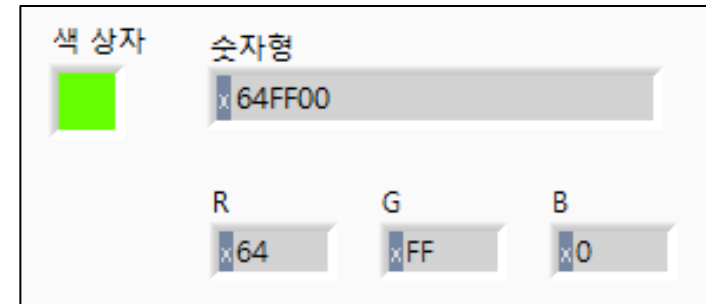
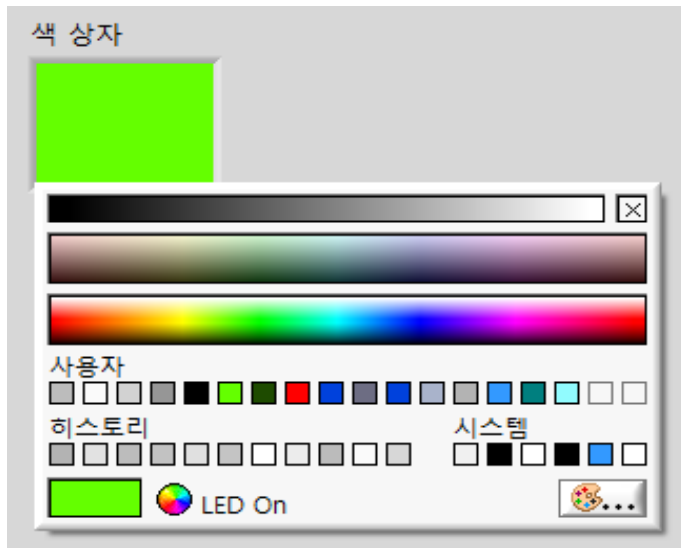
# 타임스탬프 컨트롤과 인디케이터

- UTC기준으로 “오전 00:00:00.000, 1904-01-01”을 0으로 하여 **0.001**씩 증가하는 숫자에 대응



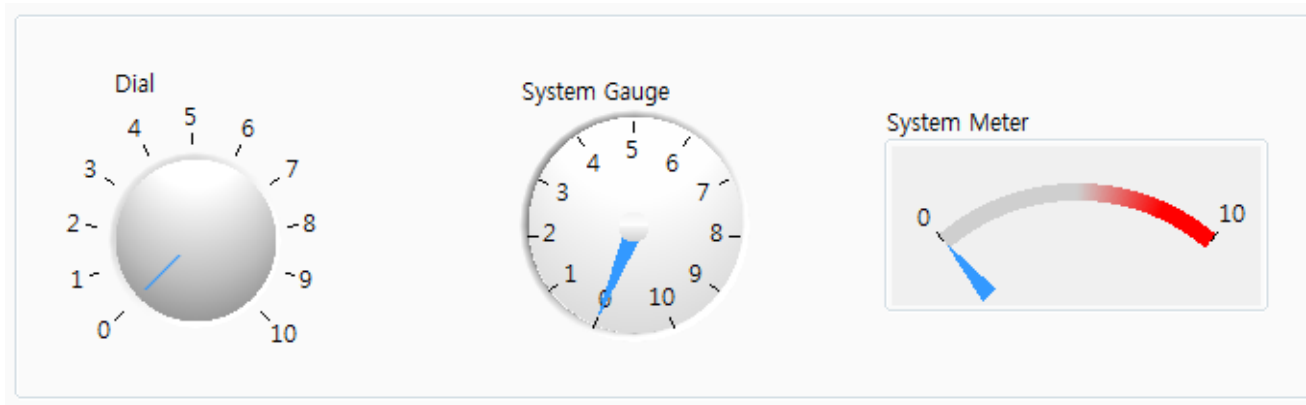
# 색 상자

- 색 값은 **RRGGBB**의 형으로 16진수의 숫자로 표시
  - 처음 두 개의 자릿수는 **빨간색**을 제어
  - 그 다음 두 개의 자릿수는 **녹색**을 제어
  - 마지막 두 자리는 **파란색**을 제어



# 숫자형 아이콘 추가

- [www.ni.com](http://www.ni.com)에서 “System Controls”로 검색하여 최신 아이콘들을 추가로 다운로드 받아서 사용할 수 있습니다.





# 숫자형 상수

- 숫자형 상수는 블록다이어그램에서 고정 데이터 값을 제공
- 원주율( $\pi$ )과 무한대( $\infty$ )



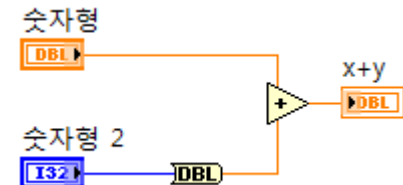
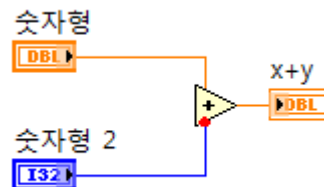
# 기호 숫자 값

- 기호 숫자는 실제 숫자는 아닙니다.
- **NaN** (숫자 아님)은 음수의 제곱근
- **Inf** (무한대)
- 부동소수 데이터 타입
- 정수와 고정 고정 소수점 수는 기호 숫자를 지원하지 않습니다.
  - $+Inf = 32,767$  (16비트 부호 있는 정수)
  - $-Inf = -32,768$
  - $NaN = 32,767$  (그 데이터 타입의 가장 큰 값)



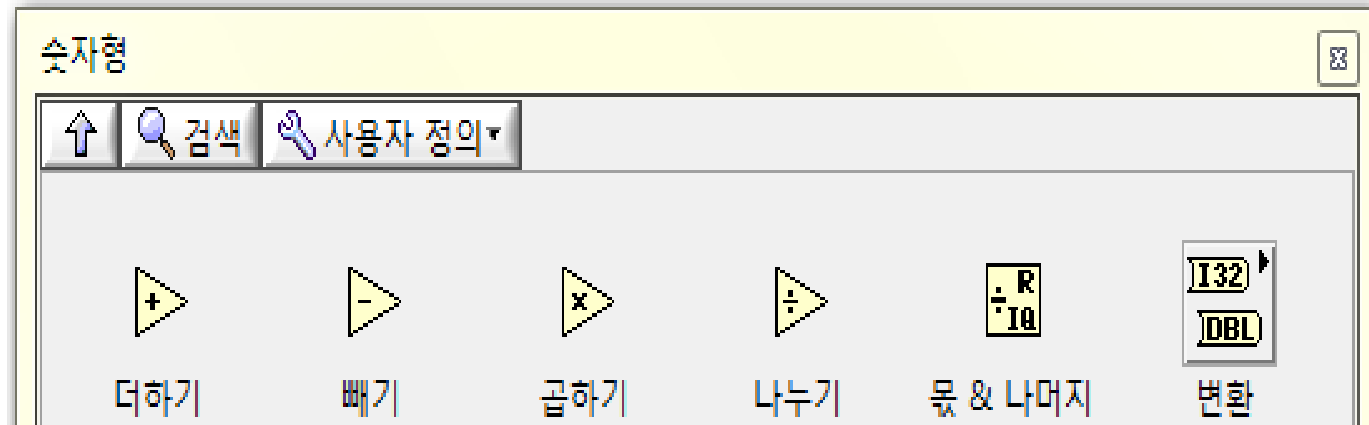
# 숫자 형 변환

- 다른 숫자 형을 연결하더라도 에러가 발생하지는 않음.
- 자동으로 형변환이 되어, 타입이 맞도록 조정.
- LabVIEW 메모리 관리자.
- 빨간색 강제 변환점으로 형변환 되었음을 알려줌.



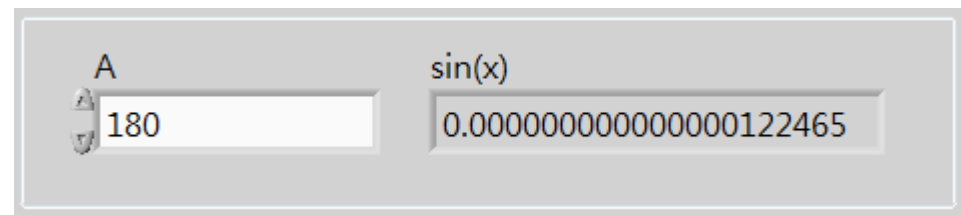
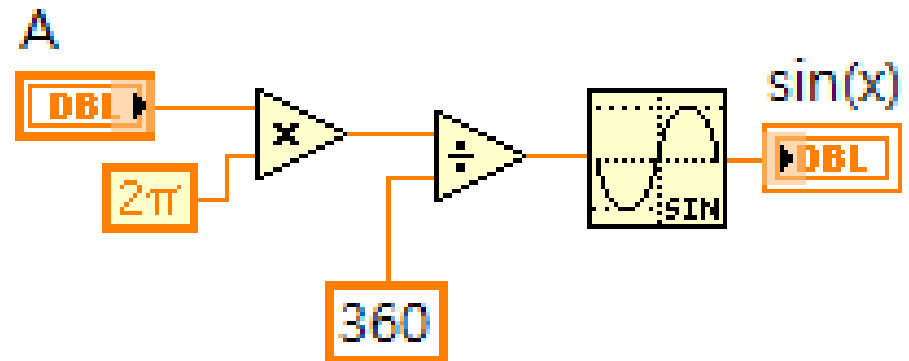
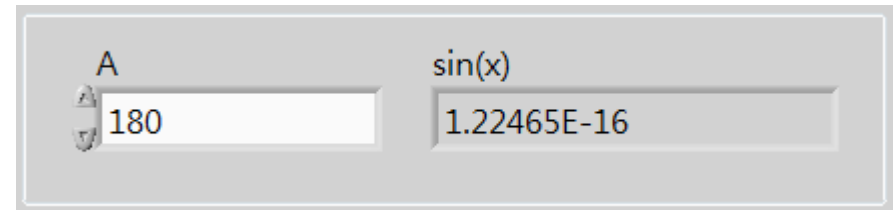
# 더하기, 빼기, 곱하기, 나누기, 몫 & 나머지

- 숫자형 스칼라 값이나 배열을 입력 받음.
- 동일한 형을 입력 받아서 연산을 수행함.
- 다른 숫자 형이 입력될 경우에는 강제 형변환.



