I

정보의 표현

우리가 일상생활에서 접하는 수많은 자료는 문자, 그림, 소리, 동영상 등의 다양한 형태로 나타나는데, 컴퓨팅 시스템에서의 자료는 숫자로 저장되어 처리된다. 스마트폰 메신저로 친구들과 주고받는 대화 속 문장이나 그림, 동영상 등이 실제로는 모두 숫자로 구성되어 있다.

이번 단원에서는

자료와 정보, 수를 표시하는 방법인 진법을 알아보고, 2진수의 연산 과정과 문자, 그림, 소리, 동영상 등이 어떻게 숫자로 저장되는지 알아본다.

01 수의 체계

자료
정보
수의 체계
진법

학습 목표

- 자료와 정보의 개념을 설명할 수 있다.
- 컴퓨팅 시스템에서 사용하는 수의 체계를 설명할 수 있다.
- 여러 진법 변환 방법을 알고 상호 변환할 수 있다.

열려라! 생각

우리는 컴퓨터를 이용하여 문서, 그림, 소리, 동영상 등 다양한 종류의 파일을 작성하거나 내려받는다. 이렇게 작성한 파일이나 내려받은 파일은 컴퓨터에 0 또는 1의 형태로 저장된다.

Q 컴퓨터는 왜 0부터 9까지의 숫자가 아닌 0과 1의 두 가지 숫자만을 사용해서 정보를 저장하는 것일까?

1 자료와 정보

현대 사회는 '정보화 사회'로 불릴 만큼 정보(information)는 일상생활과 떼려야 뗄 수 없는 요소가 되었다. 기상청은 기온, 습도, 풍향, 풍속 등의 기상 자료(data)를 바탕으로 강수 확률, 불쾌지수 등의 기상 정보를 제공하며, 의사는 환자의 혈압, 맥박, 엑스레이(X-ray) 등의 자료를 진료에 활용하여 진단 정보를 생산한다. 또한 차량용 내비게이션 소프트웨어는 지도 자료와 실시간 통행량 자료에 근거하여 최단 시간 경로 정보를 알려 주고, 최저가 비교 인터넷 쇼핑몰은 다양한 상점의 가격을 비교하여 최저가 정보를 제공한다.

자료
현실 세계에서 측정하고 수집한 사실이나 값

정보
어떠한 목적이나 의도에 맞게 자료를 가공 처리한 것

기온은 우리가 외출할 때 옷차림을 결정하는 정보가 되며, 최저가 정보는 연간 최저가 변동 추이 그래프를 작성할 때 필요한 자료가 된다.

1 자료와 정보의 개념

정보를 생성할 때 사용하는 자료는 이전의 정제되지 않은 단순한 사실을 뜻한다. 자료가 다양한 처리 과정을 거쳐 판단의 근거로 활용 가능한 의미와 가치를 지니게 되면, 비로소 정보의 틀을 갖추게 된다.

자료를 여러 가지 방법으로 가공하여 사용자가 유용하게 사용할 수 있도록 생성한 것을 정보라고 한다. 우리는 사회로부터 다양한 정보를 접하고 살아가며, 의사 결정을 할 때 중요한 근거로 활용하고 있다.

꿀팁
자료는 경우에 따라 그 자체로 정보 역할을 할 때도 있으며, 정보는 또 다른 정보를 생성하기 위한 자료가 되기도 한다.

자료	**측정한 값 또는 사실** 예 7월 26일 오후 세 시의 기온과 습도는 각각 32℃, 75%이다.
정보	**의미 있는 자료. 즉 자료를 처리하여 의사 결정에 사용할 수 있도록 가공한 것** 예 7월 26일 오후 세 시의 불쾌지수는 85.25이다.

자료와 정보의 구분

2 정보 처리

꿀팁

우리가 엑셀 프로그램을 이용해서 많은 자료를 기록하고 이로부터 통계, 확률 등 다양한 정보를 생성하는 것도 정보 처리 시스템을 이용한 정보 처리 과정의 한 예가 된다.

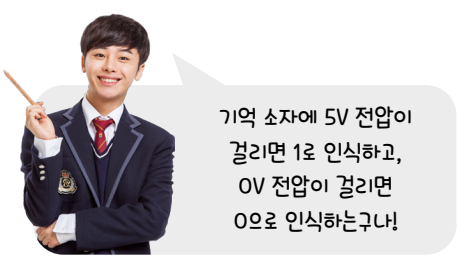
자료를 분류, 정리, 선택, 연산하여 정보를 만드는 과정을 정보 처리(information processing)라고 한다. 정보 처리는 많은 양의 자료를 다루는 경우가 많으므로 컴퓨팅 시스템을 주로 이용하는데, 이때 정보 처리를 위하여 구축한 컴퓨팅 시스템을 정보 처리 시스템(information processing system)이라고 한다.

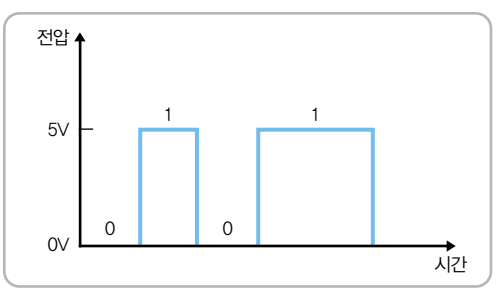
3 컴퓨팅 시스템에서의 자료 표현

꿀팁

컴퓨팅 시스템은 0과 1의 두 가지 수(2진수)만을 사용하는 2진법 체계이다.

사람은 문자나 그림, 숫자 등으로 여러 가지 정보를 표현하지만, 컴퓨팅 시스템은 0, 1 두 개의 숫자만으로 자료를 표현한다. 즉, 전기를 이용하여 내부 기억 장치의 기억 소자에 자료를 저장하는데, 전기가 흐르면 1로 표현하고 전기가 흐르지 않으면 0으로 간주한다. 기억 소자는 0과 1 중에서 하나의 숫자만 나타낼 수 있으므로 컴퓨팅 시스템은 큰 숫자나 대용량의 자료를 표현하기 위하여 많은 기억 소자를 내장하고 있다.

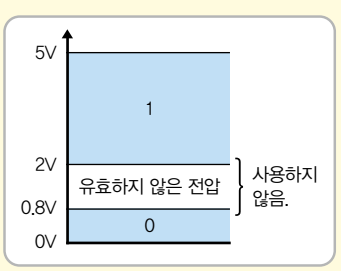




기억 소자의 전압 인식

한 걸음 더 컴퓨팅 시스템이 수를 표현하는 방법

컴퓨팅 시스템은 왜 우리가 일상에서 사용하는 10진수로 수를 표현하지 않고, 0과 1의 두 가지 숫자만을 사용하여 수를 표현할까?



하나의 기억 소자가 0부터 9까지의 열 가지 숫자를 나타내기 위해서는 기억 소자에 흘려보낼 수 있는 최대 전압인 5V를 10개의 구간으로 나누어 구분해야 한다. 하지만 나눈 구간의 개수가 많을수록 컴퓨팅 시스템 동작 중 내부와 외부에서 발생하는 잡음 때문에 숫자가 왜곡될 가능성이 커진다.

반면, 5V를 2개의 구간으로만 나누어 0과 1의 두 가지 숫자만 사용한다면 잡음으로 인한 숫자의 왜곡 가능성은 낮아진다. 따라서 하나의 기억 소자는 0과 1의 두 가지 숫자만을 표현하게 되는데, 이러한 기억 소자를 여러 개 조합하여 다양한 수를 표현한다.

2 진법

진법은 수를 표현하는 다양한 방법 중 한 가지이다. 숫자의 위치에 따른 가중치로 수를 표현하는데, 사용하는 숫자의 가짓수를 기준으로 10진법, 2진법, 8진법, 16진법 등 여러 종류로 나뉜다.

진법을 사용하여 표현한 수를 진수라고 한다. 진수는 몇 진수인지 나타내는 밑수(base)를 오른쪽에 추가로 표기한다. 10진수의 밑수는 10, 2진수의 밑수는 2로 $483_{(10)}$, $1101_{(2)}$ 등과 같이 나타낸다.

꿀팁

10진수의 밑수는 생략할 수 있다.

1 10진법

10진법은 우리가 일상에서 사용하는 진법으로, 0부터 9까지의 아라비아 숫자로 수를 표현한다. 인류는 예로부터 10진법을 널리 사용해 왔는데, 그 까닭은 사람의 손가락이 총 열 개인 것에서 비롯되었다.

10진법을 사용하여 표현한 수인 10진수의 자리 1개는 0부터 9까지의 수를 나타내며, 한 자리 위의 값은 10배가 된다. 예를 들어, 483은 실질적으로 $4\times10^2+8\times10^1+3\times10^0$이 된다. 여기서 10^2, 10^1, 10^0 등 자릿수에 곱해지는 수를 가중치라고 한다.

$$483 = 4\times10^2 + 8\times10^1 + 3\times10^0$$

가중치

2 2진법

2진법은 컴퓨팅 시스템에서 수를 다루기 위한 진법으로, 0과 1 두 가지 숫자만을 사용하여 수를 표현한다.

2진법을 사용하여 표현한 수인 2진수의 자리 1개는 0 또는 1을 나타내며, 한 자리 위의 값은 2배가 된다. 예를 들어, $1101_{(2)}$은 실질적으로 $1\times2^3+1\times2^2+0\times2^1+1\times2^0$이 된다.

$$1101_{(2)} = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$$
$$= 13_{(10)}$$

꿀팁

2진수는 한 자리가 나타낼 수 있는 수의 가짓수가 두 가지이므로 큰 수를 나타내기 위해서는 많은 자릿수가 필요하여 사람이 다루기에는 불편하다. 따라서 컴퓨터 프로그램에서 수를 사람에게 보여 줄 때는 10진수나 16진수 등으로 변환하여 출력하는 경우가 많다.

3 그 밖의 진법

n진법은 "n가지의 숫자를 사용하여 수를 표현한다."라는 뜻으로, 10진법과 2진법뿐만 아니라 8진법, 16진법, 60진법 등의 수많은 진법이 존재할 수 있다.

8진법은 0부터 7까지의 숫자를 사용하여 수를 표현하는 진법으로, 리눅스 운영체제의 파일 권한 표시 등 특정 상황에서 종종 사용된다.

16진법은 0~9까지 10개의 아라비아 숫자와 알파벳 중 A, B, C, D, E, F를 추가로 사용한다. 이때 A는 10, B는 11, C는 12, D는 13, E는 14, F는 15의 10진수에 해당한다. 16진법은 컴퓨터 내부에서 이용하는 2진수를 사람에게 출력해서 보여 줄 때 많이 사용한다.

바이트(byte)

8개의 비트를 묶은 단위. 기억 장치에 자료를 저장하거나 연산할 때는 바이트 단위로 이루어진다.

비트(bit)

0 또는 1의 2진수 한 자리

컴퓨터 내부에서 수의 저장, 연산 등은 바이트(8비트) 단위로 이루어진다. 1바이트로 표현이 가능한 가장 큰 2진수는 11111111로 여덟 자리를 차지하지만, 16진수로 출력하면 FF로 두 자리면 충분하다.

$$11111111_{(2)} = FF_{(16)} = 255_{(10)}$$

10진수 0부터 19까지의 수에 대응하는 2진수, 8진수, 16진수를 표로 정리하면 다음과 같다.

10진수	2진수	8진수	16진수
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9

10진수	2진수	8진수	16진수
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13

3 진법 변환

컴퓨터는 2진수로 수를 표현하고 연산한다. 사람은 주로 10진수로 컴퓨터에 수를 입력하므로 컴퓨터는 입력된 10진수를 2진수로 바꾸어서 저장하고 연산하며, 출력할 때는 10진수로 변환하여 보여 준다. 이처럼 임의의 진수를 다른 진수로 바꾸는 것을 진법 변환이라고 한다.

1 10진수 → 다른 진수

10진수를 2진수나 8진수, 16진수 등 다른 진수로 변환할 때는 몫이 0이 될 때까지 변환하려는 진수의 밑수로 계속 나눈 후 나머지를 역순으로 적는다.

꿀팁

10진수를 2진수로 변환한다면, 2로 계속 나눈 후 나머지를 역순으로 정리한다.

예제 따라 하기 10진수 29를 2진수, 8진수, 16진수로 각각 변환해 보자.

10진수 29를 2진수, 8진수, 16진수로 변환하면 다음과 같다.

[2진수로 변환]

2 | 29
2 | 14 … 1
2 | 7 … 0
2 | 3 … 1
2 | 1 … 1
0 … 1

$\therefore 29_{(10)} = 11101_{(2)}$

[8진수로 변환]

8 | 29
8 | 3 … 5
0 … 3

$\therefore 29_{(10)} = 35_{(8)}$

[16진수로 변환]

16 | 29
16 | 1 … 13
0 … 1

$\therefore 29_{(10)} = 1D_{(16)}$

스스로 해 보기 10진수를 다른 진수로 변환하기

10진수 180을 2진수, 8진수, 16진수로 각각 변환해 보자.

[2진수로 변환]

[8진수로 변환]

[16진수로 변환]

2 다른 진수 → 10진수

2진수, 8진수, 16진수 등 10진수가 아닌 다른 진수를 10진수로 변환할 때는 각 자릿수에 가중치를 곱한 값을 모두 더한다.

예제 따라 하기 $1101_{(2)}$, $372_{(8)}$, $8C_{(16)}$를 10진수로 각각 변환해 보자.

$1101_{(2)}$을 10진수로 변환하면 다음과 같다.

$1101_{(2)} = 1\times2^3+1\times2^2+0\times2^1+1\times2^0$

$= 8+4+1$

$= 13_{(10)}$

$372_{(8)}$를 10진수로 변환하면 다음과 같다.

$372_{(8)} = 3\times8^2+7\times8^1+2\times8^0$

$= 192+56+2$

$= 250_{(10)}$

$8C_{(16)}$를 10진수로 변환하면 다음과 같다.

$8C_{(16)} = 8\times16^1+12\times16^0$

$= 128+12$

$= 140_{(10)}$

스스로 해 보기

다른 진수를 10진수로 변환하기

$10110_{(2)}$, $205_{(8)}$, $1FD_{(16)}$를 10진수로 각각 변환해 보자.

꿀팁

2진수를 16진수로 변환할 때는 2진수 네 자리씩 묶어서 16진수로 변환하고, 반대로 16진수를 2진수로 변환할 때는 16진수 한 자리씩 2진수 네 자리로 변환한다.

3 2진수 ↔ 16진수

2진수를 16진수로, 16진수를 2진수로 변환하기 위하여 중간 과정에서 10진수로 변환한 후 해당 진수로 변환할 수 있지만, 다음 사실을 이용하면 훨씬 간단하게 변환할 수 있다.

> 2진수 네 자리가 16진수 한 자리에 해당한다.

예제 따라 하기 2진수 $101100_{(2)}$을 16진수로, 16진수 $57_{(16)}$을 2진수로 변환해 보자.

2진수 $101100_{(2)}$을 16진수로 변환하면 다음과 같다.

$\underline{0010}\ \underline{1100}$

$2\quad C = 2C_{(16)}$

16진수 $57_{(16)}$을 2진수로 변환하면 다음과 같다.

$\underline{5}\quad\underline{7}$

$0101\ 0111 = 1010111_{(2)}$

스스로 해 보기

1 2진수 $1010011101_{(2)}$을 16진수로 변환해 보자.

2 16진수 $3FB_{(16)}$를 2진수로 변환해 보자.

한 걸음 더 2진수 ↔ 8진수 변환

2진수 세 자리는 8진수 한 자리에 해당한다. 2진수와 16진수 간의 변환 방법처럼 묶어서 변환하되, 세 자리 단위로 하는 것이다.

2진수 $1101010_{(2)}$을 8진수로 변환

$$\underline{001}\ \underline{101}\ \underline{010}$$
$$1\quad 5\quad 2 \quad = 152_{(8)}$$

8진수 $307_{(8)}$을 2진수로 변환

$$\underline{3}\quad \underline{0}\quad \underline{7}$$
$$011\quad 000\quad 111 \quad = 11000111_{(2)}$$

윈도 운영 체제에 포함된 계산기 프로그램을 사용하여 진수 변환 작업을 편하게 할 수 있다. 왼쪽 위의 메뉴(≡)를 클릭한 후 '프로그래머'를 선택하면, 입력한 수에 해당하는 16진수(HEX), 10진수(DEC), 8진수(OCT), 2진수(BIN)를 쉽게 확인할 수 있다.

HEX, DEC, OCT, BIN을 클릭하면, 각각에 해당하는 진수로 입력할 수도 있다.

컴퓨팅 읽기 자료

Smart Reading

10진법만 진법일까?