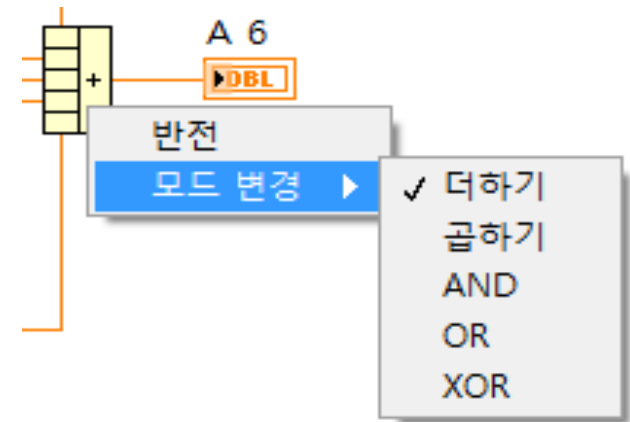
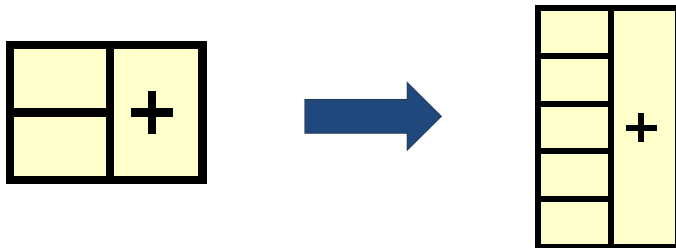
# 실습 2-1: 숫자형 데이터 타입

- 숫자형 입력
- 곱하기와 나누기
- 사인 함수



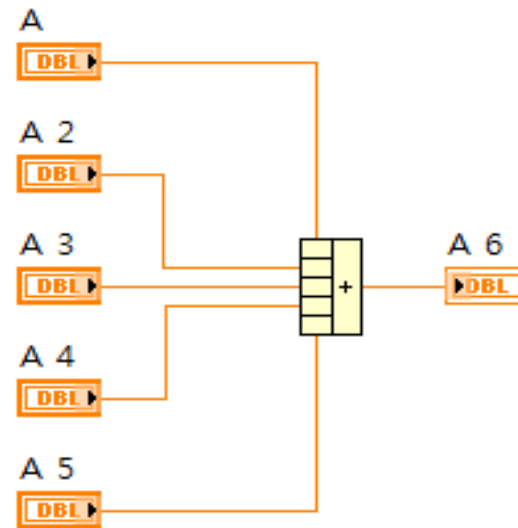
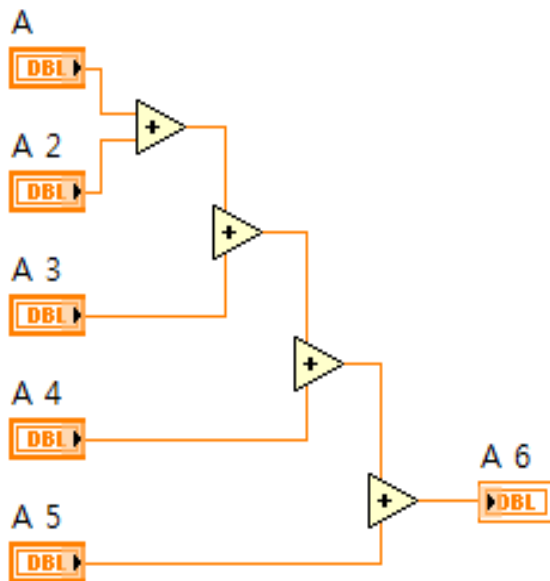
# 복합 연산

- 더하기, 곱하기 연산
- 복합 연산은 입력의 개수를 변경



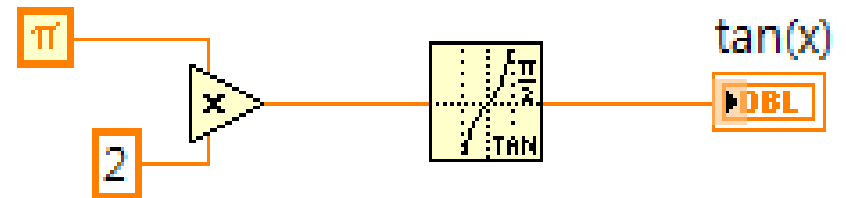
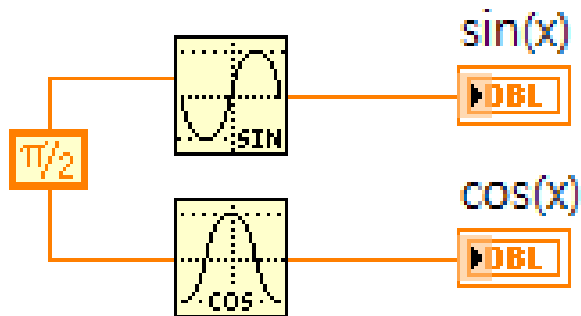
## 실습 2-2: 여러 숫자 더하기

$$A + A2 + A3 + A4 + A5 = A6$$



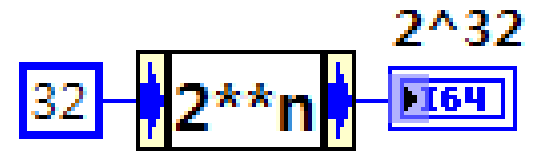
# 삼각 함수

- 함수 > 수학 > 기본 & 특수 함수 > 삼각 함수





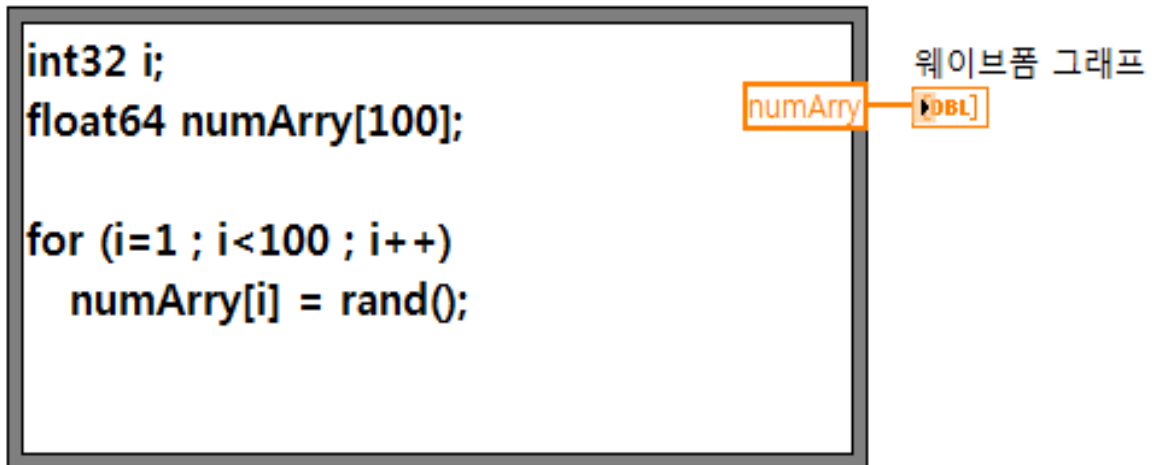
# 식 노드



- 한 개의 숫자형 입력, 한 개의 출력
- 출력 터미널은 입력 터미널과 같은 데이터 타입임
- 입력의 데이터 타입
  - 숫자, 숫자의 배열, 클러스터
  - 복소수는 지원하지 않음.
- abs, acos, acosh, asin, asinh, atan, atanh, ceil, cos, cosh, cot, csc, exp, expm1, floor, getexp, getman, int, intrz, ln, lnp1, log, log2, max, min, mod, rand, rem, sec, sign, sin, sinc, sinh, sizeofDim, sqrt, tan, tanh 등을 지원 (C 언어에서의 의미와 동일)

# 수식 노드

- C 프로그래밍과 유사한 텍스트 기반의 구문
- 기존 코드를 수식 노드로 복사하여 붙여넣을 수 있음
- 끝 지정은 세미콜론 (;)
- 입력 터널 (필수 입력)
- 출력 터널



# 수식 노드

- 입력 변수는 왼쪽에 배치하고, 출력 변수는 오른쪽에 배치
- 입력 변수는 수식 노드 내부에서 선언할 수 없음.
  - 필수 입력이므로, 반드시 입력 연결
- 출력 변수는 수식 노드 내부에서 선언할 수 있음.
- 변수 이름은 소문자와 대문자를 구분.
  - 입력 변수명과 같은 출력 변수 명을 사용할 수 있음.
- 변수와 수식에는 개수의 제한이 없음.

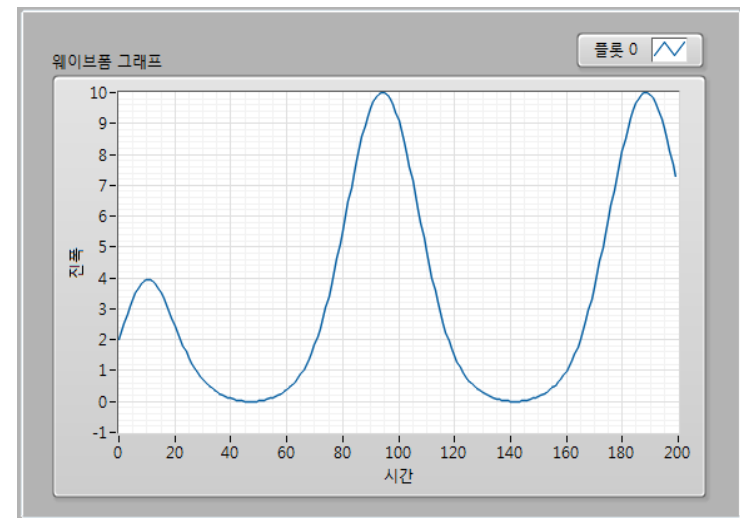
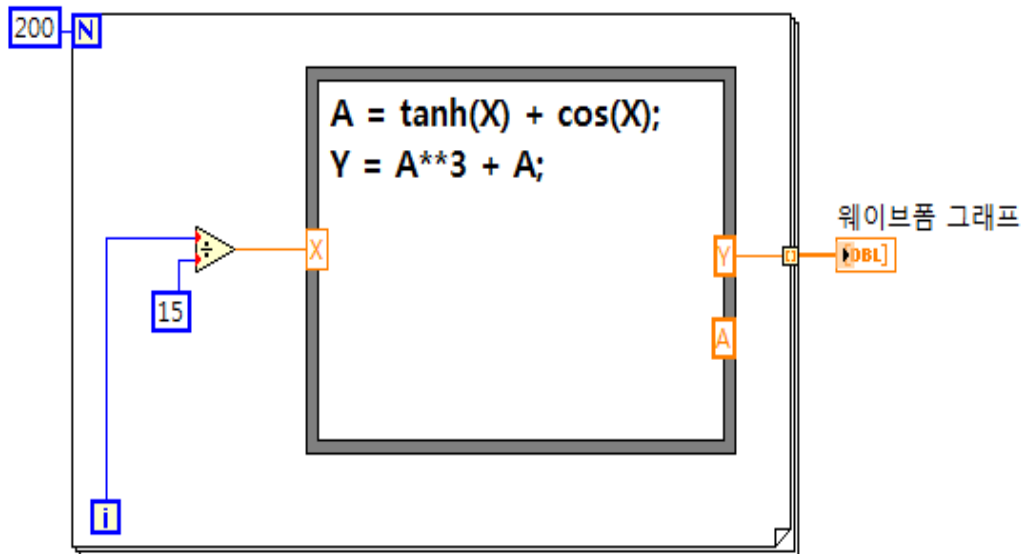
# 스크립트 노드