우리 주위에는 대부분 0부터 9까지의 숫자를 사용한 10진법으로 수치, 수량 등의 수를 나타내고 있다. 하지만 10진법 이외에 다른 진법을 사용하는 것들도 존재한다.

시계

• 시(hour)

12진법이 사용된다. 시를 10진법이 아닌 12진법으로 정한 까닭은 12의 약수(1, 2, 3, 4, 6, 12)가 10의 약수(1, 2, 5, 10)보다 많아 시간을 나누어 활용하는 것이 편리하기 때문이다. 각도기 없이 원을 12등분하는 쉬운 방법이기도 하다.

• 분(minute), 초(second)

60진법이 사용된다. 60의 약수는 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60으로 꽤 많기 때문에 시보다 더 작은 분이나 초 단위로 시간을 쪼개어 쓸 때 매우 편리하다.

달력

• **월(month)**

1월부터 12월까지로 구성된 12진법이 사용된다. 지구는 1년을 주기로 계절이 반복되는데, 그동안 하늘에서는 달이 열두 번 차올랐다 지는 것을 볼 수 있다. 달이 한 번 차올랐다 지는 것을 하나의 월로 여긴 것이다.

• **주(week)**

일, 월, 화, 수, 목, 금, 토로 구성된 7진법이 사용된다. 한 주를 7일로 하게 된 까닭은 기독교, 이슬람교, 유대교 등 주요 종교들이 7일마다 종교 예배를 하였던 것이 절대적이었다. 프랑스 혁명 시기에 한 주를 10일로 바꾸거나, 공산주의 혁명 이후 소비에트 연방에서 한 주를 6일로 바꾸려는 등 변화의 시도가 있었으나 모두 실패하였다.

기타_마야 숫자

콜럼버스 이전 시대의 마야 문명에서 사용한 숫자로 20진법이 사용된다.

기호	이름	10진수	기호	이름	10진수
	시스 임	0		라훈	10
	훈	1		불룩	11
	카아	2		라흐카	12
	오스	3		오스라훈	13
	칸	4		칸라훈	14
	호오	5		호오라훈	15
	우아크	6		우아크라훈	16
	우우크	7		우우크라훈	17
	우아사아크	8		우아사아크라훈	18
	보론	9		보론라훈	19

프로젝트 활동

진수 상호 간 변환 방법 발표하기

2진수, 8진수, 10진수, 16진수의 상호 간 변환 방법을 각각의 예시를 제시하여 발표한다.

○ 수행 순서

1. 모둠당 3~5명씩 구성하여 총 6개의 모둠을 설정하고 발표자를 정한다.
2. 모둠별로 진법 변환할 수를 선택한다.

예

- 모둠 1: 2진수 ↔ 8진수
- 모둠 2: 2진수 ↔ 10진수
- 모둠 3: 2진수 ↔ 16진수
- 모둠 4: 8진수 ↔ 10진수
- 모둠 5: 8진수 ↔ 16진수
- 모둠 6: 10진수 ↔ 16진수

3. 선택한 수를 진법 변환한다.

2진수 → 8진수

① 2진수를 오른쪽부터 세 자리씩 분리한다.
예 $0001\ 0101_{(2)}$ → 000 010 101
② 묶은 영역을 10진수로 표현한다.
예 000=0, 010=2, 101=5
③ 구한 수들을 영역 순서대로 합친다.
∴ $25_{(8)}$

8진수 → 2진수

① 8진수 한 숫자당 2진수의 세 자리 영역으로 간주하여 변환한다.
예 $60_{(8)}$ → 6=110, 0=000
② 구한 2진수 표현을 (숫자로 인식해 더하지 말고 문자처럼 취급해) 순서대로 영역을 합친다.
∴ $110\ 000_{(2)}$

2진수 → 10진수

① 2진수 수마다 2의 n제곱을 해 준다(n은 숫자의 자릿수, n>=0).
예 $01100011_{(2)}$ → $0\times2^7=0$, $1\times2^6=64$, $1\times2^5=32$, $0\times2^4=0$, $0\times2^3=0$, $0\times2^2=0$, $1\times2^1=2$, $1\times2^0=1$
② 결괏값을 모두 더한다.
예 64+32+2+1=99
∴ $99_{(10)}$

10진수 → 2진수

① 10진수 수를 몫이 0이 될 때까지 2로 나눈다.
예 $30_{(10)}$

÷	30	나머지
2	15	0
2	7	1
2	3	1
2	1	1
2	0	1

② 나머지를 아래로부터 위로 역순으로 읽는다.
∴ $11110_{(2)}$

2진수 → 16진수

① 2진수를 오른쪽부터 네 자리씩 분리한다.

예 $11000001_{(2)}$ → 1100 0001

② 나눈 수를 10진수로 변환시킨다.

예 $1100_{(2)}$ → 12, $0001_{(2)}$ → 1

③ 10진수로 변환한 것을 16진수로 변환시킨다.

16	0	1	2	3	4	5	6	7
10	0	1	2	3	4	5	6	7
16	8	9	A	B	C	D	E	F
10	8	9	10	11	12	13	14	15

예 12 → C, 1 → 1

∴ $C1_{(16)}$

16진수 → 2진수

① 16진수를 각 자리마다 분리한다.

예 $F5_{(16)}$ → F5

② 나눈 수를 10진수로 변환한다.

예 $F_{(16)}$ → 15, $5_{(16)}$ → 5

③ 각 10진수를 네 자리의 2진수로 변환하고 붙인다.

예 15 → $1111_{(2)}$, 5 → $0101_{(2)}$

∴ $11110101_{(2)}$

8진수 → 10진수

① 8진수 수마다 8의 n제곱을 해 준다(n은 숫자의 개수, n>=0).

예 $74_{(8)}$ → $7\times8^1=56$, $4\times8^0=4$

② 결괏값을 모두 더한다.

예 56+4=60

∴ $60_{(10)}$

10진수 → 8진수

① 10진수 수를 몫이 0이 될 때까지 8로 나눈다.

예 $30_{(10)}$

÷	30	나머지
8	3	6
8	0	3

② 나머지를 아래부터 위로 역순으로 읽는다.

예 $36_{(8)}$

4. 변환 방법을 모둠 내 발표자가 발표한다.

1 대상 진수로 변환한 후 다시 원래 진수로 변환하였을 때 같은 값이 되는지 확인해 보자.

2 중간에 임시로 다른 진수로의 변환 절차가 필요한 모둠이 있는지 확인한 후 해당 모둠이 있다면 어떤 진수로의 변환인지 논의해 보자.

02 디지털 정보의 연산

수의 표현
수의 연산

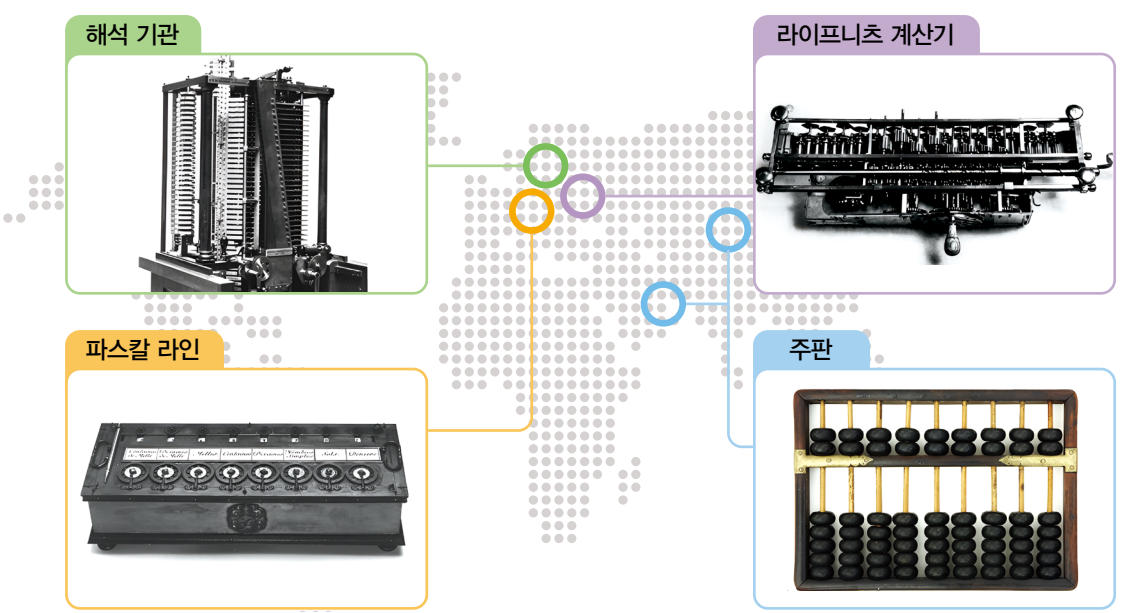
학습 목표

- 컴퓨터에서 정수와 실수가 어떻게 표현되는지 설명할 수 있다.
- 컴퓨터에서 처리하는 2진수의 덧셈과 뺄셈 연산 처리 원리를 설명할 수 있다.
- 컴퓨터에서 처리하는 2진수의 곱셈과 나눗셈 연산 처리 원리를 설명할 수 있다.

열려라! 생각

인류는 수의 연산을 쉽고 빠르게 하기 위하여 도구를 개발해 왔다. 고대 메소포타미아 지역과 중국 등에서는 주판을 사용하였으며, 1642년 프랑스의 파스칼은 톱니바퀴를 이용해서 덧셈, 뺄셈이 가능한 기계식 계산기인 파스칼 라인을 개발하였다. 1671년에는 독일의 라이프니츠가 곱셈과 나눗셈까지 가능한 계산기를 발명하였다. 이후 영국의 배비지가 1822년 다항 함수를 계산할 수 있는 차분 기관을 개발하였으며, 1837년에는 차분 기관을 개선하여 50자리의 숫자 1,000개를 저장하고 여러 용도로 사용이 가능한 해석 기관을 발표하였다.

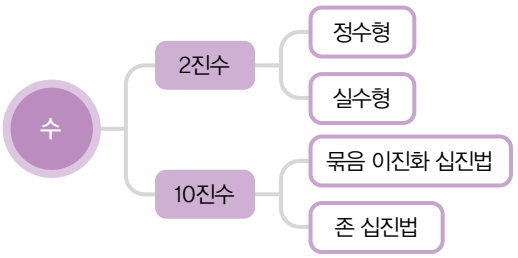
해석 기관은 증기 기관을 동력으로 사용하고 프로그램과 데이터를 천공 카드로 입력받았다. 이는 약 100년 뒤에 나올 첫 번째 범용 컴퓨터 모습과 비슷한 것으로, 컴퓨터 역사의 한 획을 긋는 중요한 사건으로 평가받는다.



Q 컴퓨터는 어떻게 2진수의 사칙 연산을 수행할까?

1 수의 표현

컴퓨터는 2진수로 수를 저장하고 연산을 수행하지만, 몇몇 컴퓨터는 10진수 방식도 지원한다. 2진수는 소수점 이하가 없는 정수형과 소수점 이하가 있는 실수형으로 구분해 처리하며, 10진수는 묶음 이진화 십진법과 존 십진법으로 구분해 처리한다.



수의 구분

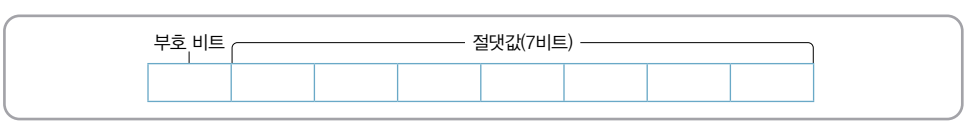
1 2진수

컴퓨터에서 수를 저장하고 연산하기 위하여 사용한다. 실제 램(RAM)이나 디스크 등의 기억 장치에는 바이트 단위로 저장된다.

1. 정수형

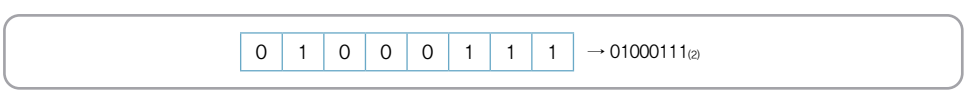
최상위 한 개의 비트가 부호를 나타내고, 나머지 비트가 절댓값을 나타낸다. 부호 비트가 1이면 음의 정수가 되고, 0이면 양의 정수가 된다.

▶ 1바이트로 정수를 표현할 때



(1) 양의 정수

10진수 71을 1바이트에 표현하면 다음과 같다.



(2) 음의 정수

10진수 −71을 다음 방법 중 하나를 사용하여 표현할 수 있다.

부호화 절대치 방법	① 부호 비트를 1로 설정하고, 나머지 비트에 절댓값을 설정한다. ② 사람이 보기에 직관적인 면이 있으나 음의 정수가 포함된 연산에 추가적인 보정 작업이 필요하다.	1 1 0 0 0 1 1 1 ↓ 부호 비트를 1로 1 1 0 0 0 1 1 1
1의 보수 방법	① 양숫값의 모든 비트를 반전시킨다(0은 1로, 1은 0으로 바꿈). ② 0을 나타내는 값이 $00000000_{(2)}$과 $11111111_{(2)}$ 두 가지가 존재한다.	0 1 0 0 0 1 1 1 ↓ 모든 비트를 반전(0→1, 1→0) 1 0 1 1 1 0 0 0
2의 보수 방법	① 1의 보수 방법에 1을 더한다. 즉, 양숫값의 모든 비트를 반전시킨 후 1을 더한다. ② 부호화 절대치 방법과 1의 보수 방법에서 발생하는 문제점이 없어 컴퓨터에서 널리 사용한다.	0 1 0 0 0 1 1 1 ↓ 모든 비트를 반전한 후 +1 1 0 1 1 1 0 0 1

램

데이터가 저장된 위치와 관계없이 일정한 시간 내에 기억 내용을 읽거나 쓸 수 있는 기억 장치

꿀팁

음의 정수를 표현하는 방법에는 부호화 절대치 방법, 1의 보수 방법, 2의 보수 방법 등이 있다.

절댓값(절대치)

양 또는 음의 부호를 떼어 버린 수

보수

각 자리의 숫자의 합이 어느 일정한 수가 되게 하는 수

2. 실수형

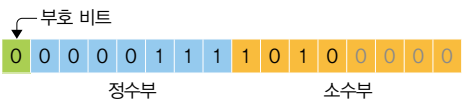
소수점이 포함된 실수를 표현할 때는 고정 소수점 방식 또는 부동 소수점 방식을 사용한다.

(1) 고정 소수점 방식

꿀팁

고정 소수점 방식은 구현이 편리하지만, 표현이 가능한 수의 범위와 정밀도가 낮아서 거의 사용하지 않는다.

소수점의 위치가 고정된 것으로, 정수 부분이 들어갈 공간의 크기와 소수점 이하 부분이 들어갈 공간의 크기가 미리 정해져 있다. 예를 들어 $7.625_{(10)}$라는 실수를 2진수로 바꾸면 $111.101_{(2)}$이며, 이를 2바이트 고정 소수점 방식으로 나타내면 다음과 같다.



(2) 부동 소수점 방식

주로 사용하는 실수형 표현 방법이다. 고정 소수점 방식처럼 정수부와 소수부에 2진수를 그대로 넣는 것이 아니라 '$1.xxx \times 2^n$'과 같은 형식의 수로 바꾸는 정규화 작업을 수행한 후 부호, 지수, 소수점 이하 가수를 각각 저장한다.

지수

어떤 수나 문자의 오른쪽 위에 덧붙여 쓰여 그 거듭제곱을 한 횟수를 나타내는 문자나 숫자

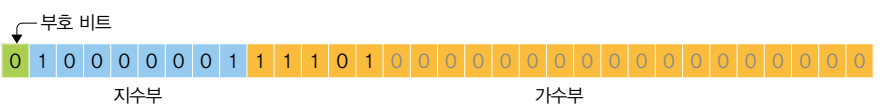
가수

상용로그의 값에서 0과 같거나 0보다 크고 1보다 작은 소수

예를 들어 $7.625_{(10)}$를 정규화하면 다음과 같다.

$$7.625_{(10)} = 111.101_{(2)} = 1.11101 \times 2^2$$

4바이트를 사용하여 표현하면 다음과 같다.



*부호 비트: $0_{(2)}$(양의 정수) *지수부: $127 + 2 = 129 = 10000001_{(2)}$ *가수부: $11101_{(2)}$

꿀팁

지수부의 경우 127을 더한 값으로 저장한다.

한 걸음 더 실수의 2진수 변환

소수점 이하의 값을 2진수로 변환할 때는 소수점 이하 부분이 0이 될 때까지 2를 계속 곱해 나가며 정수 부분을 순서대로 적는다.