- 스크립트 노드를 사용하여 수학 스크립트를 실행.
- MATLAB® 스크립트
- MathScript 스크립트
- 관련 소프트웨어가 설치되어 있어야만 이들 스크립트 노드들이 활성화되고 사용이 가능함.

# 실습 2-3: 수식 노드 사용하기

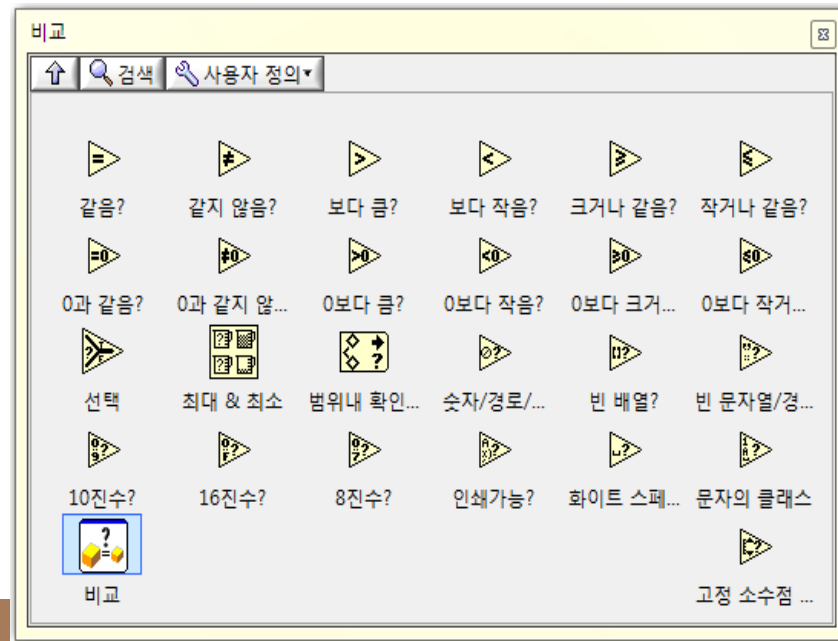
$$A = \tanh(X) + \cos(X);$$

$$Y = A^{**}3 + A;$$



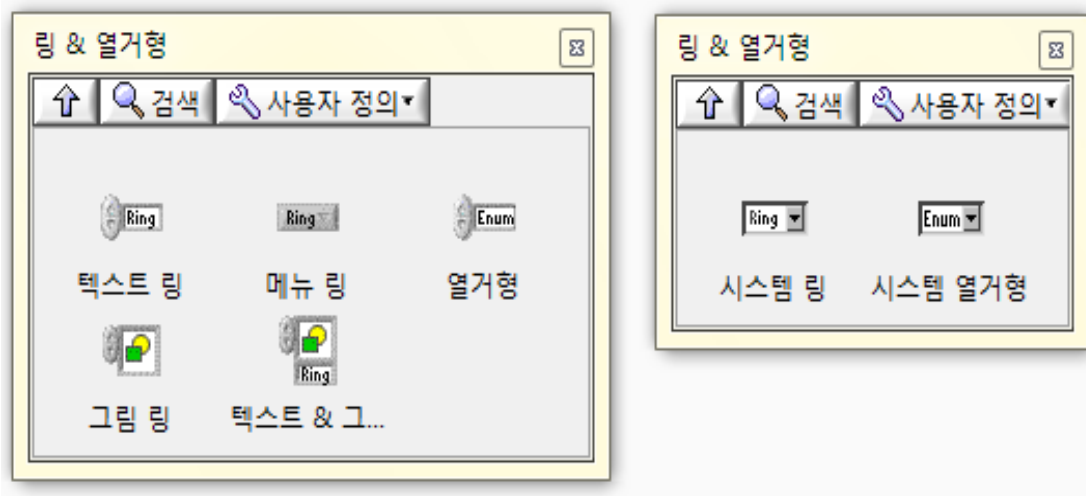
# 비교 함수

- 숫자형 데이터 타입들을 서로 비교함.
- 문자열 데이터 타입들을 서로 비교함.
- 비교 함수의 결과 값은 불리언.



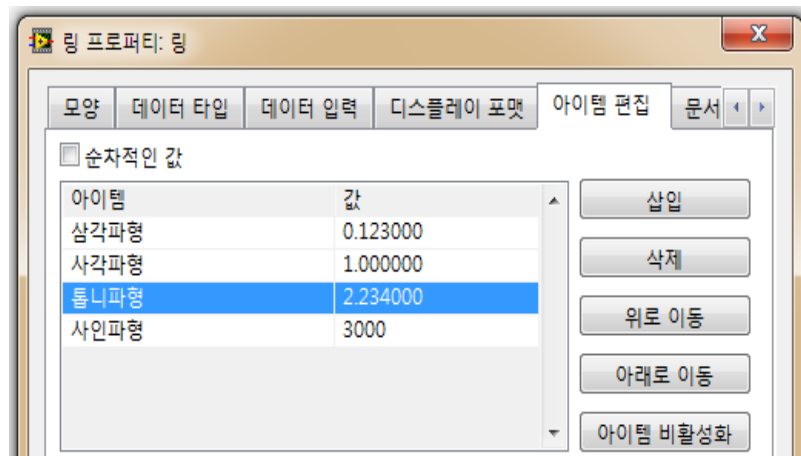
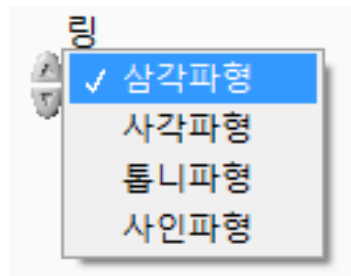
# 링 & 열거형

- 풀다운 메뉴에서 아이템을 선택할 수 있는 문자열의 리스트
- 링과 열거형의 사용 방법은 비슷하지만 많은 차이점이 있음



# 링

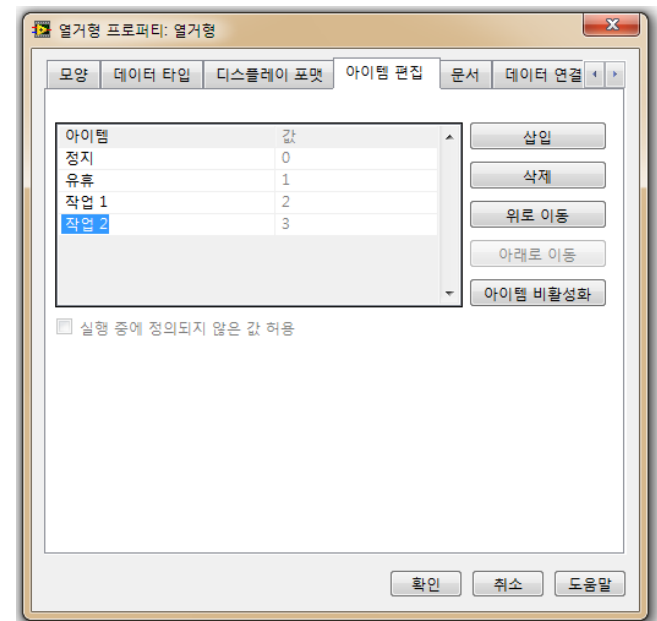
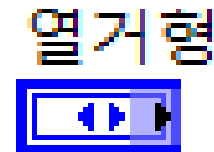
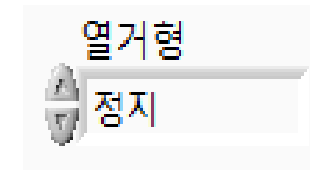
- 숫자 값을 문자열 또는 그림에 연관시키는 숫자형 객체
- 링 컨트롤은 사용자가 아이템을 선택할 수 있는 풀다운 메뉴
- “순차적인 값”은 0부터 시작하여 1씩 증가
- 순차적인 값을 선택 해제하면 임의의 숫자로 지정할 수 있음. (스케일)
- 텍스트 링, 메뉴 링, 그림 링, 텍스트 & 그림 링, 시스템 링 등.





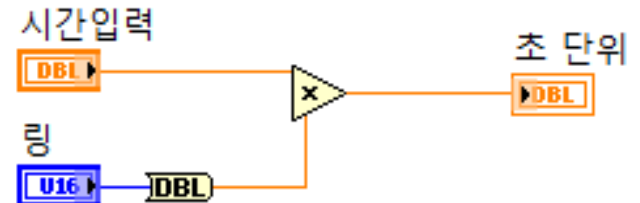
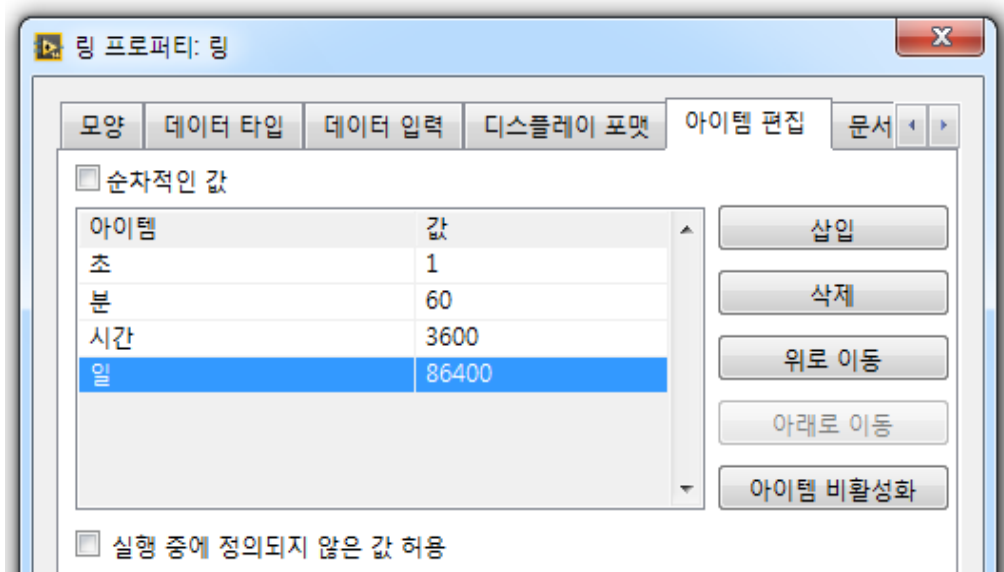
# 열거형

- 열거형 만의 독특한 터미널로 표현됨.
- 열거형은 U32, U16, U8만 지원.
- 순차적인 값 선택 항목이 없음.
- 케이스 구조의 선택자로 사용됨.

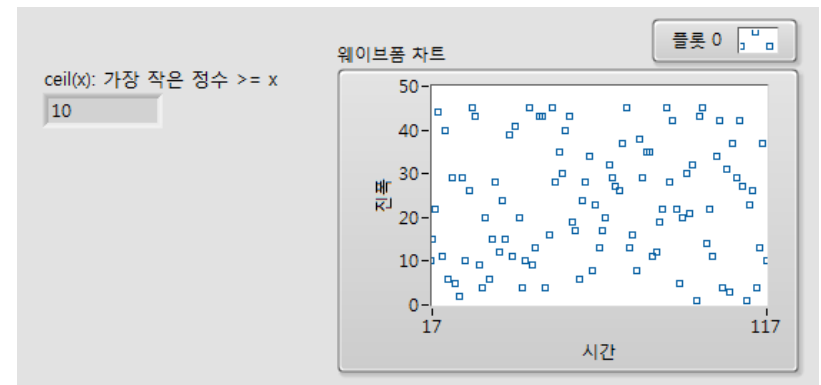
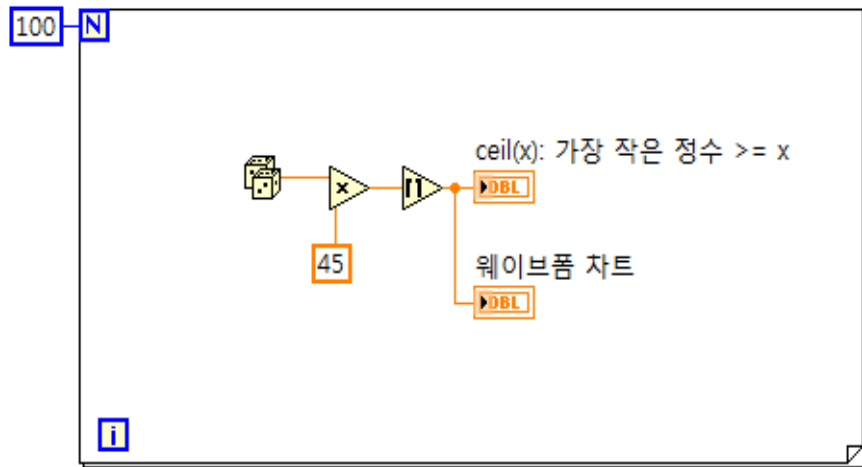


# 실습 2-4: 텍스트 링

아이템	초	분	시간	하루
값	1	60	3600	86400

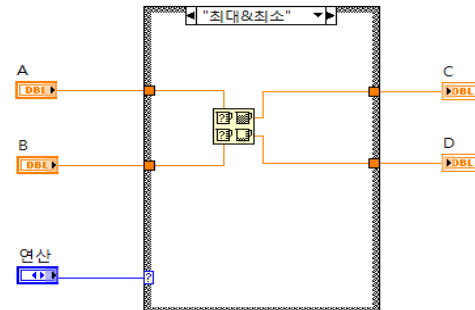
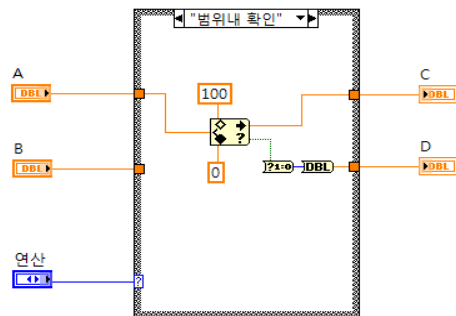
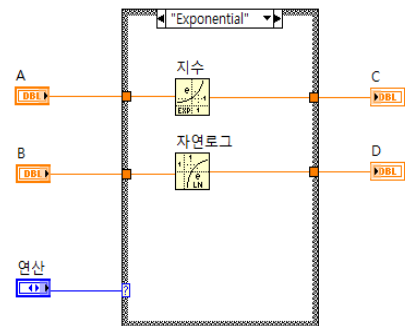
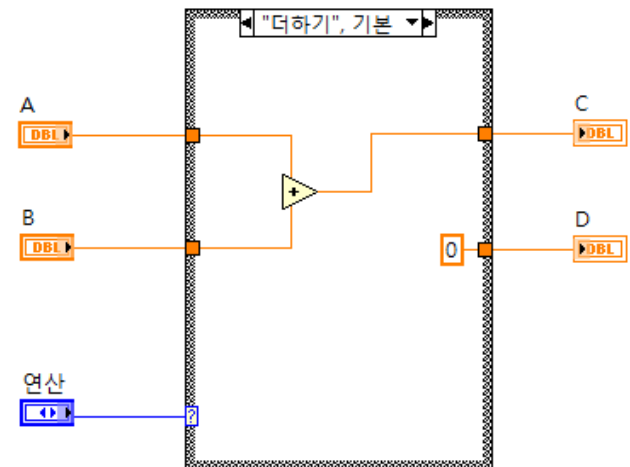
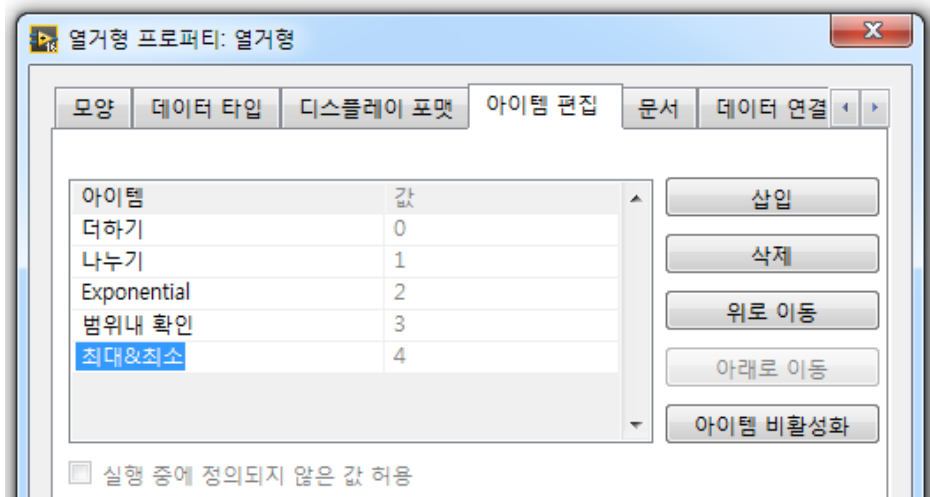


# 실습 2-5: 난수 발생 45



# 실습 2-6: 다양한 숫자형 연산

아이템	더하기	나누기	Exponential	범위내 확인	최대&최소
값	0	1	2	3	4



---

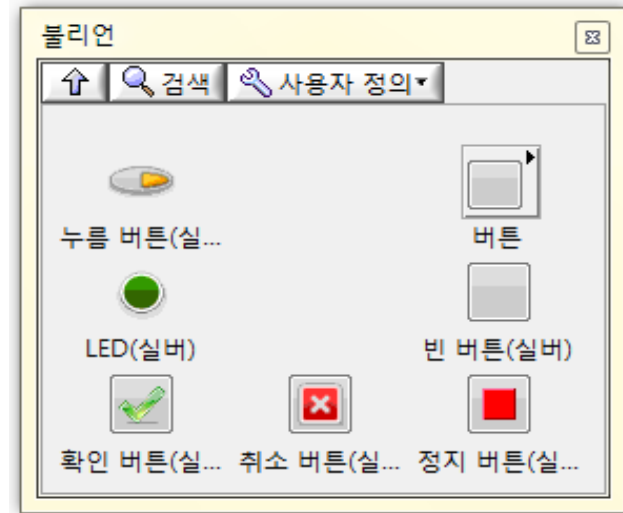
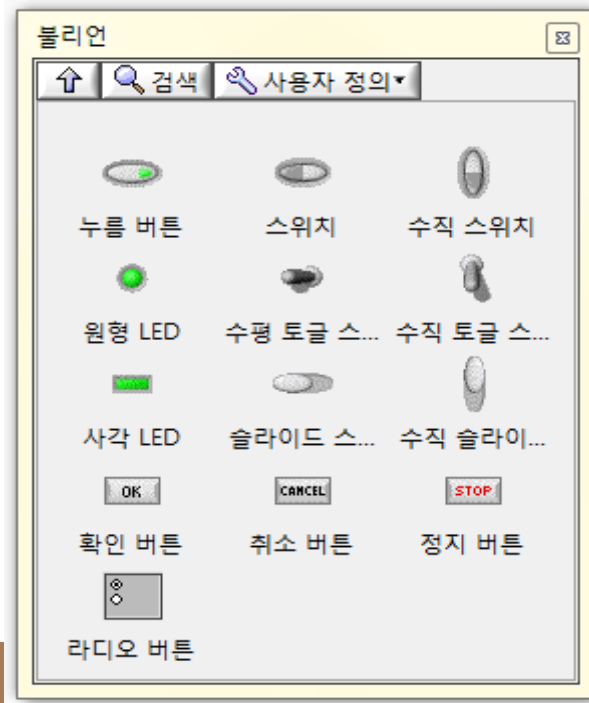
## 2. 불리언 데이터 타입과 함수