예 10진수 0.375를 2진수로 변환

$$\begin{array}{rcrcrl} 0.375 & & 0.750 & & 0.500 & \\ \times\ \ 2 & \rightarrow & \times\ \ 2 & \rightarrow & \times\ \ 2 & \therefore\ 0.375_{(10)}=0.011_{(2)} \\ \hline \text{⓪}.750 & & \text{①}.500 & & \text{①}.000 & \end{array}$$

2 10진수

10진수도 처리할 수 있는 일부 컴퓨터는 2진수 네 자리를 묶어 10진수 한 자리를 나타내는 이진화 십진법(BCD: Binary-Coded Decimal)을 사용한다.

10진수	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

이를테면, 10진수 $29_{(10)}$는 2진수로는 $11101_{(2)}$이지만, 이진화 십진법 방식으로는 $00101001_{(2)}$이 된다. 컴퓨터는 주로 바이트 단위로 수를 처리하므로 이진화 십진법을 사용하는 방법은 묶음 이진화 십진법 또는 존 십진법 중 하나로 구현된다.

꿀팁

이진화 십진법은 2진수 네 자리가 10진수 한 자리에 바로 대응되어 사람이 직관적으로 수를 파악하기 쉬운 면이 있으나 버리게 되는 수가 있어 같은 수를 표현하더라도 더 많은 비트가 필요하다.

1. 묶음 이진화 십진법

한 바이트에 두 자릿수를 묶어서(packed) 저장하고 마지막 자리 다음의 4비트를 부호로 사용하는 이진화 십진법 구현 방법이다.

예 $729_{(10)} = 01110010\ 10011100_{(2)} = 72\ 9C_{(16)}$
(7 2 9 부호(+))

$-729_{(10)} = 01110010\ 10011101_{(2)} = 72\ 9D_{(16)}$
(7 2 9 부호(−))

꿀팁

비트는 0 또는 1의 값을 가질 수 있다. 니블(nibble)은 4비트에 해당하며, 바이트는 8비트(2니블)에 해당한다.

꿀팁

부호로 사용하는 수는 다른 자리들과의 구별을 위하여 양의 정수는 1100을 사용하고, 음의 정수는 1101을 사용한다.

2. 존 십진법

한 바이트에 한 자릿수를 하위 4비트에 저장하고, 상위 4비트에 존(zone)이라고 하는 특정한 비트열(1111)을 채우는 이진화 십진법 구현 방법이다.

예 $729_{(10)} = 11110111\ 11110010\ 11001001_{(2)} = F7\ F2\ C9_{(16)}$
(존 7 존 2 부호(+) 9)

$-729_{(10)} = 11110111\ 11110010\ 11011001_{(2)} = F7\ F2\ D9_{(16)}$
(존 7 존 2 부호(−) 9)

꿀팁

가장 오른쪽 바이트의 존은 부호를 나타내며, 양의 정수는 1100을 사용하고, 음의 정수는 1101을 사용한다.

한 걸음 더 3비트 및 4비트로 표현 가능한 수의 범위

3비트로 표현이 가능한 수의 범위: 0~7, 8(2^3)가지 / 4비트로 표현이 가능한 수의 범위: 0~15, 16(2^4)가지

10진수 0부터 9까지의 열 가지 숫자를 나타내기 위해서 3비트로는 부족하며 적어도 4비트가 필요하다. 하지만 4비트의 경우 나머지 여섯 가지(1010, 1011, 1100, 1101, 1110, 1111)는 사용하지 않고 버려진다.

2 2진수의 연산

컴퓨터는 덧셈 연산을 기본으로 뺄셈, 곱셈, 나눗셈을 수행한다. 뺄셈은 음의 정수를 더하는 방식으로 처리하고, 곱셈은 덧셈을 여러 번 하는 방식으로 처리하며, 나눗셈은 뺄셈을 여러 번 하는 방식으로 처리한다. 연산은 바이트 단위로 이루어지는데, 주로 1, 2, 4, 8바이트 중 하나가 된다.

1 덧셈

덧셈은 10진수의 덧셈과 마찬가지로 가장 아래 자릿수부터 더해 나간다. 10진수의 덧셈에서 10 이상이 되면 자리 올림을 하는 것처럼 2진수에서는 2 이상이 되면 자리 올림을 한다.

예 $1010_{(2)} + 11_{(2)}$

```
  1010
+   11
------
  1101
```

2 뺄셈

뺄셈은 음의 정수를 더하는 방식으로 처리된다. 즉, 2의 보수가 되는 값을 더하는 것이다.

꿀팁

$1001_{(2)}$의 2의 보수 = $1001_{(2)}$의 1의 보수 $0110_{(2)} + 1_{(2)} = 0111_{(2)}$

예 $1011_{(2)} - 1001_{(2)} = 1011_{(2)} + 0111_{(2)}$

$= 1\ 0010_{(2)}$

↑ 자리 올림의 수는 버린다.

3 곱셈

곱셈은 덧셈을 여러 번 하는 것으로서, 11×5로 예를 들면 11+11+11+11+11처럼 11을 5회 더하는 개념이다. 이러한 방식으로 곱셈을 연산할 수도 있지만, 만약 승수가 매우 큰 숫자일 경우 더하는 횟수가 지나치게 많아져 연산이 오래 걸리므로 사용하지 않는다.

승수

어떤 수에 곱하는 수

예 $1011_{(2)} \times 101_{(2)}$

```
  1011
×  101
------
  1011
 1011
------
110111
```

따라서 10진수 곱셈을 할 때와 같이 각 자릿수별로 계산한 후 이들을 모두 합하는 방식으로 처리한다. 곱하는 자릿수가 1일 때는 더하고, 자릿수가 0일 때는 더하지 않는다.

4 나눗셈

나눗셈은 뺄셈을 여러 번 하는 것으로서, 13÷5로 예를 들면 5보다 작아질 때까지 5를 계속 빼서(13−5−5=3) 그 횟수인 2가 몫이 되고 최종값인 3이 나머지가 되는 개념이다. 이러한 방식으로 나눗셈을 연산할 수도 있지만, 나누어지는 수가 클 경우는 빼는 횟수가 매우 많아져 시간이 오래 걸리므로 사용하지 않는다.

따라서 10진수 나눗셈을 할 때와 같이 각 자릿수별로 계산하여 빼는 방식으로 처리한다.

예 $1101_{(2)} \div 101_{(2)}$

$$
\begin{array}{r}
0010 \\
101\overline{)1101} \\
\underline{101} \\
11
\end{array}
$$

한 걸음 더 나눗셈의 연산

맨 왼쪽 숫자부터 나누는 수와 비교하여 이보다 작으면(뺄 수 없으면) 몫에 0을 추가하고, 오른쪽 숫자를 붙인 후 비교한다. 나누는 수 이상이 될 때까지(뺄 수 있을 때까지) 이를 반복한다. $1_{(2)}$부터 시작하여 $110_{(2)}$이 되면 $101_{(2)}$ 이상이 되므로 몫에 1을 추가하고 이를 뺀다. 빼고 난 값인 $11_{(2)}$과 나누는 수 $101_{(2)}$에 대하여 다시 위의 과정을 적용한다. $11_{(2)}$은 $101_{(2)}$보다 작으므로 몫에 0을 추가한다. 더 이상 나누어지는 수에 살펴볼 숫자가 없으므로 몫은 $10_{(2)}$, 나머지는 $11_{(2)}$이 된다.

1 < 101	몫에 0 추가
11 < 101	몫에 0 추가
110 >= 101	몫에 1 추가 후 나누어지는 수에서 뺌.
11 < 101	몫에 0 추가

몫: $10_{(2)}$, 나머지: $11_{(2)}$

프로젝트 활동

4비트 공간에서 표현 가능한 범위 알기

4비트 공간으로 부호화 절대치 방법, 1의 보수 방법, 2의 보수 방법 등을 사용하여 음의 정수를 나타낼 때, 표현 가능한 수의 범위를 알아본다.

○ 수행 순서

1. 모둠당 4~6명씩 구성하여 총 3개의 모둠을 설정한다.

2. 모둠별로 방법을 지정한 후 발표자를 정한다.

예

모둠 1	부호화 절대치 방법
모둠 2	1의 보수 방법
모둠 3	2의 보수 방법

3. 모둠별 지정된 방법으로 표현 가능한 수를 모두 조사한다.

맨 왼쪽의 비트를 음의 정수 부호 비트로 활용하자!

맨 왼쪽의 비트가 0이면 양의 정수, 1이면 음의 정수로 구분해야 해!

양식

모둠 1 부호화 절대치 방법	
2진수	10진수
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

모둠 2 1의 보수 방법	
2진수	10진수
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

모둠 3 2의 보수 방법	
2진수	10진수
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

4. 조사한 결과를 모둠 내 발표자가 발표한다.

1 0의 경우 +0, −0의 두 가지가 나오는지 확인해 보자.

2 어떤 방법이 수의 표현 범위가 가장 넓은지 확인해 보자.

3 음의 정수의 경우 어떤 표현 방법이 가장 직관적인지 확인해 보자.

03 디지털 정보의 표현

문자
그림
소리
동영상

학습 목표

- 컴퓨터에서 문자가 디지털 정보로 표현되는 원리를 설명할 수 있다.
- 컴퓨터에서 그림, 소리, 동영상 등이 디지털 정보로 표현되는 원리를 설명할 수 있다.
- 공개 소프트웨어를 이용하여 문자, 그림, 소리, 동영상 등을 디지털 정보로 표현할 수 있다.

열려라! 생각

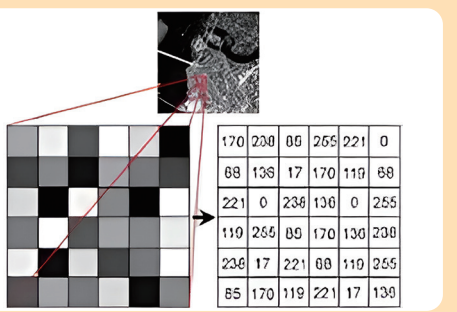
컴퓨터에서 모든 데이터(문자, 그림, 소리, 동영상 등)는 2진수로 표현된다.

문자

Binary	Letter	Binary	Letter	Binary	Letter
0100 0001	A	0100 1010	J	0101 0011	S
0100 0010	B	0100 1011	K	0101 0100	T
0100 0011	C	0100 1100	L	0101 0101	U
0100 0100	D	0100 1101	M	0101 0110	V
0100 0101	E	0100 1110	N	0101 0111	W
0100 0110	F	0100 1111	O	0101 1000	X
0100 0111	G	0101 0000	P	0101 1001	Y
0100 1000	H	0101 0001	Q	0101 1010	Z
0100 1001	I	0101 0010	R		

알파벳 A는 $01000001_{(2)}$, 알파벳 B는 $01000010_{(2)}$ 등으로 나타냄.

그림

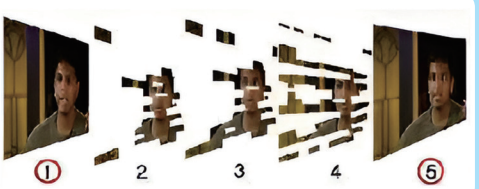


색깔 정보를 빛의 3원색인 빨강, 초록, 파랑으로 수치화해서 나타냄.

소리

주파수와 진폭 등을 수치화하여 저장함.

동영상



여러 장의 그림과 소리 정보를 함께 저장하며, 이때 용량이 커지므로 바뀐 부분만을 압축해서 저장함.

Q 한글 한 글자를 표현하기 위해서는 몇 바이트가 필요할까?

1 문자의 표현

컴퓨터는 특정 문자에 대응되는 2진수를 부여해 놓은 문자 인코딩(character encoding)이 있어 숫자뿐만 아니라 문자도 표현하고 저장할 수 있다. 초창기에는 세계 각국이 자신의 언어에 대하여 독자적인 문자 인코딩을 정하였기 때문에 여러 언어가 혼합된 문서에서 문제가 발생하였으나, 현재는 전 세계 모든 문자를 동시에 표현할 수 있는 유니코드(unicode) 방식의 문자 인코딩이 보편화되어 문제점이 개선되었다.

유니코드
컴퓨터에서 세계 각국의 언어를 통일된 방법으로 쓸 수 있게 만든 국제적인 문자 코드 규약

1 아스키코드

아스키코드(ASCII code)는 미국표준화협회가 7비트에 95개의 출력 가능한 문자와 33개의 제어 문자를 배치한 문자 인코딩이다. 52개의 알파벳 대소문자와 10개의 아라비아 숫자, 32개의 특수 문자, 1개의 공백 문자 등의 출력이 가능하다. 제어 문자는 $0000000_{(2)}$부터 $0011111_{(2)}$, $1111111_{(2)}$로 구성되는데, 현재는 다음 줄로 넘김, 백스페이스 등 몇 개를 제외하고는 거의 사용하지 않는다.

꿀팁
아스키(ASCII)는 미국정보교환표준부호(American Standard Code for Information Interchange)의 약자이다.

2 한글 문자 인코딩

유니코드가 보편화되기 이전에는 우리나라에서도 한글 표현을 위한 독자적인 문자 인코딩을 정하여 사용하였다. 한글 문자 인코딩은 표현 원리에 따라 크게 완성형 문자 인코딩과 조합형 문자 인코딩으로 구분한다.

꿀팁
일본어를 표현하는 Shift JIS, 중국어를 표현하는 GBK 등 언어마다 고유의 문자 인코딩이 존재한다.

꿀팁
조합형 문자 인코딩은 한글의 모든 글자를 온전히 표현할 수 있는 장점이 있다. 그러나 다양한 문자 인코딩이 난립하여 서로 호환되지 않는 점과 윈도 운영 체제의 확장 완성형 문자 인코딩 도입 등으로 인하여 결국 완성형 문자 인코딩에 시장을 내주게 되었다.

완성형 문자 인코딩

① 한글의 개별 글자마다 2진수를 부여해 놓은 문자 인코딩이다.
② KS X 1001(EUC-KR) 등 초기 완성형 문자 인코딩은 2,350자에만 2진수를 부여해 놓아 이외의 글자(쓗, 쌰 등)는 표현할 수 없는 문제점이 있었으나, 이를 개선한 CP949(MS949, 확장 완성형 등으로도 불림) 문자 인코딩이 개발되었다.
③ 2바이트를 이용하여 표현하며, 윈도 운영 체제에서 주로 사용되었다.

예

가	$1011000010100001_{(2)}$
각	$1011000010100010_{(2)}$
간	$1011000010100011_{(2)}$

조합형 문자 인코딩

① 한글의 초성, 중성, 종성별로 따로 2진수를 부여해 놓고, 실제 개별 글자에 대하여 이를 조합한 값을 사용하는 문자 인코딩이다.
② 2바이트를 이용하여 최상위 1비트는 항상 1로 설정해서 한글임을 나타내고, 나머지 15비트를 초성, 중성, 종성에 대하여 각각 5비트씩 할당하여 표현한다.

예

	한글 여부	ㄷ	ㅏ	ㄹ
달	1	$00101_{(2)}$	$00011_{(2)}$	$01001_{(2)}$
	$1001010001101001_{(2)}$			

3 유니코드

꿀팁

유니코드 방식의 문자 인코딩 중 UTF-8 문자 인코딩이 가장 널리 사용된다.

꿀팁

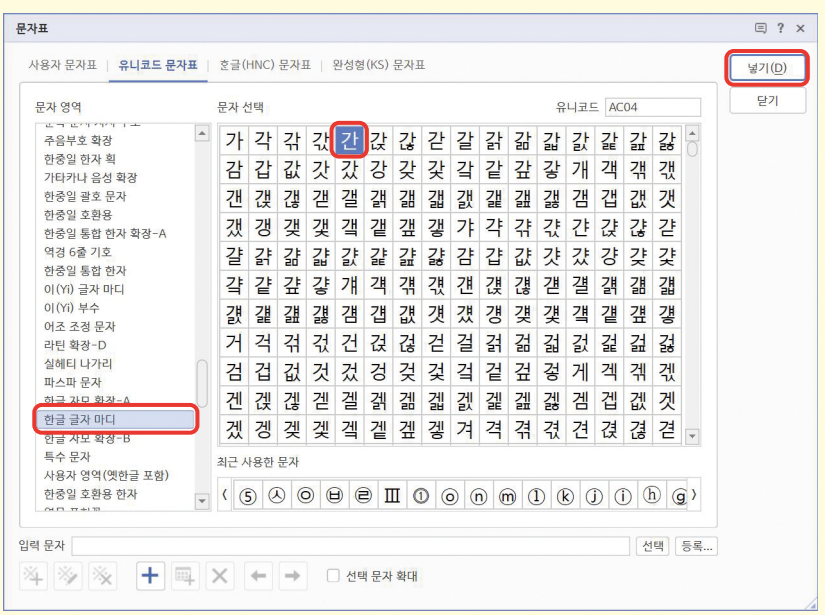
UTF-8은 가변 길이의 유니코드 문자 인코딩으로 아라비아 숫자, 알파벳 등은 글자당 1바이트를 차지하고, 한글은 글자당 3바이트를 차지한다.