불리언 데이터 타입

불리언 함수

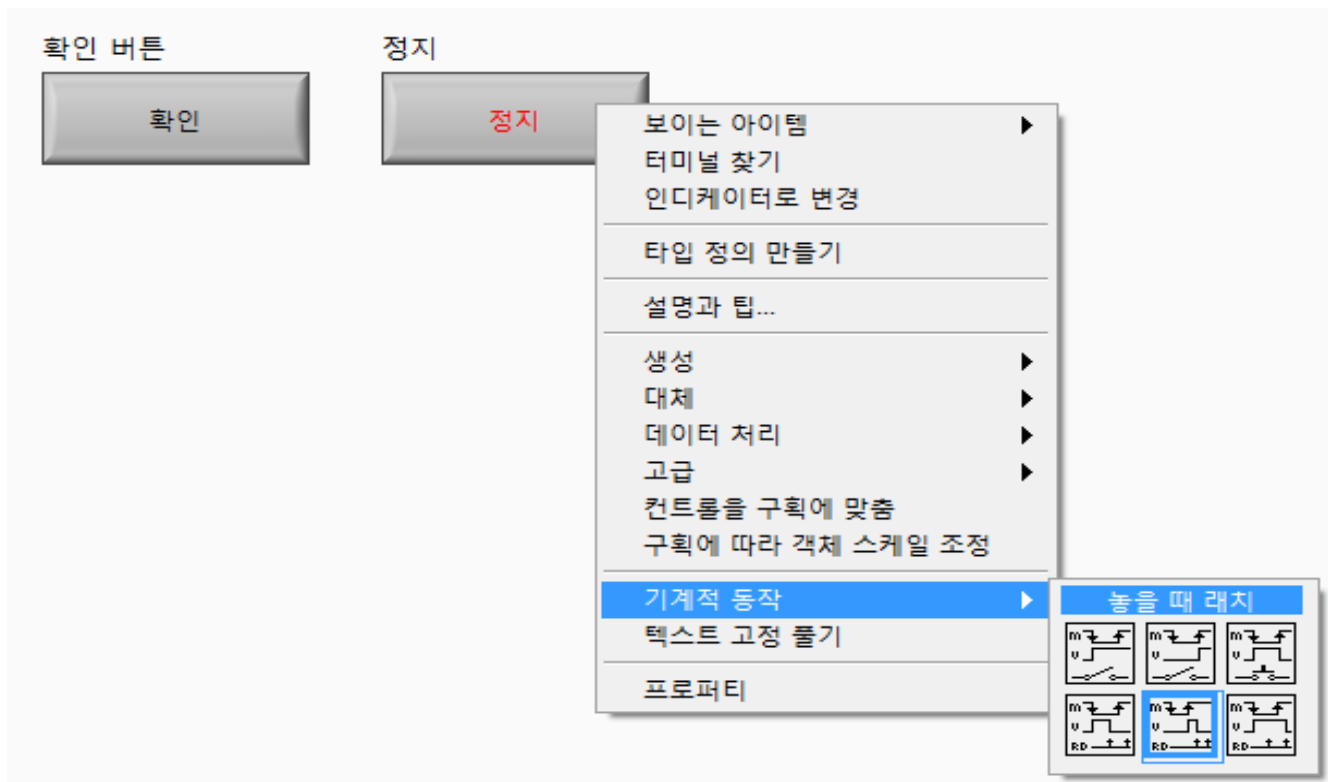
# 블리언

- 버튼이나 스위치, LED
- 참/거짓
- 객체의 크기를 키우거나 줄일 수 있음.



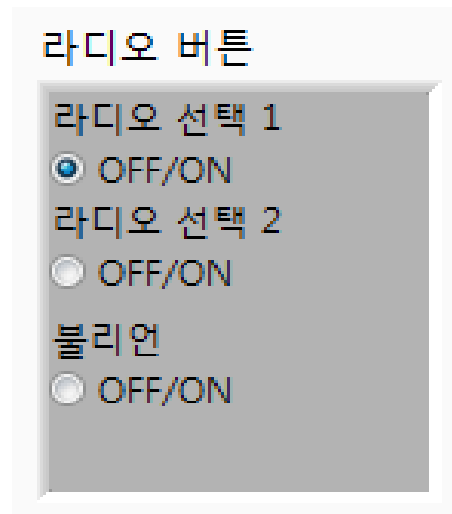
# 기계적 동작

- 6가지의 기계적 동작
- 정지 버튼과 확인 버튼의 기본 설정은 **놓을 때 래치**



# 라디오 버튼

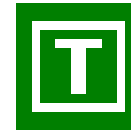
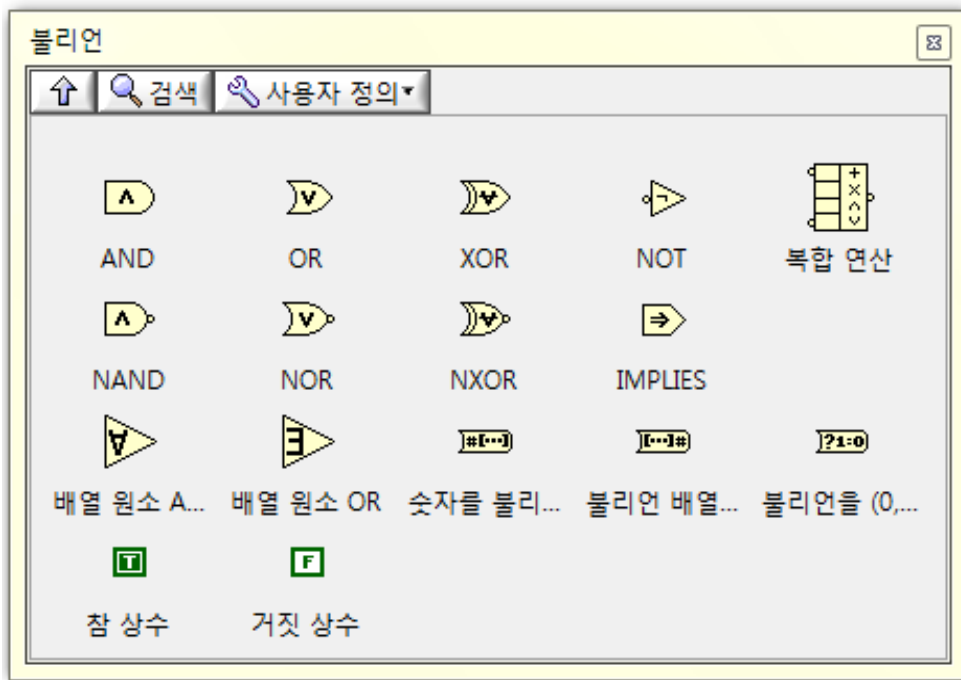
- 한 번에 하나만 선택할 수 있는 아이템 리스트
  - 동시에 두 개의 버튼을 선택할 수는 없습니다.
- 열거형 타입임 → 케이스 구조의 선택자로 사용됨.
- 컨트롤 > 시스템 > 불리언 팔레트의 시스템 라디오 버튼