기존의 언어별 문자 인코딩은 서로 간에 협의 없이 독자적으로 숫자를 부여하였기 때문에 다양한 문자가 섞여 있는 문서를 작성할 때 문제가 발생하였다. 예를 들어 한글과 일본어가 섞인 문서에서 한글의 특정 글자를 나타내는 값이 일본어 문자 인코딩에서도 사용하는 값일 경우, 한글과 일본어 중 어떤 글자로 표현할지 결정할 수 없게 된다.

유니코드는 이러한 문제점을 해결하기 위하여 만든 개념으로, 전 세계의 모든 문자를 겹치지 않게 일렬로 줄 세워 놓고 숫자를 부여한 방식이다. 유니코드는 컴퓨터에서 문자를 일관되게 표현하고 다룰 수 있도록 설계한 산업 표준이다. UTF-7, UTF-8, UTF-16 등이 유니코드 방식의 문자 인코딩에 속한다.

한 걸음 더 문자표에서 특정 문자에 대응하는 유니코드 숫자 확인하기

한글 프로그램에서 '입력 → 문자표' 메뉴를 선택하거나 Ctrl+F10 단축키를 눌러서 나오는 문자표에서 특정 문자에 대응하는 유니코드 숫자를 확인할 수 있다.



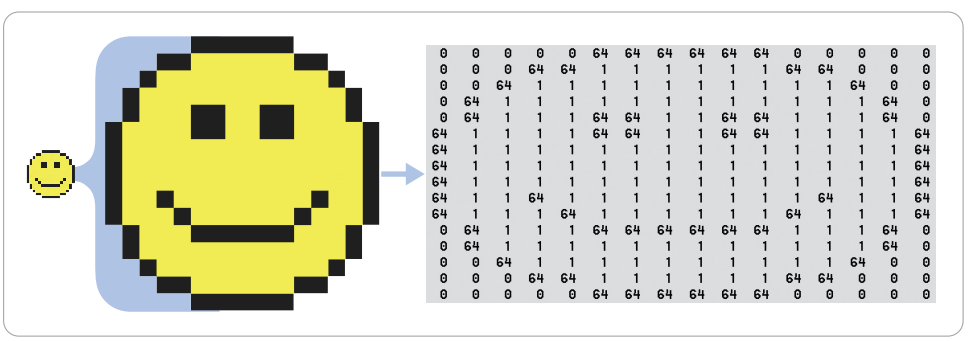
- '문자 영역'에서 '한글 글자 마디'를 선택하면, 한글이 모여 있는 유니코드 영역으로 이동할 수 있다.
- '문자 선택'에서 글자를 선택하면, 오른쪽 위의 '유니코드'에 해당 글자에 배정된 유니코드 숫자가 표시된다.
- 한글은 $AC00_{(16)}$부터 $D7A3_{(16)}$까지의 숫자를 배정받았다.

2 그림의 표현

그림은 색을 지닌 화소(pixel)들의 집합으로, 문자보다 다양한 정보를 나타낼 수 있다. 그림은 크기가 크고 많은 색상을 표현할수록 더 큰 공간이 필요하며, 저장 방식에 따라 비트맵(bitmap) 방식과 벡터(vector) 방식 등으로 구분된다.

1 비트맵 방식

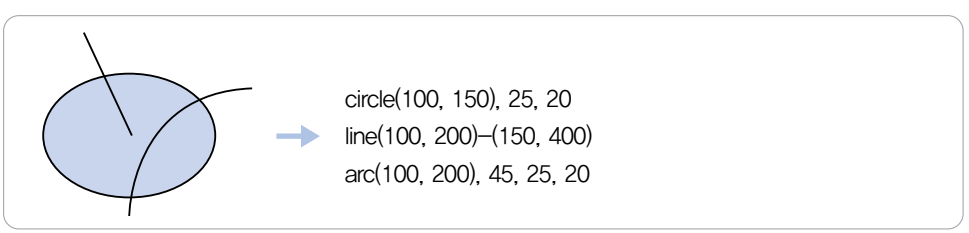
비트맵 방식은 색을 지닌 화소들의 2차원적 나열 정보를 그대로 저장하는 것으로, 래스터(raster) 방식이라고도 한다. 너비와 높이가 정의되어 화소 수가 고정되므로 확대하면 화소 자체가 크게 보이는 계단 현상이 일어난다. JPG, GIF, PNG, BMP 등 일반적으로 널리 사용되는 그림 파일 형식이 비트맵 방식에 해당하며, 보통 용량을 줄이기 위하여 압축 기술이 적용된다.



비트맵 방식

2 벡터 방식

벡터 방식은 점과 점 사이를 수학과 관련된 계산 형태로 표현하여 그림을 저장하는 방식이다. 표현에 한계가 있어 복잡한 그림은 형상으로 나타내기 어렵다. 하지만 확대 또는 축소할 때 배율을 적용하여 각 화소의 놓일 자리를 재배치함으로써 계단 현상을 피할 수 있다. AI, CDR, SVG 등의 그림 파일 형식이 벡터 방식에 속한다.



벡터 방식

JPG
표준으로 사용되는 그림 파일 형식

GIF
웹툰과 같이 색상 수가 제한된 웹 그래픽 작업이나 용량이 적은 애니메이션에 사용되는 그림 파일 형식

PNG
다양한 색과 투명 배경을 지원하는 그림 파일 형식

BMP
윈도 운영 체제의 표준 그림 파일 형식

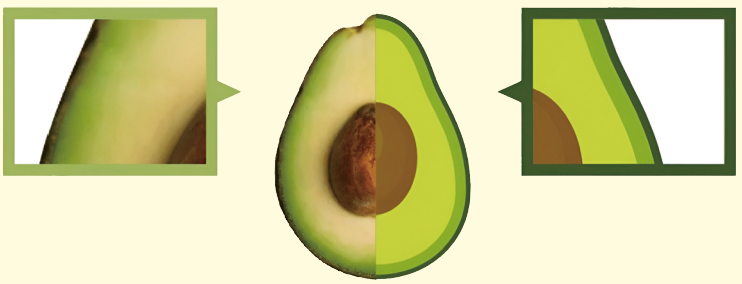
AI
어도비 일러스트레이터 프로그램에서 기본으로 사용하는 그림 파일 형식

CDR
코렐 드로 프로그램에서 기본으로 사용하는 그림 파일 형식

SVG
XML(eXtensible Markup Language) 기반의 파일 형식

한 걸음 더 비트맵과 벡터의 확대

[비트맵: 확대할 때 계단 현상이 발생함.] [벡터: 확대하여도 계단 현상이 발생하지 않음.]



한 걸음 더 색의 표현

[빛의 색 표현]

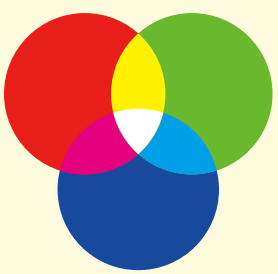
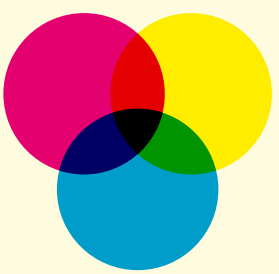
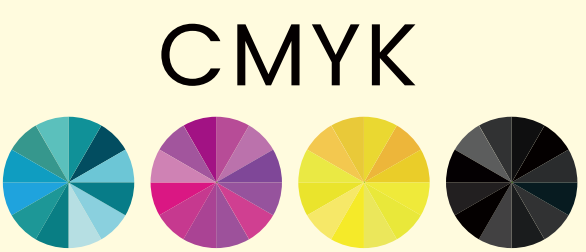


그림 파일은 일반적으로 색을 나타내기 위하여 빛의 3원색인 빨간색(Red), 초록색(Green), 파란색(Blue) 정보를 담은 RGB 방식을 사용한다. 모니터, 노트북이나 휴대 전화 액정 등 컴퓨팅 시스템에서 사용하는 영상 출력 장치는 빛으로 색을 표현하므로 RGB 방식의 처리가 간편하다.

[잉크의 색 표현]

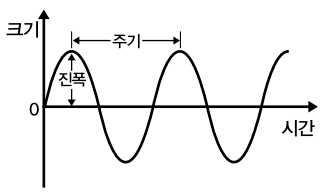


영상 출력 장치가 아닌 인쇄물 출력에서 주로 이용하는 그림 파일은 인쇄 장치가 사용하는 잉크의 색인 옥색(Cyan), 자홍색(Magenta), 노란색(Yellow), 검은색(blacK) 정보를 담은 CMYK 방식을 사용한다.



3 소리의 표현

소리는 물체의 진동으로 발생한 파동이 공기를 통하여 사람의 귀청을 울리어 귀에 들리는 것이다. 컴퓨팅 시스템에서 소리는 과거 스피커를 통하여 출력되는 정도였지만, 최근에는 목소리로 명령하거나 주변 음악으로 검색하는 등 입력 방식으로도 많이 이용한다.



진폭: 진동 중심에서 최대 변위의 크기
주기: 1회 진동하는 데 걸린 시간
주파수: 1초 동안 진동한 횟수(주파수 = $\frac{1}{\text{주기}}$)

▲ 파동의 구성 요소

▲ 소리(안녕하세요)의 파동 예

꿀팁

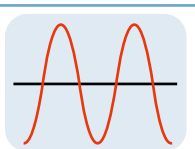
목소리의 기본 주파수는 남성이 100~150Hz, 여성이 200~250Hz이다. 여성의 목소리 기본 주파수가 남성보다 높으므로 고음으로 들린다. 참고로 200Hz라면, 1초에 성대가 200회 진동한다.

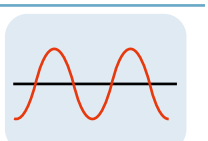
1 소리의 구성 요소

소리는 세기, 높낮이, 음색 등으로 특징을 나타낸다.

세기

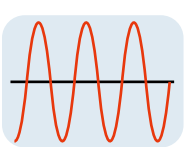
① 소리의 크기를 나타내는 것으로, 음량이라고도 한다.
② 파동의 진폭으로 결정되며, 진폭이 클수록 소리가 커진다.
예 음악을 재생할 때 볼륨을 조절하게 되면 파동의 진폭을 높이거나 낮추는 것
③ 단위는 주로 데시벨(dB)을 사용한다.

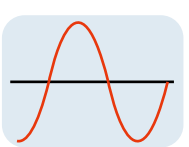

시끄러운 소리


조용한 소리

높낮이

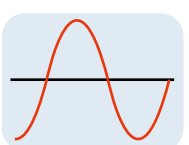
① 소리의 높낮이를 나타내는 것으로, 음고라고도 한다.
② 파동의 주파수로 결정되며, 주파수가 높을수록(주기가 짧을수록) 고음이 된다.
③ 일정 시간 동안 진동 횟수가 많을수록 고음이 된다.
예 피아노 건반에서 오른쪽으로 갈수록 주파수가 높은 파동이 발생한다.
④ 단위는 주로 헤르츠(Hz)를 사용한다.

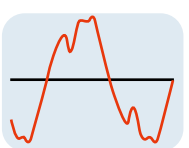

고음


저음

음색

① 소리의 고유성을 나타낸다.
② 다양한 악기의 소리를 서로 다르게 느끼는 까닭은 파동의 모양이 서로 다르기 때문이다.





2 소리의 표현 과정

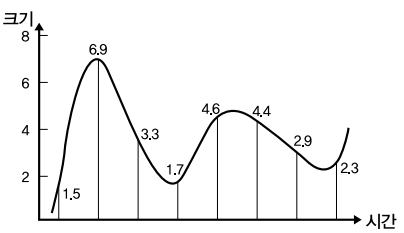
꿀팁

채널(channel)은 소리 파동을 표본화하는 장소의 개수를 뜻한다. 한 곳에서만 하는 1채널을 모노(mono)라 부르고, 두 곳에서 하는 2채널을 스테레오(stereo)라고 부른다. 최근에는 4채널, 5채널 등도 보편화되었다.

컴퓨팅 시스템에서 소리를 포함한 모든 데이터는 2진수로 저장된다. 일상의 소리를 2진수로 저장하기 위해서는 표본화(sampling), 양자화(quantization), 부호화(coding) 등의 절차를 순서대로 거쳐야 한다.

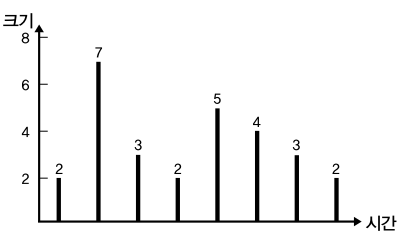
표본화

① 파동을 2진수로 변환하기 위하여 일정 시간 간격으로 진폭값을 추출하는 것이다.
② 표본화 시간 간격이 짧을수록 실제 소리에 가까운 고음질로 저장할 수 있으나 용량을 많이 차지한다.
③ 1초당 44,100회의 표본화(44.1kHz) 정도면, 사람이 들을 수 있는 대부분의 소리 대역을 지원할 수 있다.



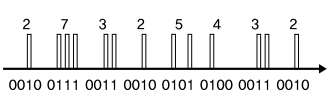
양자화

① 표본화한 각 점을 어느 정도의 정밀도로 표현할지 정하는 것이다.
② 4비트로 양자화하면 값을 16단계로 표현할 수 있지만, 8비트나 16비트로 양자화하면 각각 256, 65,536단계로 세밀한 표현이 가능하다.



부호화

표본화와 양자화를 거친 소리 정보를 2진수로 표현하고 저장하는 과정이다.



한 걸음 더 4, 8, 16비트로 표현 가능한 수의 가짓수

- 4비트로 표현 가능한 수의 가짓수: 2^4=16가지
- 8비트로 표현 가능한 수의 가짓수: 2^8=256가지
- 16비트로 표현 가능한 수의 가짓수: 2^{16}=65,536가지

4 동영상의 표현

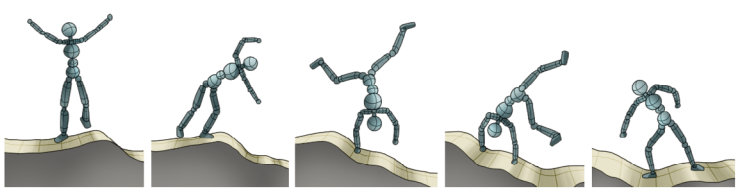
동영상은 여러 장의 그림과 소리를 저장한 것으로, 이를 연속해서 보여 주면 실제로 움직이는 것처럼 느끼게 된다. 많은 수의 그림과 소리를 담고 있어 다양한 방식으로 압축되므로 재생을 위해서는 빠른 중앙 처리 장치뿐만 아니라 큰 용량의 기억 장치와 저장 장치 등이 필요하다.

1 동영상의 구성 요소

동영상은 프레임(frame), 소리, 컨테이너, 코덱(CODEC: Coder/Decoder) 정보 등으로 구성된다.

프레임

① 동영상에 들어가는 직사각형의 그림이다.
② 프레임의 크기는 직사각형의 너비와 높이로 표현하며, 이를 해상도(resolution)라 한다.
예 320×240 해상도: 너비 320화소, 높이 240화소의 총 76,800개 화소로 이루어진 프레임이다.
③ 해상도와 1초당 사용되는 프레임 수(FPS: Frames Per Second)가 높을수록 자연스러워지나 용량이 커진다.
④ 1280×720 이하의 해상도와 24~30FPS가 많이 사용되었으나 최근에는 3840×2160 이상의 해상도와 60FPS 이상도 보편화되는 흐름이다.
⑤ 프레임을 압축할 경우에는 이전 프레임과의 차이점만 저장된다.



한 걸음 더 해상도의 약칭

일반적으로 사용하는 해상도는 짧게 줄여 부르는 약칭이 존재한다.

약칭	해상도	
HD(High Definition)	1280×720. 720p라고도 한다.	
FHD(Full HD)	1920×1080. 1080p라고도 한다.	
QHD(Quad HD)	2560×1440	
UHD(Ultra HD)	4K UHD	3840×2160
	8K UHD	7680×4320

소리

보통 압축하여 저장한다.

컨테이너

① 프레임과 소리를 담고 있는 틀이다.
② 프레임과 소리를 과자에 비유할 때, 과자를 담는 포장 방식이 여러 가지가 있듯이 컨테이너의 종류도 다양하다.
③ 보통 동영상 파일의 확장명으로 컨테이너의 종류를 확인한다.

꿀팁

많은 사람이 사용하는 컨테이너로는 MP4, MKV, AVI, MOV, WMV, WebM 등이 있다.