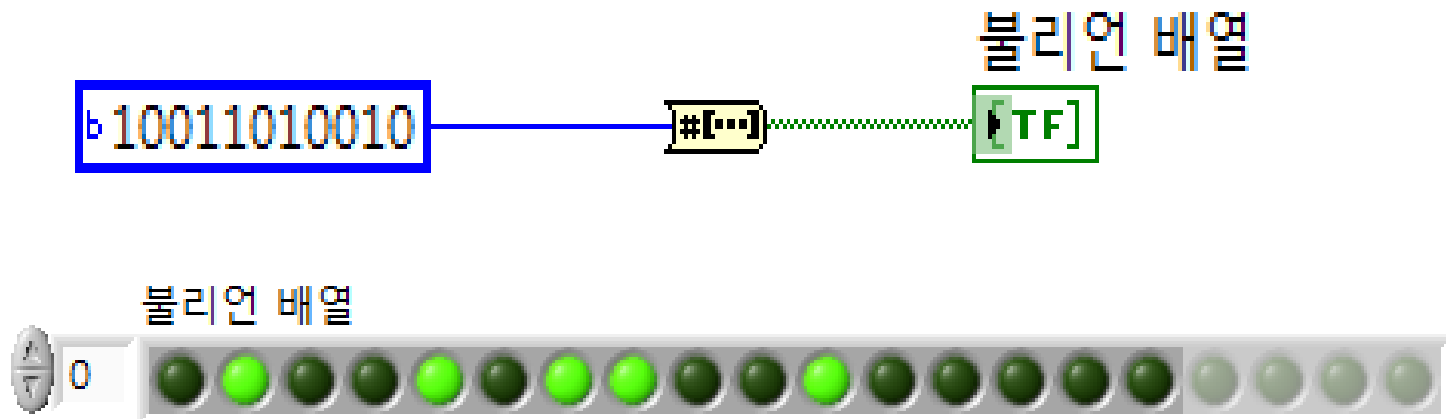
# 불리언 함수

- AND, OR, XOR, NOT 등의 논리 연산
- 불리언 상수: 참 상수와 거짓 상수



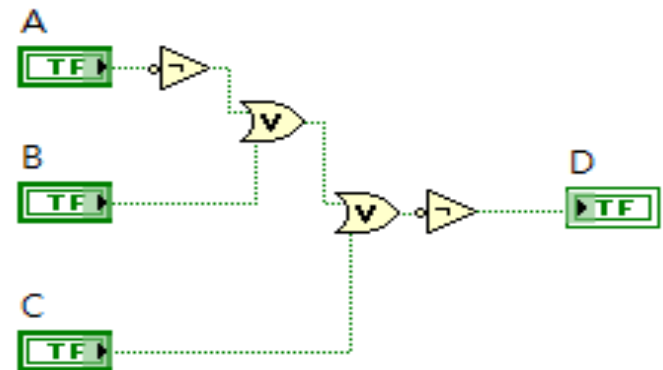
# Integer 숫자형 = 불리언 배열

- 불리언 배열: 8bits, 16bits, 32bits



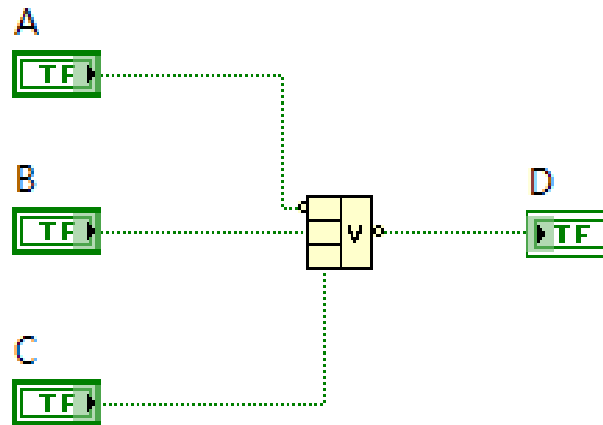
## 실습 2-7: 복합 불리언 연산

$$\text{NOT } ((\text{NOT } A) \text{ OR } B \text{ OR } C) = D$$



# 실습 2-6: 복합 불리언 연산 (연속)

$$\text{NOT} ((\text{NOT } A) \text{ OR } B \text{ OR } C) = D$$



---

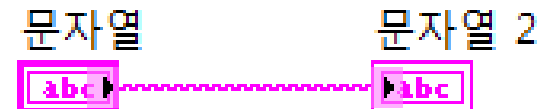
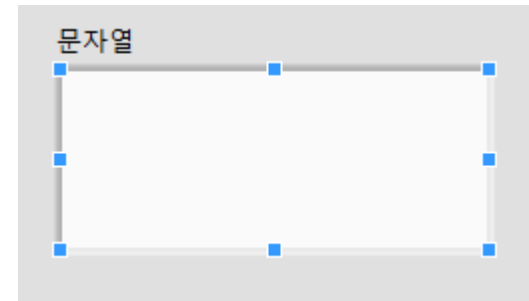
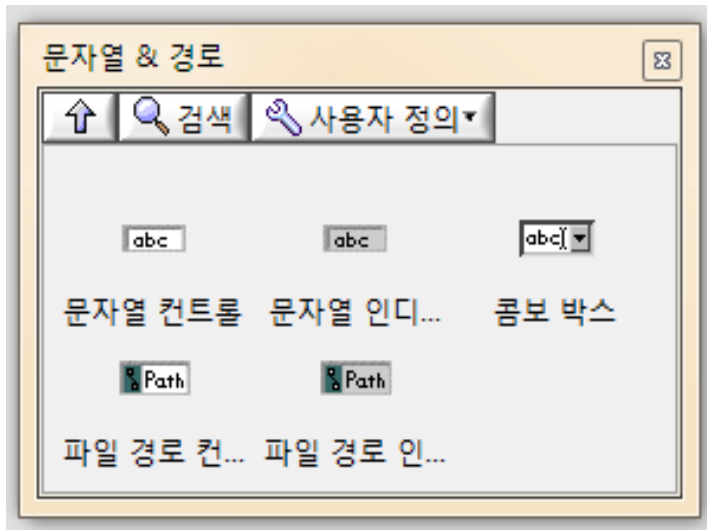
# 3. 문자열 데이터 타입과 함수

문자열 데이터 타입

문자열 함수

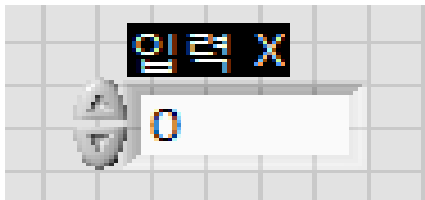
# 문자열

- 문자열은 문자(Character)의 나열입니다.
- LabVIEW 문자열은 유니코드를 사용하여 모든 문자와 기호를 지원합니다.
- 유니코드는 전 세계의 모든 문자를 컴퓨터에서 일관되게 표현하고 다룰 수 있도록 설계된 표준 (ISO 10646)입니다.



# 문자열 사용 예

- 객체의 라벨, 캡션, 그래프나 차트의 X 스케일, Y 스케일, 플롯 이름



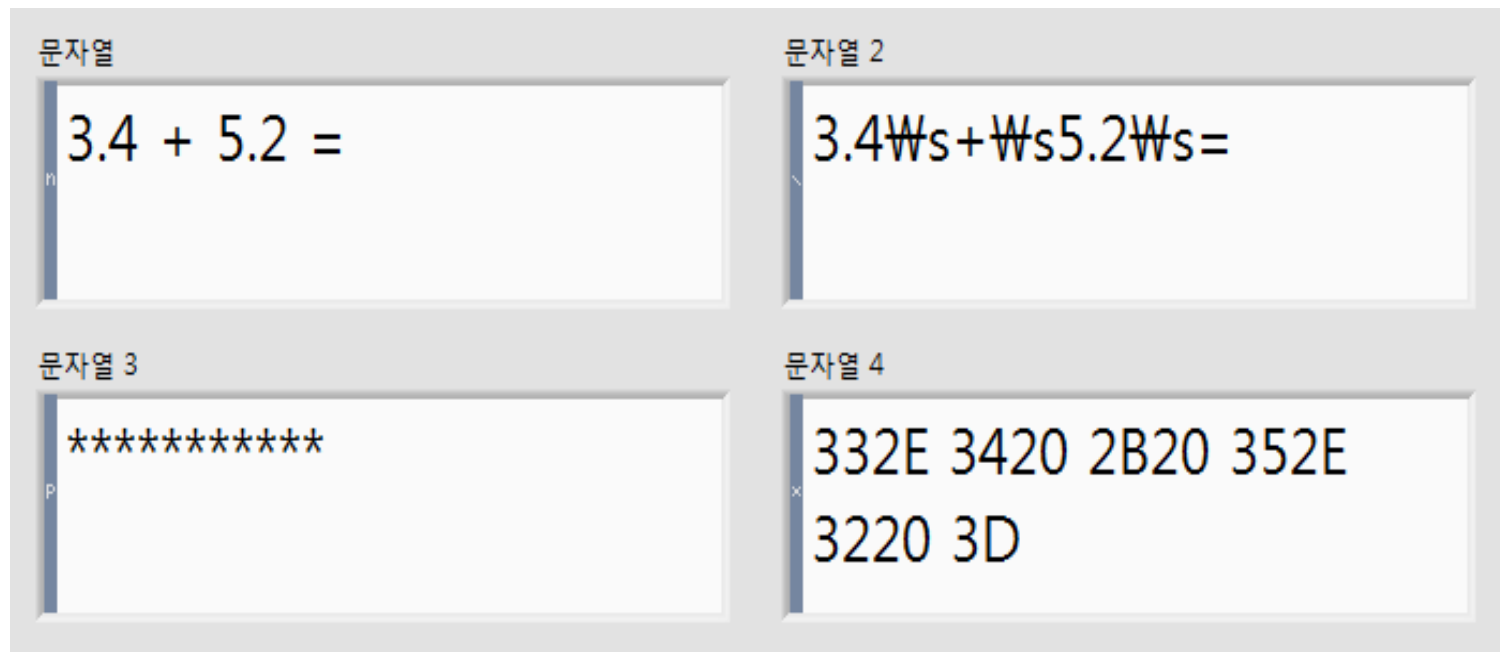
# ASCII 코드

- 컴퓨터와 인스트루먼트, 인스트루먼트와 인스트루먼트 사이의 데이터 통신에서는 128개 코드로 구성된 **ASCII**를 주로 사용합니다.
- **ASCII**는 미국 표준화 협회가 제정한 정보 교환용 표준 코드입니다.
- **ASCII**에는 영어 대문자, 소문자, 문장 부호, 기본 숫자 10개, 그리고 컨트롤 코드가 포함됩니다.
- **ASCII**는 7bits(0부터 127까지)로 구성되어 128개 문자를 나타낼 수 있고, 1 parity bit를 둬으로써 데이터를 전송에서 발생할 수 있는 오류 검사가 가능합니다.



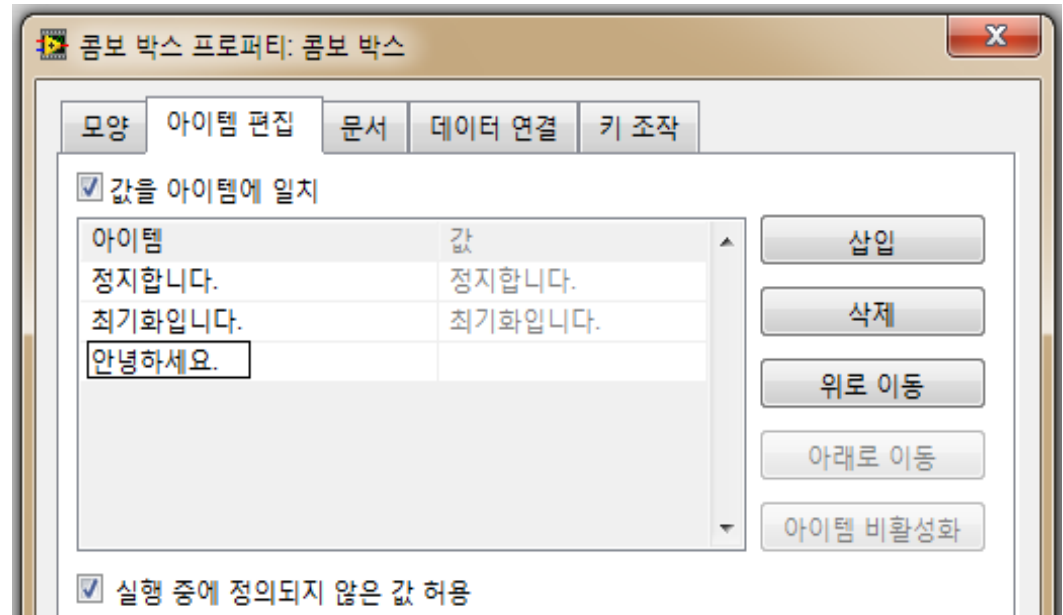
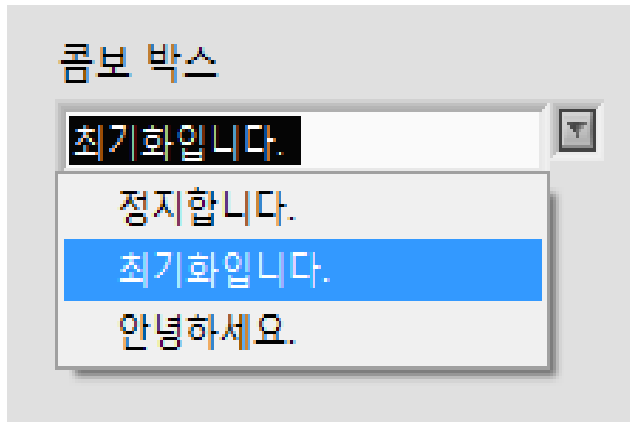
# 디스플레이 스타일

- 보이는 아이템 > 디스플레이 스타일
- 일반(n), \ 코드, 암호(p), 16진수(x) 디스플레이 방식을 바꿔서 선택해 줄 수 있습니다.



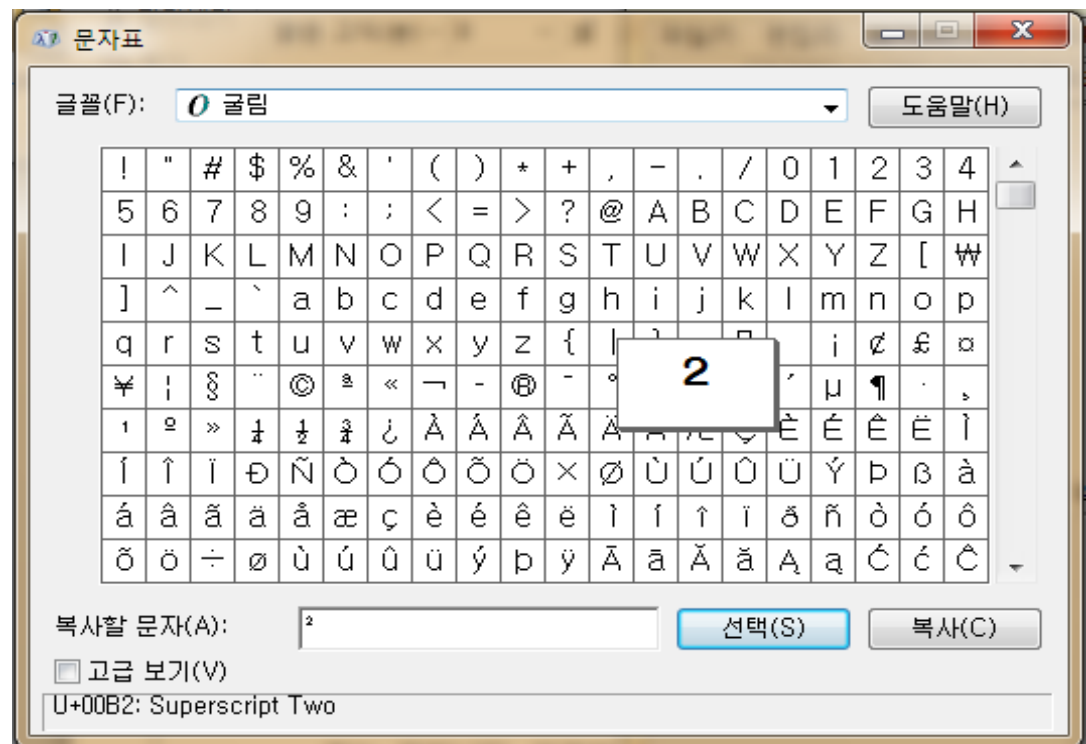
# 콤보 박스

- 풀다운 메뉴에서 선택 가능한 문자열 리스트



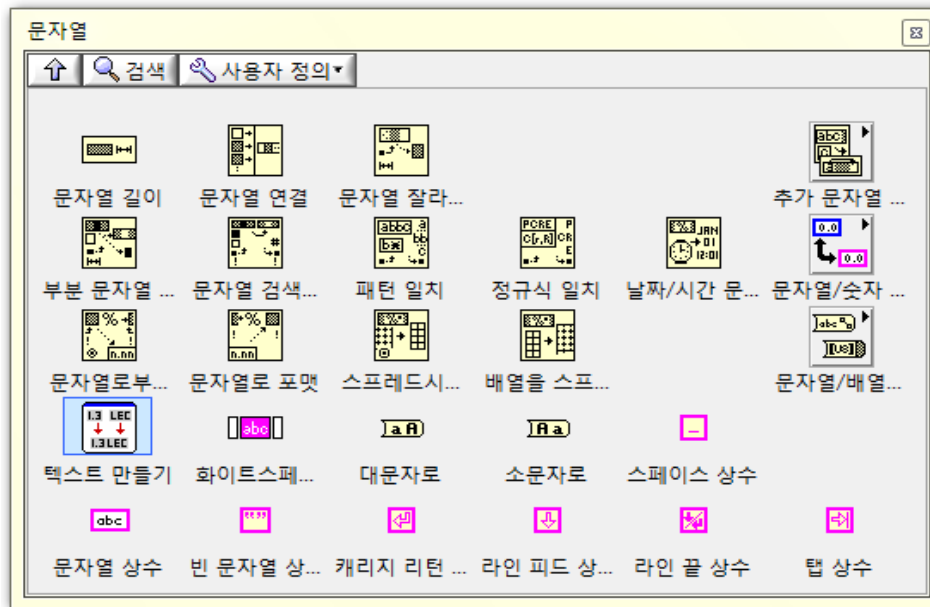
# 객체의 라벨

- 문자열 컨트롤이나 객체의 라벨 등을 키보드로 직접 입력 가능하지만, 키보드 입력만으로 원하는 문자를 모두 입력할 수는 없습니다.
- 문자표 (Character Map) 사용
- 코드 번호 사용



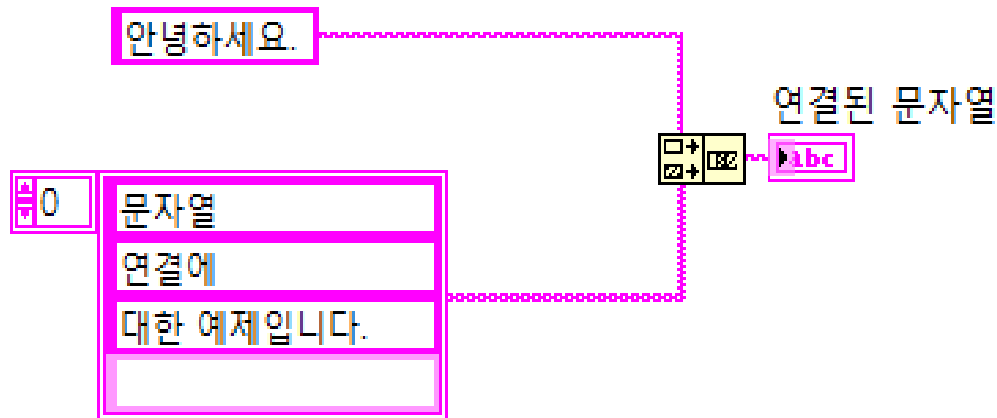
# 문자열 함수

1. 새 문자열 만들기
2. 문자열에서 특정 문자(패턴) 찾기/검색
3. 숫자형/문자열 변환하기 (포맷하기)



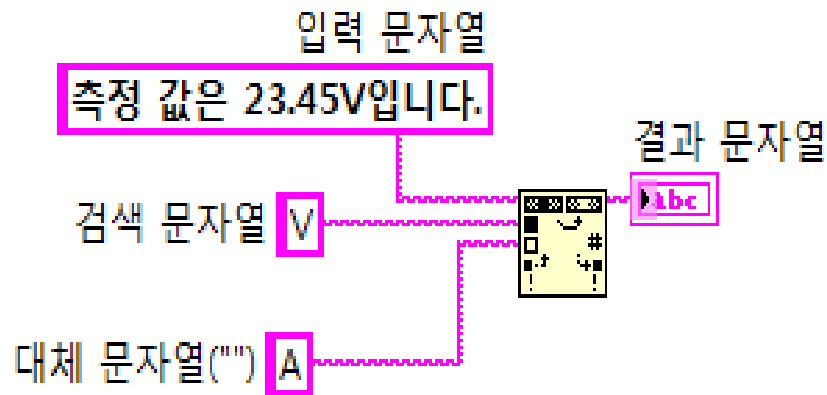
# 새 문자열 만들기

- 문자열 연결
- 문자열 잘라내기
- 부분 문자열 대체 함수



# 문자열에서 특정 문자(패턴) 찾기/검색

- 문자열 검색과 대체
- 패턴 일치
- 정규식 일치

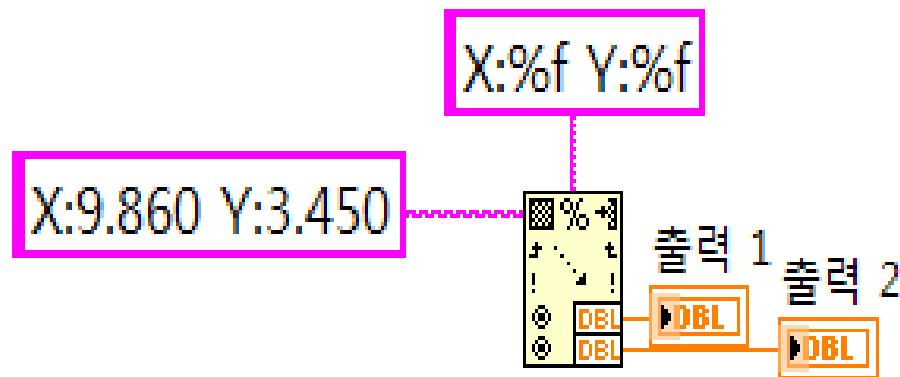


결과 문자열

측정 값은 23.45A입니다.