코덱 정보

① 동영상 내 프레임과 소리는 압축된 경우가 많은데, 코덱은 압축하거나 압축을 해제하는 하드웨어나 소프트웨어를 뜻한다.
② 다양한 압축 방식이 존재하는 만큼 여러 가지 코덱이 존재한다.
③ 영상 편집 프로그램이나 영상 재생 프로그램은 코덱을 내장하거나 가져다 사용할 수 있도록 구성된다.
④ 동영상에는 프레임과 소리의 압축에 사용된 코덱 정보가 수록된다.

꿀팁

많은 사람이 사용하는 코덱으로는 H.264, VP8, DivX 등(이상 프레임)과 MP3, AAC, Vorbis, FLAC, AC3 등(이상 소리)이 있다.

꿀팁

컨테이너와 코덱은 서로 독립적이다. 예를 들어 MP4, MKV, AVI 컨테이너 모두 H.264 코덱으로 압축된 프레임을 담을 수 있다.

5 디지털 정보의 제작

컴퓨터에서 사용 가능한 그림, 소리, 동영상 등을 운영 체제에 내장된 프로그램을 이용하여 제작해 보자.

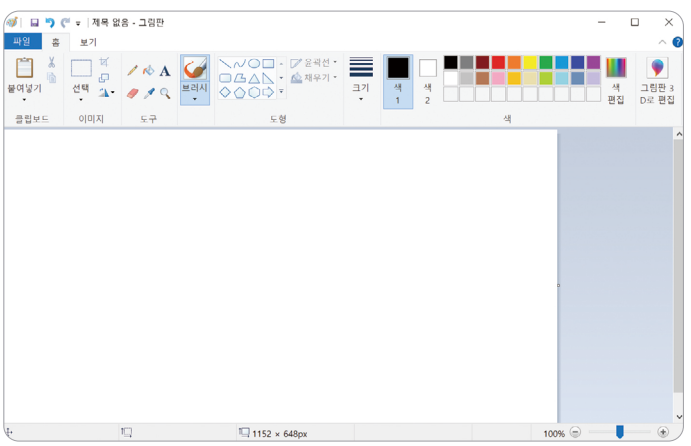
무료 공개 소프트웨어를 이용해도 돼!

1 그림 파일 제작

개인용으로 가장 널리 사용하는 윈도 운영 체제에 내장된 '그림판' 프로그램을 이용하여 그림 파일을 새로 만들거나 기존 그림 파일을 편집한다.

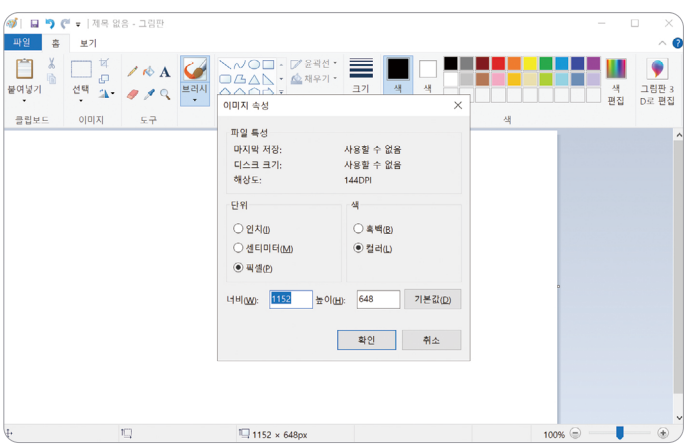
1. 그림 파일 새로 만들기

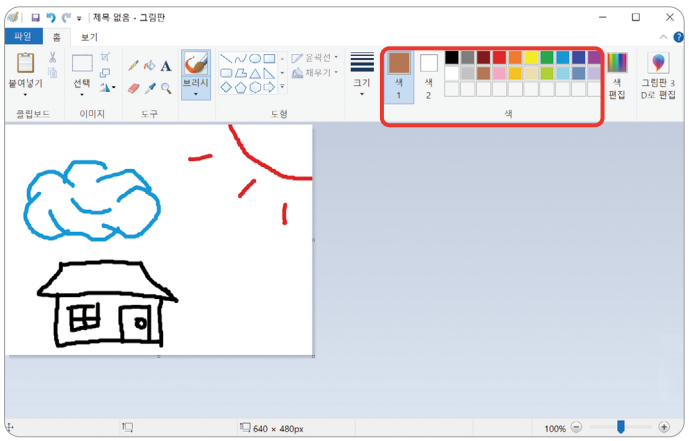
① 윈도 키를 누르고 '그림판'을 검색하여 그림판을 실행한다.



② 기본적으로 새로운 그림을 만드는 상태에 진입한다. 아래 상태 표시줄에 그림의 크기, 즉 화소 수가 너비와 높이로 표시되며, 이를 변경하려면 [파일]–[속성]을 선택한다.

※ 윈도 11의 경우 [파일]–[이미지 속성]을 선택한다.

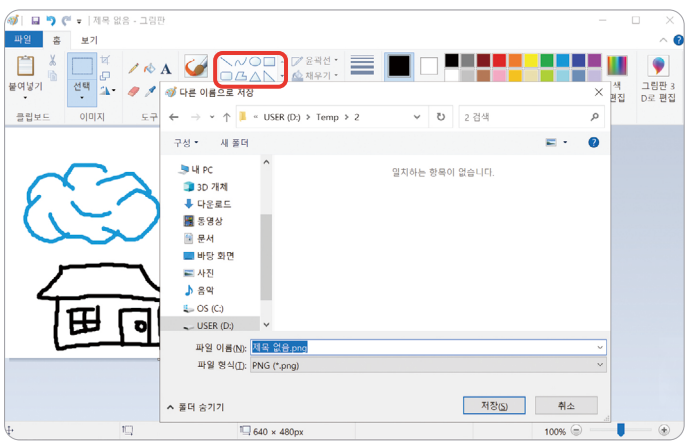




③ 연필 버튼을 클릭한 후 그림에 대고 마우스 버튼을 누르면서 움직이면, 연필로 종이에 그림을 그리듯 그림을 그릴 수 있다.

④ 브러시 버튼과 크기 버튼을 클릭하여 선의 모양과 굵기를 바꿀 수 있으며, 색 도구를 통하여 선의 색을 변경할 수도 있다. 색 1과 색 2는 각각 마우스의 왼쪽, 오른쪽 버튼을 눌렀을 때 나오는 색을 나타낸다.

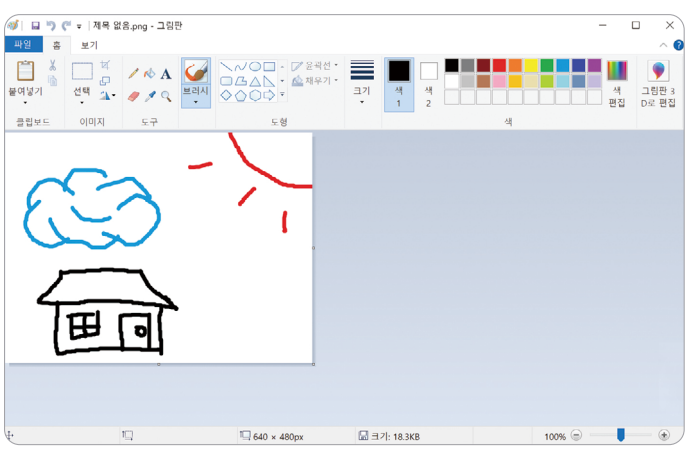
※ 윈도 11의 경우 '색 1'과 '색 2' 글자가 보이지 않는다.



⑤ 색 채우기 버튼, 텍스트 버튼, 도형 도구 등을 직접 사용하여 기능을 알아본다.

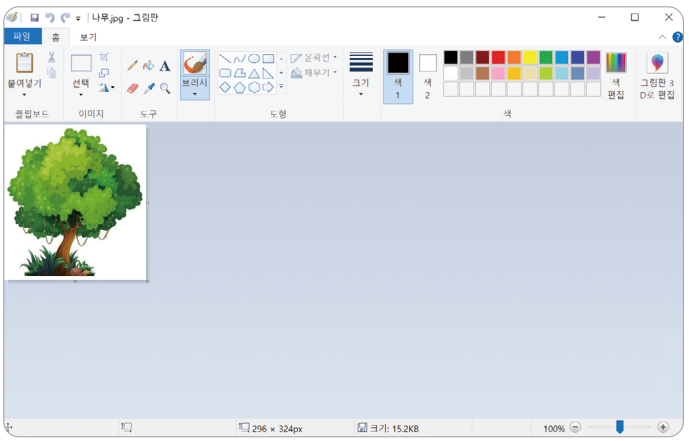
⑥ 왼쪽 위의 저장 버튼을 클릭하여 파일로 저장한다. 사진이나 복잡한 그림이라면 압축률이 좋은 JPG를 선택하고, 단순한 그림이라면 무손실 압축 방식인 PNG를 선택하는 것이 좋다.

2. 기존 그림 파일 편집하기

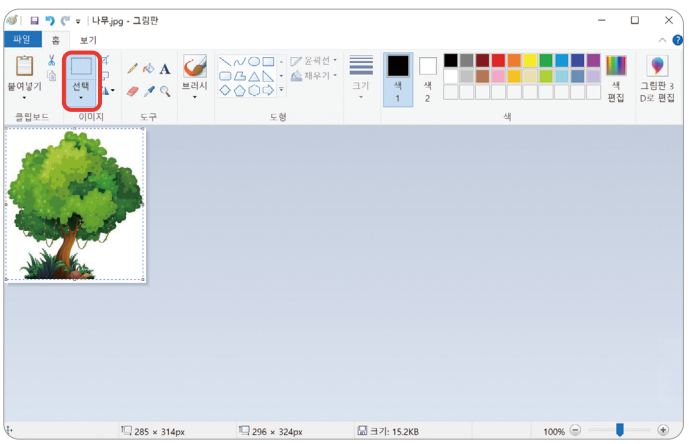


① 배경 그림으로 사용할 이전에 저장한 그림 파일을 [파일]-[열기] 메뉴를 선택하여 불러온다.

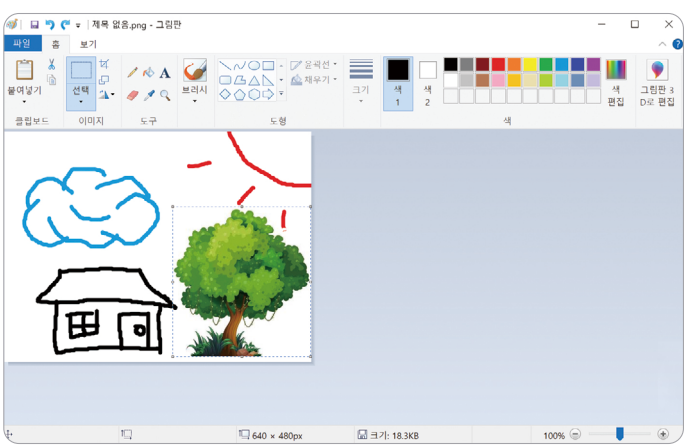
② 배경 그림에 합성할 그림 파일을 인터넷에서 내려받은 후 그림판 프로그램을 새로 하나 더 실행하여 불러온다. 저작권이 없는 무료 그림을 검색하여 사용한다.



③ 선택 버튼을 클릭하고 원하는 영역만 선택한 후 Ctrl+C 키를 눌러 클립보드에 복사한다.



④ 배경 그림으로 와서 Ctrl+V 키를 눌러 붙여 넣는다. 배경과 자연스럽게 배치될 수 있도록 마우스로 위치와 크기를 적절하게 조정한다. 선택 버튼 클릭 후 '선택 영역 투명하게'를 선택하면 그림의 전경 부분만을 판단하여 붙여 넣을 수 있다.



스스로 해 보기

그림판 기능 사용하기

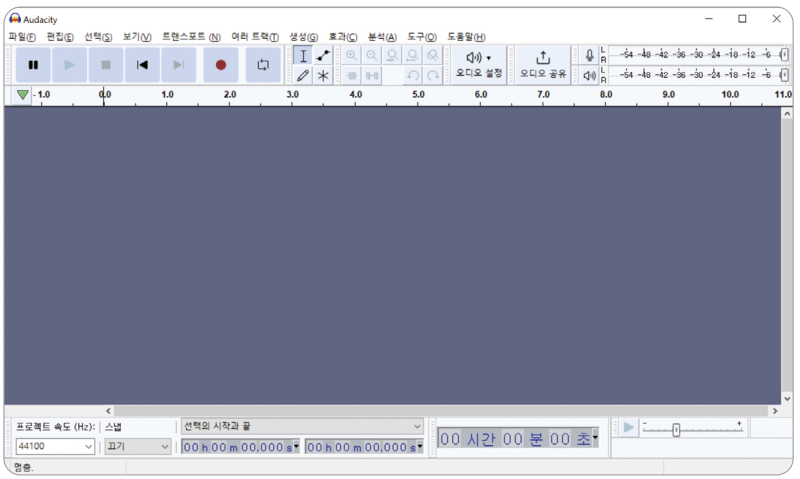
색 채우기, 텍스트 기능을 사용해 보고, PNG 외에 GIF, JPG, BMP 등의 파일 형식으로도 저장해 보자.

2 소리 파일 제작

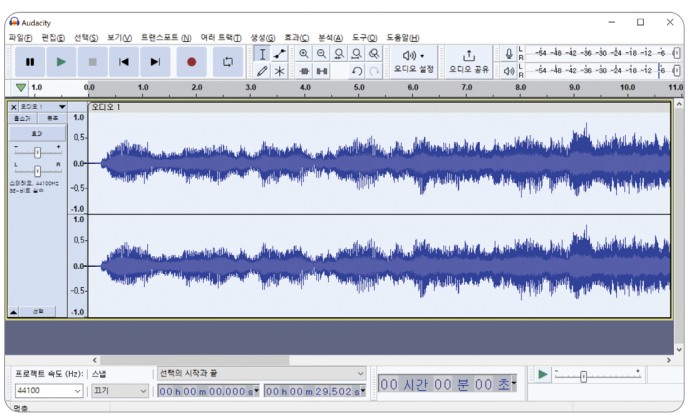
꿀팁

Audacity는 소스 코드가 공개된 공개 소스 프로그램으로 윈도, 맥, 리눅스에서 모두 사용이 가능하다.

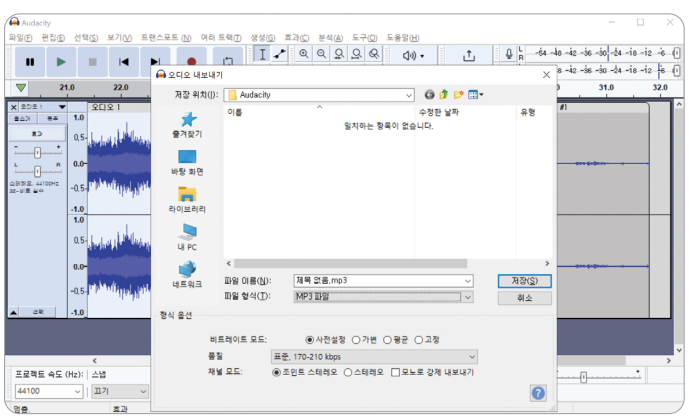
윈도 운영 체제에 내장된 소리 편집 프로그램이 없으므로 인터넷에서 무료 프로그램인 Audacity를 내려받아 사용한다.



1. 주변 소리를 직접 녹음하여 새로운 소리 파일 제작하기



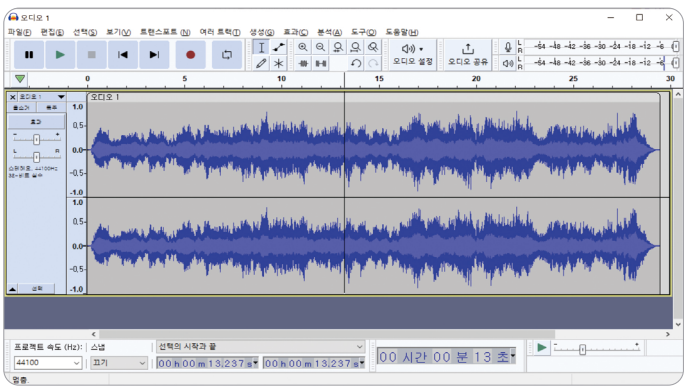
① ● 버튼을 눌러 녹음을 시작한 후 ■ 버튼을 눌러 녹음을 끝낸다.



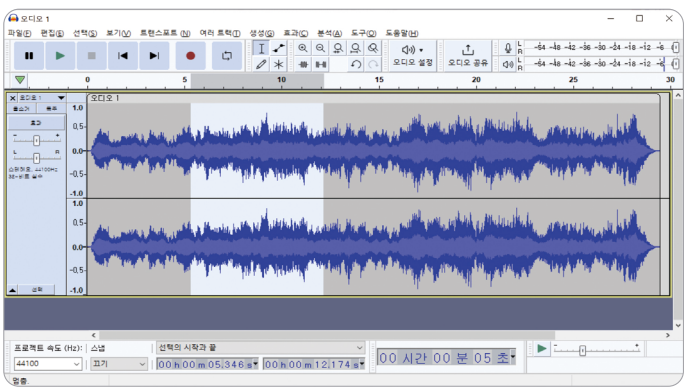
② [파일]–[내보내기]를 선택하여 MP3, WAV, OGG, FLAC 등의 소리 파일로 저장할 수 있다.

2. 기존 소리 파일을 편집하여 새로운 소리 파일 제작하기

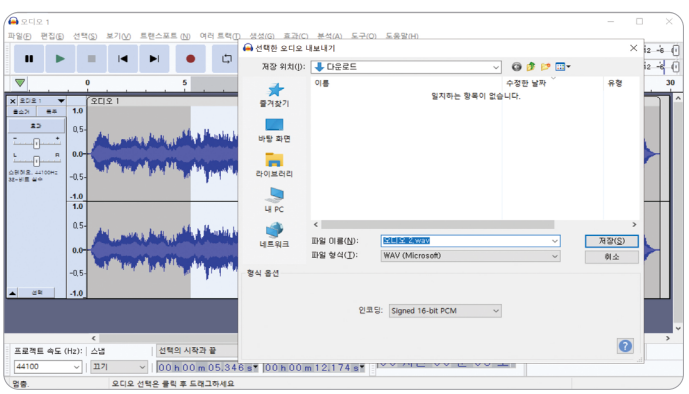
① [파일]–[열기]를 선택하여 앞서 녹음해서 저장한 소리 파일을 불러온다.



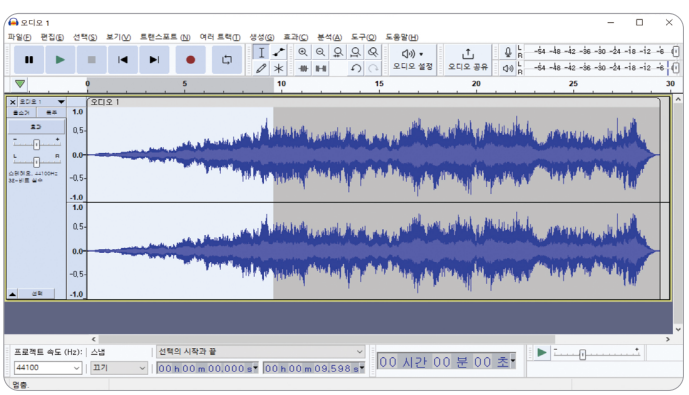
② 마우스 버튼을 누른 상태에서 드래그하여 임의의 범위를 선택하고 Space 키 또는 ▶ 버튼을 눌러 재생한다.



③ [파일]–[내보내기]–[선택한 오디오 내보내기]를 통하여 선택된 범위의 소리만을 별도의 파일로 저장할 수 있다.



④ 소리의 앞부분을 선택한 후 [효과]–[페이딩]–[페이드 인(등장 효과)]을 선택하여 소리 앞부분의 크기가 자연스럽게 커지도록 효과를 적용한다.



스스로 해 보기 소리 제거하기

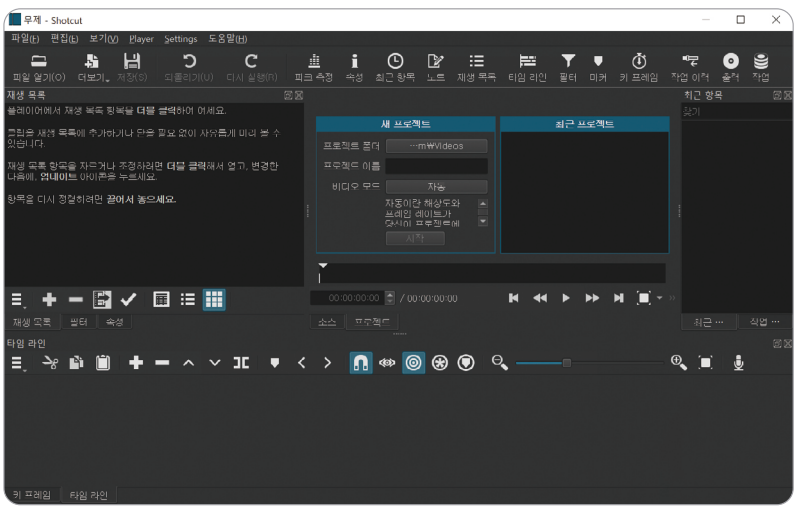
[편집]–[삭제]를 선택하여 소리의 뒷부분 1초를 제거해 보자.

3 동영상 파일 제작

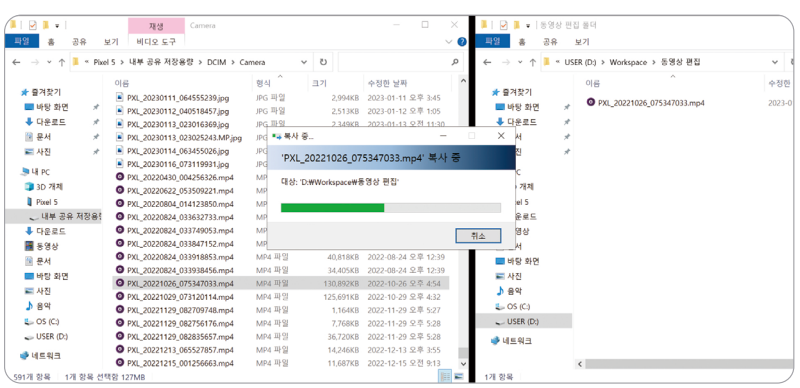
꿀팁

Shotcut은 소스 코드가 공개된 공개 소스 프로그램으로 윈도, 맥, 리눅스에서 모두 사용이 가능하다.

인터넷에서 무료 프로그램인 Shotcut을 내려받아 기존의 동영상 파일로부터 원하는 부분만을 자른 후 전환 효과, 자막 등을 추가하여 새로운 동영상 파일로 저장해 보자.

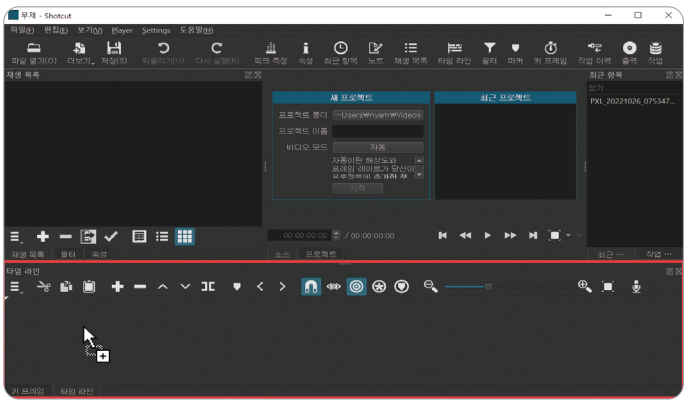


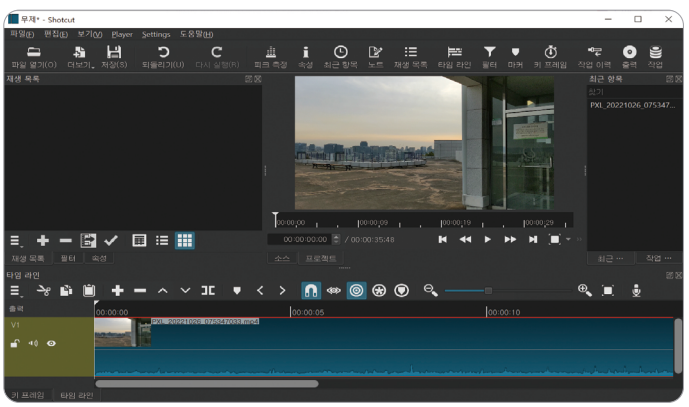
1. 동영상 파일 편집하기



① 스마트폰, 디지털 캠코더, 웹캠 등으로 동영상을 촬영한 후, 해당 파일을 컴퓨터로 복사한다.

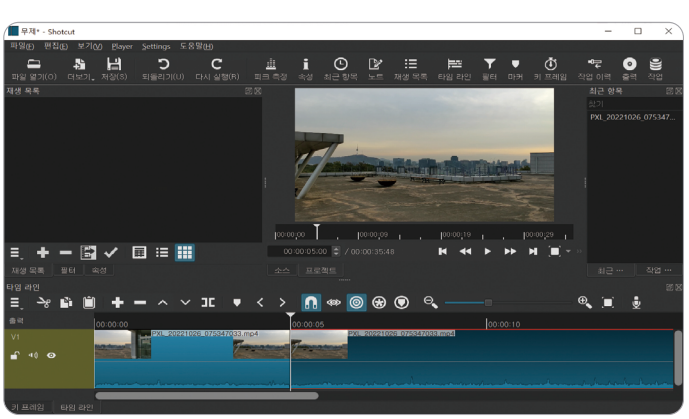
② Shotcut 프로그램을 실행하여 타임라인 부분으로 동영상 파일을 드래그 앤 드롭한다. Shotcut은 대부분의 동영상 편집 프로그램과 마찬가지로 시간의 흐름을 나타내는 타임라인상에 여러 영상, 텍스트(자막), 소리 등을 추가하여 편집한 후 새로운 동영상 파일로 저장할 수 있다.

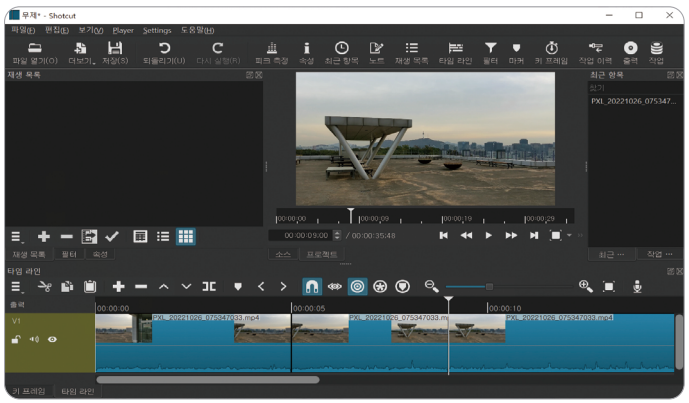




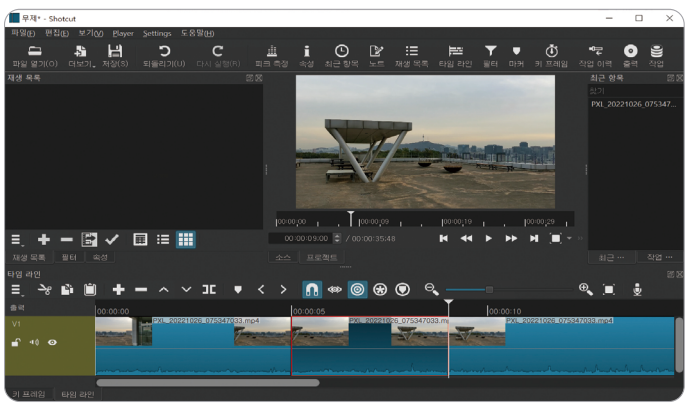
③ 중앙에는 편집 결과 영상을 확인할 수 있는 미리보기가 있으며, Space 키를 눌러 재생할 수 있다. 여기에서 사용한 영상은 35.48초라는 것이 미리 보기 영상 아래에서 확인된다. 이 영상에서 5초부터 9초 사이를 삭제한다. 아래쪽 끝 타임라인상 위쪽에는 시간을 나타내는 영역이 있는데, 여기서 00:00:05 부근을 클릭한 후, 'S' 키를 눌러 영상을 5초 이전까지의 영상과 5초 이후의 영상으로 분리한다.

④ 타임라인상에서 5초 이전과 5초 이후를 각각 클릭하면, 각기 다른 부분으로 나뉘어 있음을 확인할 수 있다.

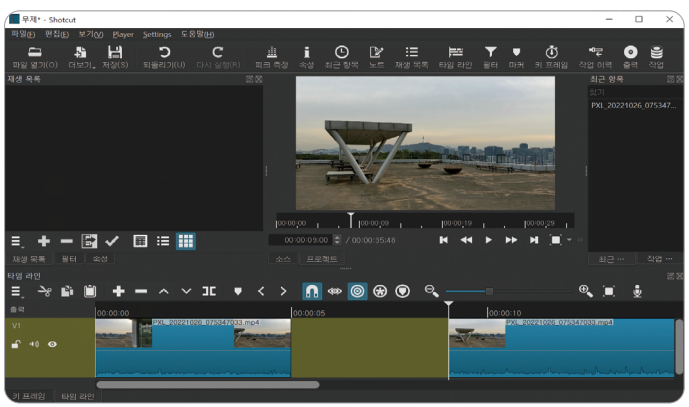




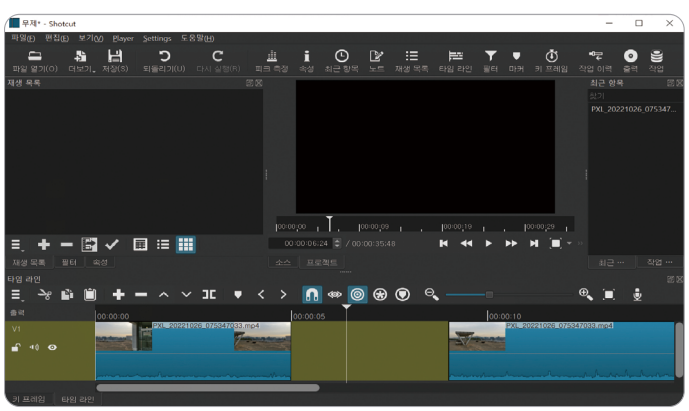
⑤ 앞서 실행한 것처럼 타임라인상의 시간 영역에서 9초 부근을 클릭한 후, 'S' 키를 눌러 영상을 9초 전까지의 영상과 9초 이후의 영상으로 분리하면, 맨 나중에는 3개의 부분으로 나뉘는 것을 확인할 수 있다.



⑥ 5초부터 9초 사이의 부분을 클릭하여 선택한다.

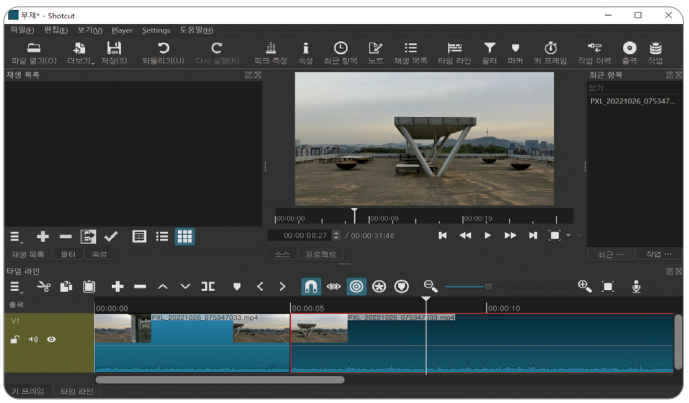


⑦ 'Del' 키를 눌러 해당 부분을 삭제한다.

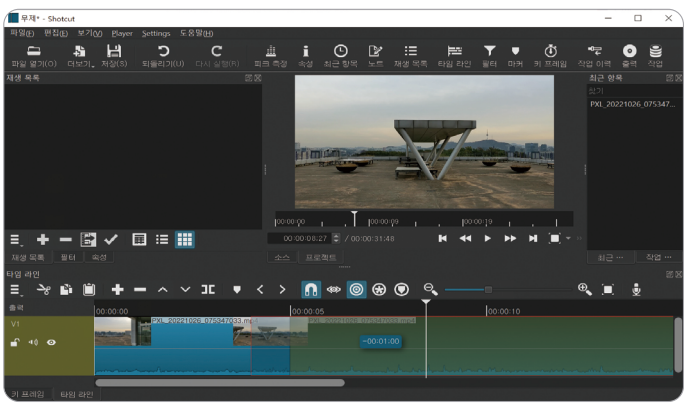


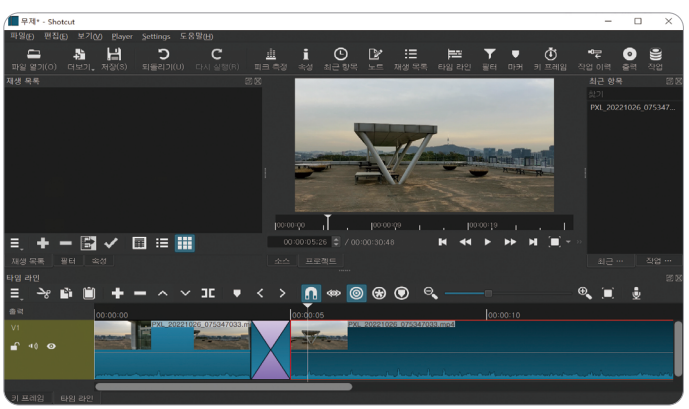
⑧ 'Home' 키를 눌러 영상의 시작 시간인 00:00:00으로 이동 후, Space 키를 눌러 처음부터 재생하면, 잘린 5초부터 9초 사이는 검게 표시되는 것을 확인할 수 있다.