# 숫자형/문자열 포맷하기

- 문자열로부터 스캔
- 문자열로 포맷
- 스프레드시트 문자열을 배열로
- 배열을 스프레드시트 문자열로



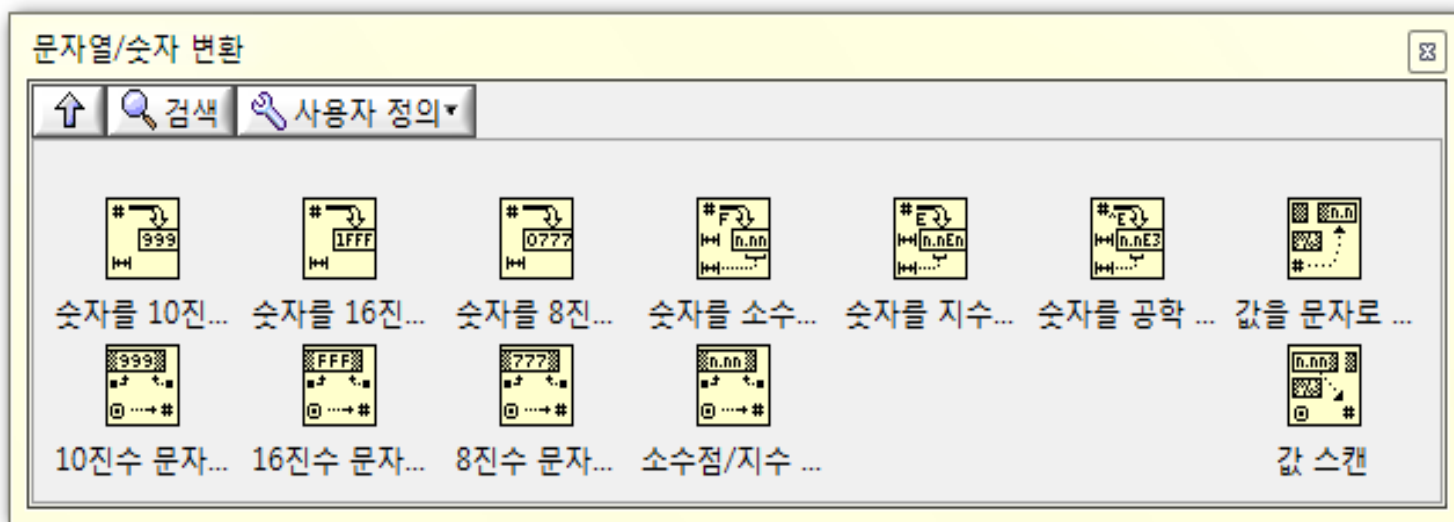
# 포맷 구문

포맷 구문	설명
%s	문자열
%d	부호 있는 10진수 정수
%u	부호 없는 10진수 정수
%b	2진수
%x	16진수
%o	8진수
%f	소수점 포맷의 부동소수(예를 들면, 12.345)
%e	과학적 표기법의 부동소수(예를 들면, 1.234E1).
.	정밀도를 나타냅니다. (예를 들면, %.4f는 소수점 네 자리로 포맷합니다.)



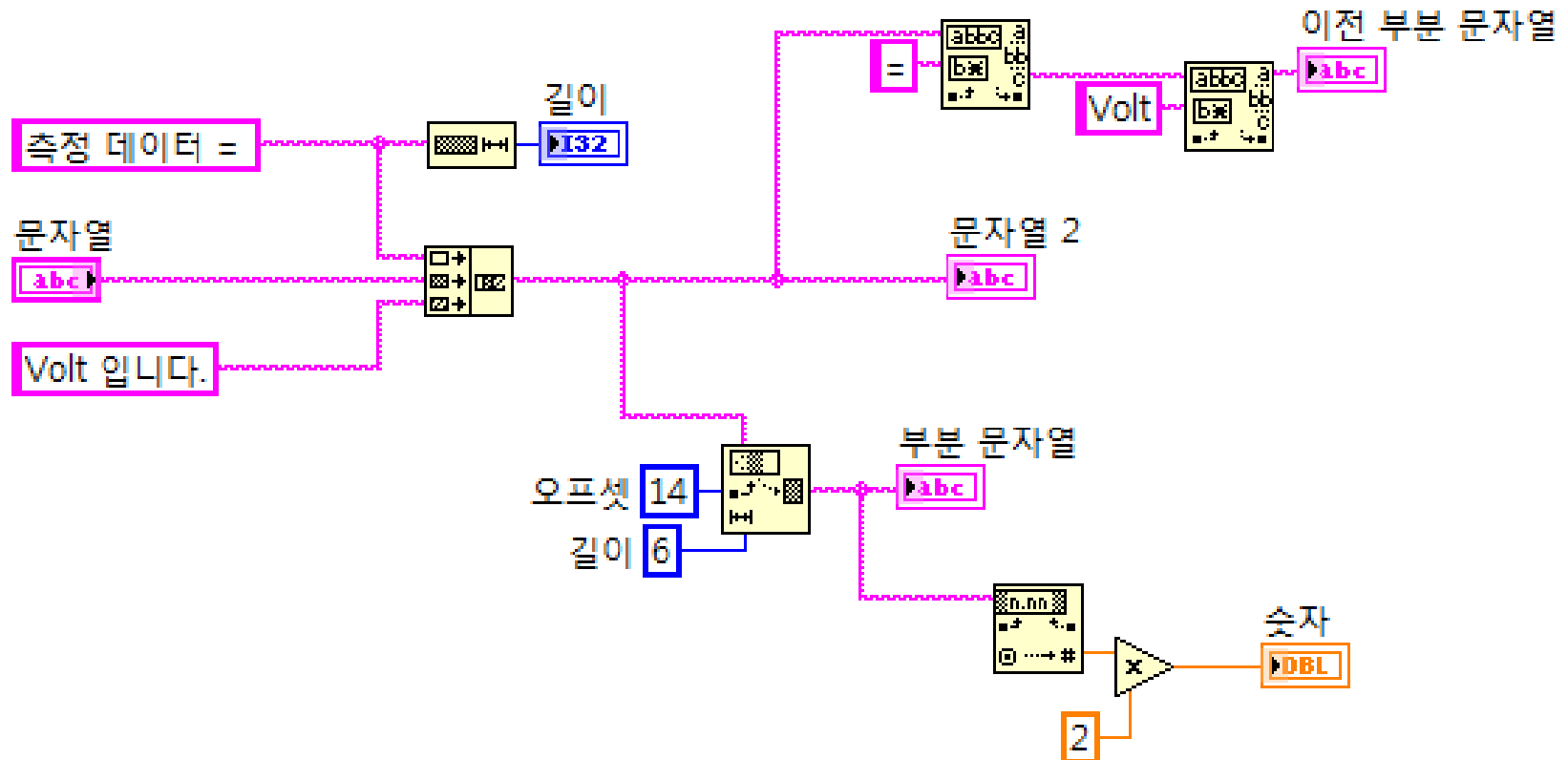
# 문자열/숫자 변환

문자열의 하위 팔레트인 문자열/숫자 변환 팔레트에는 문자열을 숫자 변환하거나 숫자를 문자열로 변환하는 함수들이 있는 팔레트입니다.

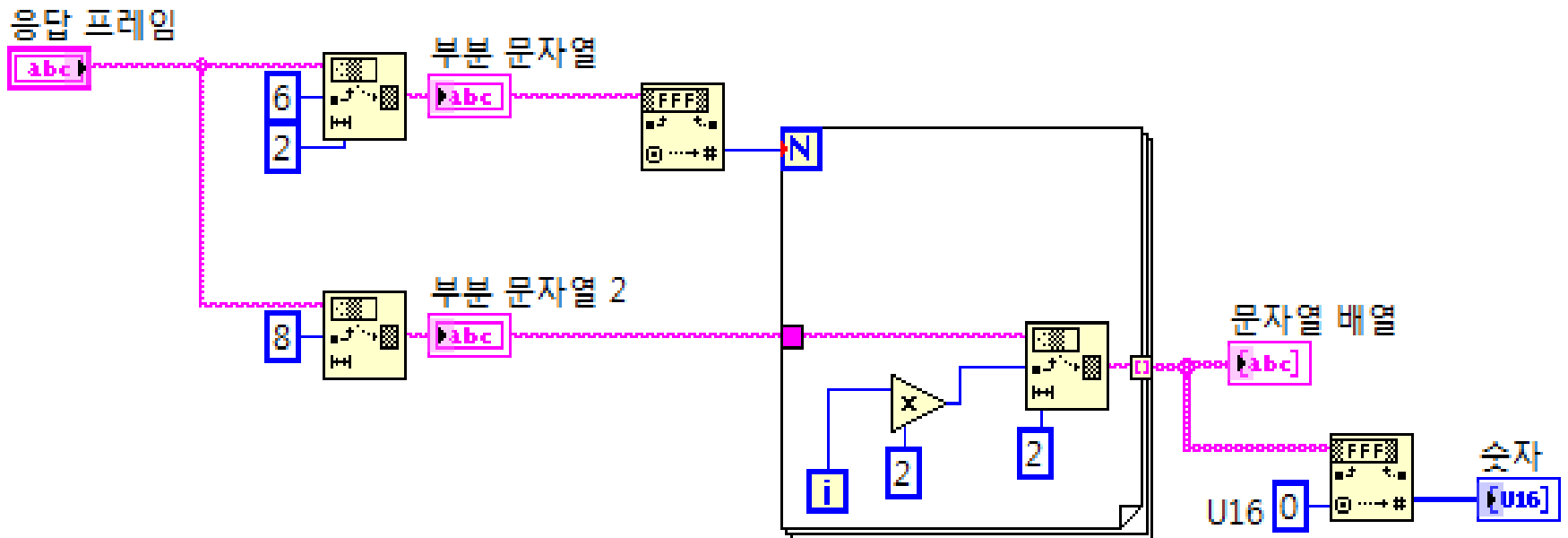


# 실습 2-8: 문자열 함수 사용

- 문자열 연결, 문자열 길이, 문자열 잘라내기, 소수점/지수 문자열을 숫자로, 패턴 일치 함수



# 실습 2-9: 응답 프레임 문자열 처리



응답 프레임	문자열 배열	숫자
-0ARSB0430391A85 L	30	x 30
부분 문자열	39	x 39
04	1A	x 1A
부분 문자열 2	85	x 85
30391A85 L		

# 경로

- 파일 또는 폴더의 위치 경로
- 프로그래밍 > 파일 I/O 팔레트
  - 경로 만들기
  - 경로 분리
  - 파일 상수