⑨ 9초 이후의 영역을 드래그하여 5초 바로 뒤로 붙인다.

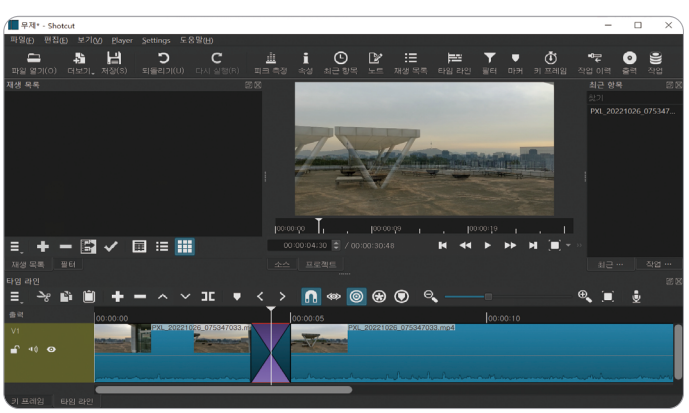


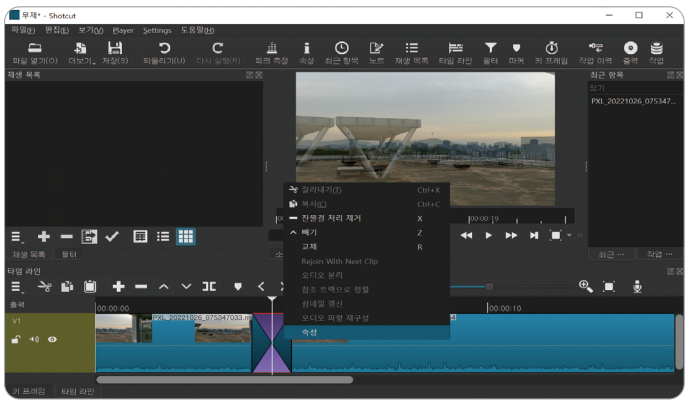
ⓐ ‘Home’ 키를 눌러 영상의 시작 시간인 00:00:00으로 간 후, ‘Space’ 키를 눌러 처음부터 재생하면 검은 공백은 없어졌으나 영상이 갑자기 바뀌어서 자연스럽지 못하다. 5초 이후의 뒷부분 영상을 다시 드래그하여 5초 이전의 앞부분 영상의 일부와 1초가량 겹치게 한다.



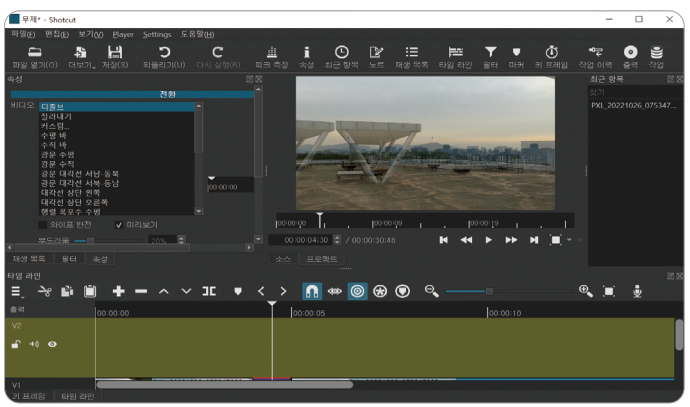


ⓑ ‘Home’ 키를 눌러 영상의 시작 시간인 00:00:00으로 간 후 ‘Space’ 키를 눌러 처음부터 재생하면, 1초가량의 전환 효과가 적용된 것을 확인할 수 있다.

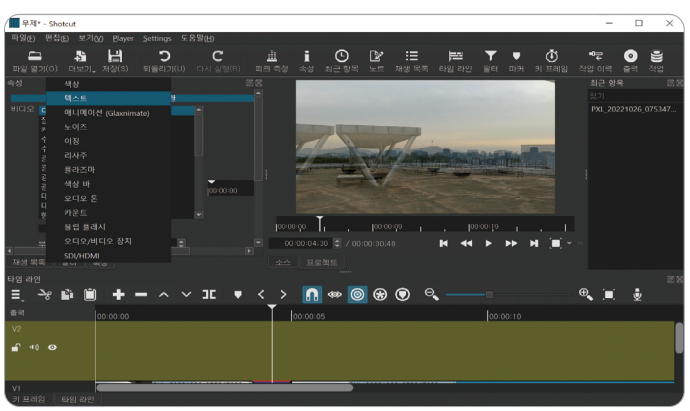




ⓒ 영상 전환 부분에 대고 마우스 오른쪽 버튼을 누른 후 '속성'을 선택하면, 현재 '디졸브' 전환 효과가 적용되어 있음을 확인할 수 있으며 이를 변경할 수 있다.



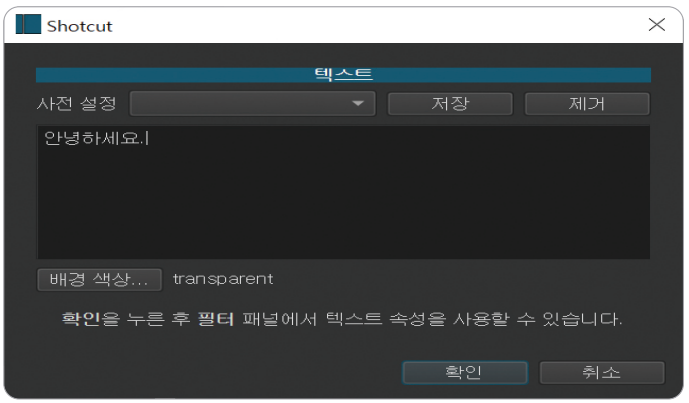
ⓓ 영상에 자막을 넣는다. 이를 위해서는 자막을 위한 별도의 새로운 트랙을 추가해야 한다. 타임라인상의 왼쪽 위 끝에 있는  아이콘을 클릭한 후 [트랙 작업]–[트랙 삽입]을 선택하여 새로운 트랙을 삽입한다.



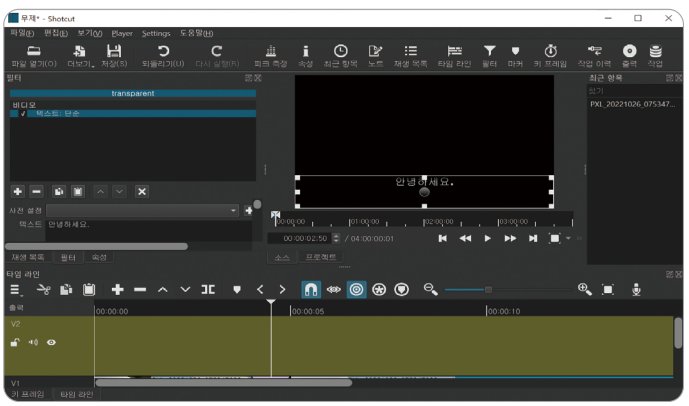
ⓔ 툴바의 

버튼을 클릭한 후 '텍스트'를 선택한다.

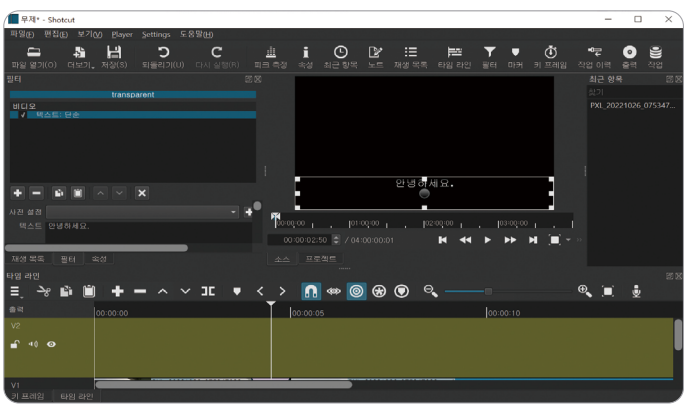
ⓕ 텍스트를 입력할 수 있는 대화 상자가 나오면, 여기에 원하는 자막 텍스트를 입력한 후 '열기' 버튼을 누른다.

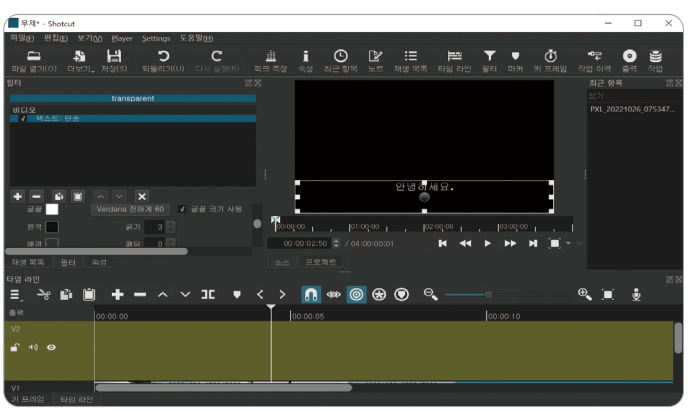


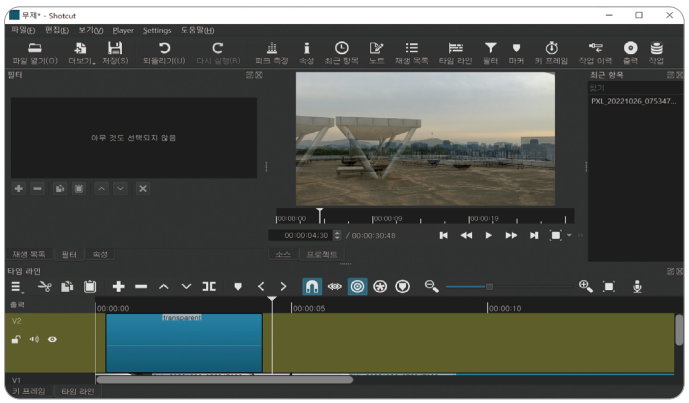
ⓖ 입력한 자막 텍스트가 미리보기 영역에 나타나며 위치와 크기를 조정할 수 있다.



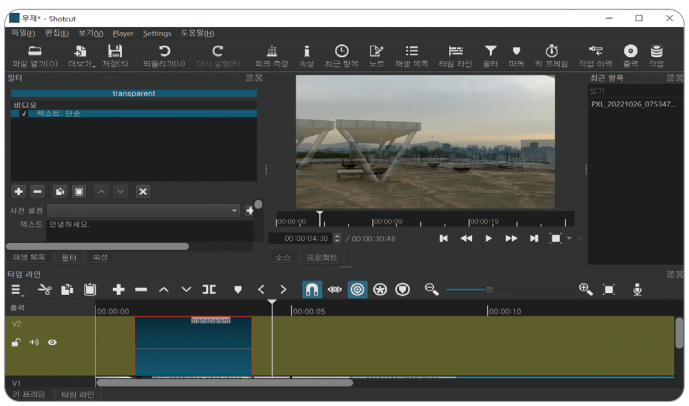
ⓗ 왼쪽 가운데 부근의 인터페이스를 스크롤한 후 글꼴과 색깔을 변경할 수 있다.



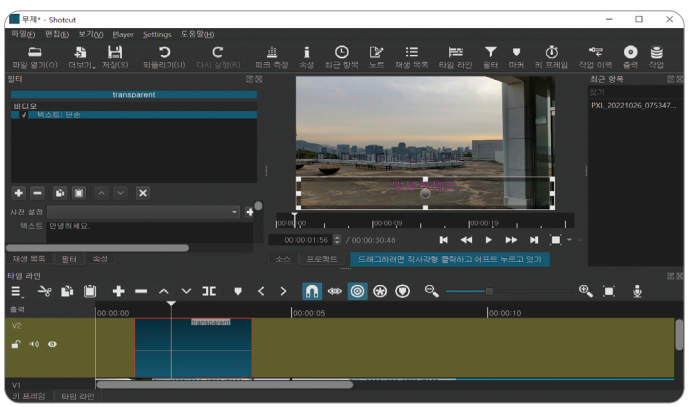




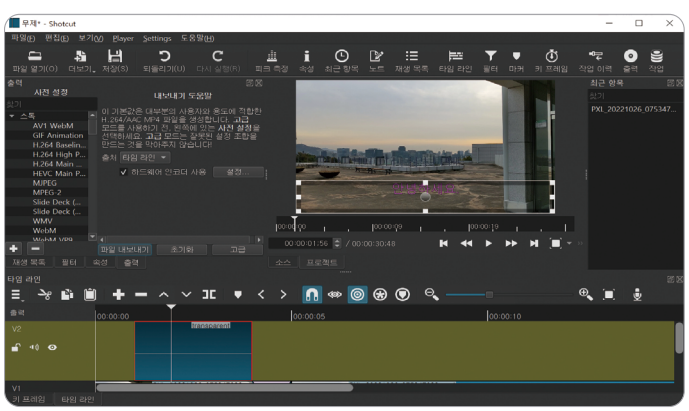
ⓘ 원하는 모양의 자막 텍스트가 완성되었다면, 미리 보기 화면을 앞서 새로 만든 타임라인상의 자막 트랙으로 드래그 앤 드롭한다.



ⓙ 트랙 내에서의 위치와 크기를 조정함으로써 자막이 나타나는 시간과 사라지는 시간을 변경할 수 있다. 여기에서는 영상의 1초부터 4초까지 나타나도록 하였다.



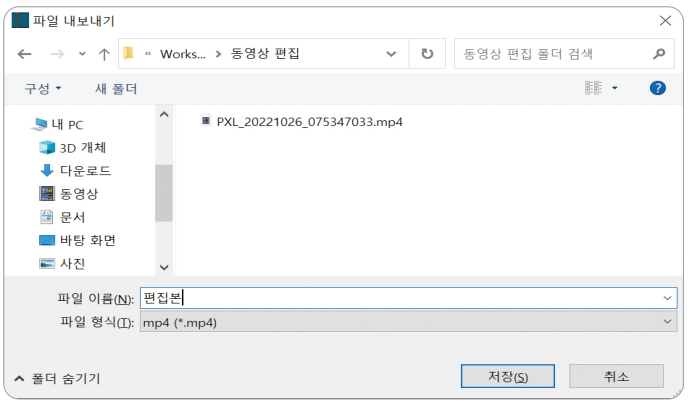
ⓚ 'Home' 키를 눌러 영상의 시작 시간인 00:00:00으로 간 후, 'Space' 키를 눌러 처음부터 재생함으로써 자막이 정상적으로 출력되는지 확인한다.



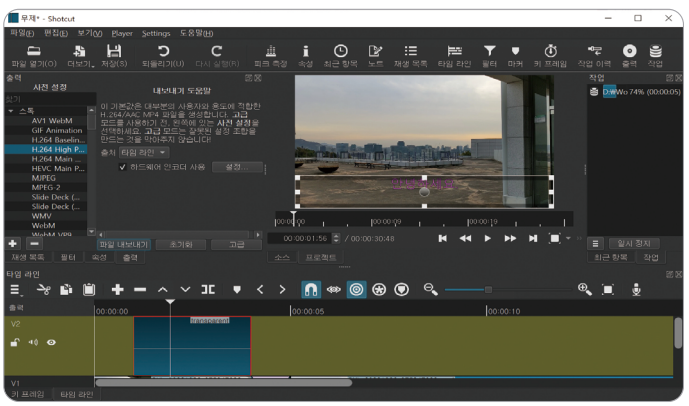
ⓛ 지금까지의 작업물을 동영상 파일로 저장한다. 툴바의 

버튼을 클릭하면 왼쪽에 '출력' 영역이 표시된다.

ⓜ 동영상 파일 형식(컨테이너)과 압축 코덱을 직접 지정할 수 있으나, 자주 사용되는 조합을 왼쪽의 '사전 설정'에서 선택할 수도 있다. 여기서는 고화질 동영상에 적합한 H.264 High Profile을 선택한 후 '파일 내보내기' 버튼을 클릭한다.



ⓝ 오른쪽에 '작업' 영역이 생기며 동영상 파일이 저장되는 진행 상황을 확인할 수 있다.



ⓞ 저장이 완료된 동영상 파일을 재생해서 편집 내용이 모두 적용되었는지 확인한다.

스스로 해 보기

영상에 배경 음악 삽입하기

타임라인 메뉴에서 [트랙 작업]–[오디오 트랙 추가]를 선택한 후 wav, mp3, ogg, flac 등의 소리 파일을 드래그하여 넣어 보자.

컴퓨팅 읽기 자료

Smart Reading

바코드와 QR 코드

바코드

- 바코드는 1948년 미국에서 발명된 것으로, 숫자나 문자를 굵기가 다른 흑백 막대들을 조합하여 표현한 코드이다. 컴퓨터에 연결된 리더기가 바코드의 흑백 막대를 구분하여 숫자를 빠르게 인식하는데, 모든 국가의 코드 체계가 표준화되어 있어 제조업, 물류 관리 등에서 널리 사용한다. 우리나라는 1988년 국제 코드 관리 기관에 가입하면서 처음 사용하기 시작하였다.
- 여러 규격의 바코드 중 열세 자리 숫자를 표현한 EAN-13 바코드가 가장 많이 사용된다. 열세 자리의 고유 숫자는 신분증과 같은 역할을 하는데, 국가 번호, 업체 번호, 상품 번호, 검증 번호 등으로 구성된다.
- 바코드의 구성 요소

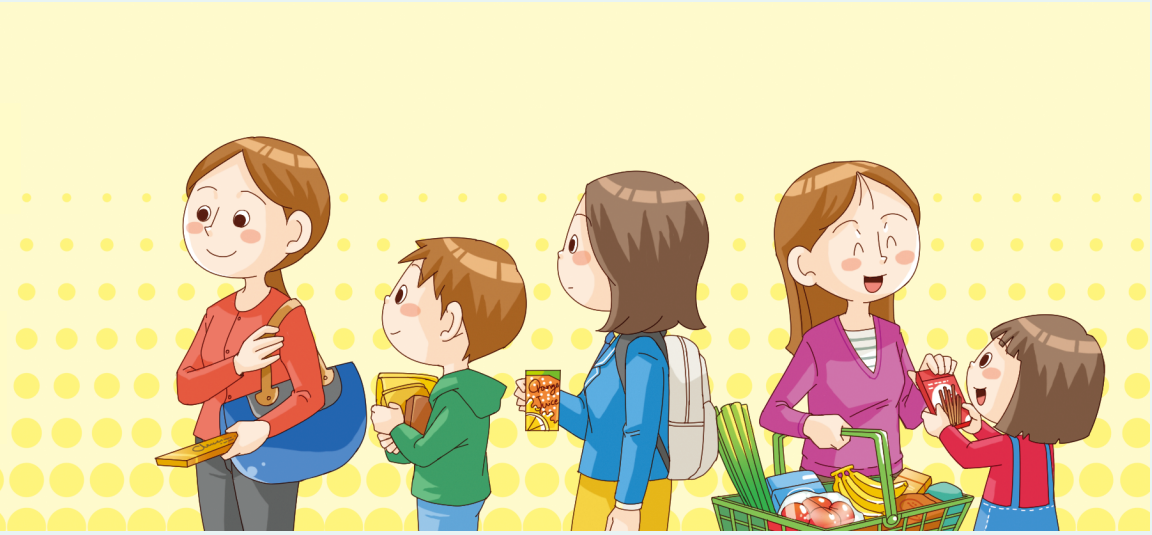
국가 번호	국가
00*~13*	미국, 캐나다
45*, 49*	일본
50*	영국
69*	중국
880	대한민국

국가 번호 국제 코드 관리 기관에서 할당한 세 자리 숫자로 구성되며, 물품 생산 및 등록 국가를 나타낸다. 우리나라의 경우 서울 올림픽을 개최한 1988년도에 바코드를 도입하여 '880'이라는 상징적인 숫자를 부여받았다.

업체 번호 생산자 또는 판매자를 나타내는데, 우리나라에서는 유통물류진흥원에서 번호를 부여한다.

상품 번호 업체가 독자적 기준에 따라 물품에 직접 부여하는 번호이다.

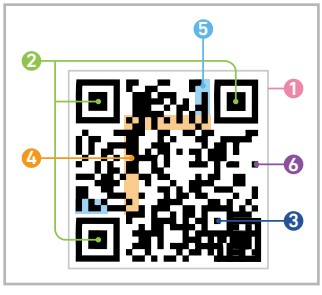
검증 번호 국가 번호, 업체 번호, 상품 번호를 특정 계산식에 넣어 계산한 숫자와 비교함으로써 오류 발생 여부를 판단하는 데 사용한다.



QR 코드

• 1994년 일본에서 발명된 것으로, 숫자나 문자를 흑백 2차원 격자무늬 패턴으로 표현한 코드이다. 2차원 패턴으로 종횡의 정보를 가질 수 있어 종래 많이 사용하던 바코드의 용량 제한을 극복할 수 있다. 문자는 최대 4,296자, 한자는 최대 1,817자, 숫자는 최대 7,089자를 기록할 수 있어 글뿐만 아니라 사진, 인터넷 주소(URL), 지도, 명함 등의 정보도 담을 수 있다.

• QR 코드의 구성 요소



❶ 경계: 다른 이미지와 구분해 주는 공간
❷ 파인더: QR 코드임을 알 수 있게 해 주는 세 개의 정사각형. 코드 방향을 잡아 준다.
❸ 얼라인먼트: 코드 스캔 방향을 일러 주는 정사각형. 파인더의 정사각형보다 작다. 코드가 약간 훼손되었을 때도 정보를 읽어 낼 수 있다.
❹ 타이밍 패턴: 코드 내 정보가 담긴 개별 단위를 인지할 수 있도록 해 준다.
❺ 버전 정보: 해당 QR 코드의 버전을 나타낸다.
❻ 개별 단위: ❶~❺의 요소를 제외한 '희고 검으며 작은' 사각형들. 각각 데이터를 담은 단위(cell) 역할을 한다.

프로젝트 활동

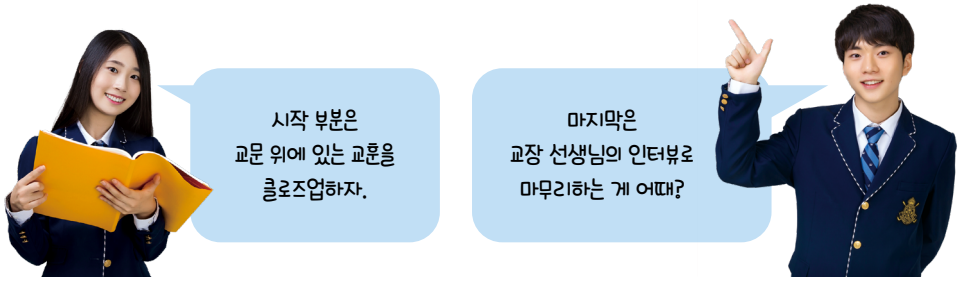
학교 소개 동영상 제작하기

서연이는 자신이 다니는 학교에 대한 자부심이 남다르다. 개교 30주년을 맞아 서연이는 자랑스러운 학교를 더욱 널리 알리고 싶어 반 친구들과 함께 학교 소개 동영상을 제작하기로 하였다.

○ 수행 순서

1. 회의를 통하여 제작할 동영상의 내용과 길이를 정한다.

2. 동영상에 대한 스토리보드를 구상한다.



3. 동영상에 들어갈 자료를 직접 검색하거나 촬영하여 수집한다.

4. 영상에 들어갈 배경 음악을 수집한다. 저작권이 있는 음악을 무단으로 사용할 경우, 저작권법을 어길 수 있으므로 저작권이 없는 음악을 찾아서 수집한다.

참고 사이트 https://www.youtube.com/c/NoCopyrightSounds

5. 동영상 편집 프로그램으로 수집된 사진, 영상 및 배경 음악을 적용하여 학교 소개 동영상 파일을 만든다.

6. 완성한 학교 소개 동영상을 반 친구들 앞에서 발표한다.

1 제작한 학교 소개 동영상을 소셜 네트워크 서비스(SNS)나 동영상 공유 웹사이트 등에 올려 보자.

대단원 마무리 정리

01. 수의 체계

1 자료(data): 현실 세계에서 측정하고 수집한 사실이나 값

2 정보(information): 어떠한 목적이나 의도에 맞게 자료를 가공 처리한 것

3 정보 처리: 자료로부터 정보를 만들 때 다양한 방법으로 가공하는 절차

4 진법: 수를 표현하는 다양한 방법의 하나로, 숫자의 위치에 따라 다른 가중치로 수를 나타내는 방법

5 진수: 진법을 사용하여 표현된 수

10진법	우리가 일상적으로 사용하는 진법으로, 0부터 9까지 열 가지로 구성된 아라비아 숫자로 수를 표현
2진법	컴퓨팅 시스템에서 수를 다루기 위하여 사용하는 진법으로, 0과 1 두 가지 숫자를 사용하여 수를 표현
8진법	0부터 7까지의 숫자를 사용하여 수를 표현하는 진법으로, 리눅스 운영 체제의 파일 권한 표시 등 특정 상황에서 종종 사용
16진법	0부터 9까지의 숫자와 A부터 F까지의 알파벳을 사용하여 수를 표현하는 진법으로, 컴퓨터 내부에서 사용되는 2진수를 사람에게 출력해 보여 줄 때 많이 사용

6 진법 변환

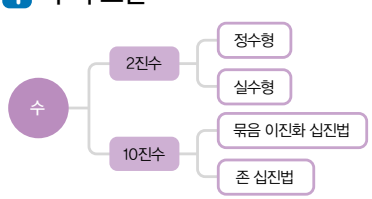
① **다른 진수를 10진수로 변환:** 2진수, 16진수 등 10진수가 아닌 다른 진수를 10진수로 변환할 때는 각 자릿수에 가중치를 곱한 값들을 모두 더한다.

② **10진수를 다른 진수로 변환:** 10진수를 2진수나 16진수 등 다른 진수로 변환할 때는 몫이 0이 될 때까지 변환하려는 진수의 밑수로 계속 나눈 후 나머지를 역순으로 적는다.

③ **2진수와 16진수 간의 변환:** 2진수 네 자리가 16진수 한 자리에 해당하므로 네 자리씩 묶어서 변환한다.

02. 디지털 정보의 연산

1 수의 표현



2 2진수

① **정수형**

- 최상위 한 개의 비트가 부호를, 나머지 비트들이 절댓값을 나타낸다.

❶ 현실 세계에서 측정하고 수집한 사실이나 값을 ㅈㄹ(이)라고 한다.

❷ 어떠한 목적이나 의도에 맞게 자료를 가공 처리한 것을 ㅈㅂ(이)라고 한다.

❸ 2진수에서 음수는 2의 ㅂㅅ 방식으로 나타낸다.

❹ 특정 문자에 대응되는 2진수를 부여해 놓은 체계를 문자 ㅇㅋㄷ(이)라고 한다.

❺ 미국표준화협회가 7비트에 95개의 출력 가능한 문자와 33개의 제어 문자를 배치한 문자 인코딩을 ㅇㅅㅋㅋㄷ(이)라고 한다.

❻ 전 세계의 모든 문자를 컴퓨터에서 일관되게 표현하고 다룰 수 있도록 설계한 산업 표준을 ㅇㄴㅋㄷ(이)라고 한다.

정답 ❶ 자료 ❷ 정보 ❸ 보수 ❹ 인코딩 ❺ 아스키코드 ❻ 유니코드

- 부호 비트가 1이면 음의 정수이고, 0이면 양의 정수이다.
- 수의 크기에 따라 1, 2, 4, 8바이트 등이 될 수 있으며, 음수는 부호화 절대치 방법, 1의 보수 방법, 2의 보수 방법 등으로 절댓값을 표현한다.
- 2의 보수 방법: 컴퓨터에서 널리 사용되는 음수 표현 방식으로서, 양숫값의 모든 비트를 반전시킨 후 1을 더한다.

② **실수형**

- 주로 부동 소수점 방식으로 표현한다.
- 정규화 작업을 수행한 후 부호, 지수, 소수점 이하 가수를 각각 저장한다.

③ **2진수의 덧셈**: 10진수의 덧셈과 마찬가지로 가장 아래 자릿수부터 더해 나가며, 10진수의 덧셈에서 10 이상이 되면 자리 올림하는 것처럼 2진수에서는 2 이상이 되면 자리 올림한다.

④ **2진수의 뺄셈**: 음의 정수, 즉, 2의 보숫값을 더하는 방식으로 처리한다.

03. 디지털 정보의 표현

1 문자의 표현

① **문자 인코딩**: 특정 문자에 대응되는 2진수를 부여해 놓은 체계

② **아스키코드(ASCII)**: 미국표준화협회가 7비트에 95개의 출력 가능한 문자와 33개의 제어 문자를 배치한 문자 인코딩

③ **유니코드(Unicode)**

- 전 세계의 모든 문자를 컴퓨터에서 일관되게 표현하고 다룰 수 있도록 설계한 산업 표준이다.
- 전 세계의 모든 문자를 겹치지 않게 일렬로 줄 세워 놓고 숫자를 부여한 개념이다.
- UTF-8 문자 인코딩이 가장 널리 사용된다.

2 그림, 소리, 동영상의 표현

① **비트맵 방식**: 색깔을 가진 화소들을 2차원으로 나열하여 그림을 표현하는 방식으로, 널리 사용되는 그림 파일 형식인 JPG, GIF, PNG, BMP 등에서 사용된다.

② **벡터 방식**: 점과 점 사이를 수학적 계산식을 이용하여 그림을 표현하는 방식으로, AI, CDR, SVG 등의 그림 파일 형식에서 사용된다.

③ **소리의 표현**: 소리의 근원인 파동에 대하여 표본화, 양자화, 부호화 작업을 거쳐 2진수로 저장한다.

④ **동영상의 표현**: 프레임, 소리, 컨테이너, 코덱 정보 등으로 구성된다.

대단원 마무리 평가

선택형

01 ㉠과 ㉡에 들어갈 용어로 올바른 것은?

> ㉠ : 현실 세계에서 측정하고 수집한 사실이나 값
> ㉡ : 어떠한 목적이나 의도에 맞게 ㉠을(를) 가공 처리한 것이나 값

	㉠	㉡
①	소재	지식
②	지식	지혜
③	상식	개념
④	자료	정보
⑤	인식	의식

02 기억 장치에 자료를 저장하거나 연산할 때 다루는 단위로서 8개의 비트를 묶은 것은?

① 워드
② 니블
③ 헥사
④ 옥타
⑤ 바이트

03 10진수 20을 16진수로 나타낸 것은?

① $20_{(16)}$
② $14_{(16)}$
③ $16_{(20)}$
④ $10_{(16)}$
⑤ $10100_{(2)}$

04 16진수 $5C_{(16)}$를 2진수로 나타낸 것은?

① $1011100_{(2)}$
② $1010110_{(2)}$
③ $1101100_{(2)}$
④ $1010001_{(2)}$
⑤ $1100101_{(2)}$

05 8비트로 표현한 2진수 $00101101_{(2)}$을 2의 보수로 나타낸 것은?

① $10101101_{(2)}$
② $11010010_{(2)}$
③ $11010011_{(2)}$
④ $00101110_{(2)}$
⑤ $00101101_{(2)}$

06 컴퓨팅 시스템이 일상에서의 소리를 2진수로 저장하기 위하여 수행하는 절차를 순서대로 나열한 것은?

① 표본화, 부호화, 양자화
② 부호화, 표본화, 양자화
③ 부호화, 양자화, 표본화
④ 표본화, 양자화, 부호화
⑤ 양자화, 표본화, 부호화

단답형

07 특정 문자에 대응되는 2진수를 부여해 놓은 체계를 무엇이라 하는가?

( )

08 다음 2진수 연산의 결과를 써 보자.

(1) $110_{(2)} + 1011_{(2)}$ ( )

(2) $11010_{(2)} \times 101_{(2)}$ ( )

09 다음은 무엇에 대한 설명인가?

- 전 세계의 모든 문자를 컴퓨터에서 일관되게 표현하고 다룰 수 있도록 설계된 산업 표준
- 전 세계의 모든 문자를 겹치지 않게 일렬로 줄 세워 놓고 숫자를 부여한 개념
- UTF-8 문자 인코딩이 널리 사용됨.

( )

서술형

10 컴퓨팅 시스템에서 음의 정수를 표현할 때 부호화 절대치 방법, 1의 보수 방법, 2의 보수 방법 중 2의 보수 방법을 주로 사용하는 까닭을 써 보자.

11 그림의 표현 방식 중 벡터 방식이 비트맵 방식에 비하여 가지는 장점을 써 보자.

12 그림판에서 임의로 그림을 그린 후 이를 PNG 파일 형식, JPG 파일 형식으로 각각 저장하여 두 그림 파일 간에 어떠한 차이가 있는지 써 보자.

13 동영상 파일에 사용하는 다음 코덱들을 조사해 보자.

코덱명	특징
H.264	
WMV	
Apple ProRes	
VP9	