멀티미디어

2024.

목차

1. 텍스트의 개요
2. 텍스트의 표현
3. 텍스트 관련 기술의 변화, 저장기술
4. 텍스트 기반 멀티미디어 서비스
5. 문자 인식 기술
6. 전자책

01. 텍스트 (TEXT) 의 개요

■ 텍스트의 개념과 특징

▪ 텍스트 (TEXT)

- **여러 문장이 모여 만들어진 문장의 집합**
- 특정한 의도에 따라 문자를 사용하여 작성된 문서
- **멀티미디어에서의 텍스트 : 사람이 이해할 수 있게 인공적으로 만든 2차원 형태의 미디어**
- 문자, 기호, 단어, 구, 문장, 다이어그램, 도표, 인터넷 주소 등과 같이 문자의 배열 형태로 나타남
- 콘텐츠에 포함되는 멀티미디어 데이터 중 가장 많이 사용
- 다른 종류의 멀티미디어 데이터보다 기억 용량을 적게 차지
- 텍스트를 사용하여 문서를 작성하기 위해서는 해당 언어를 지원하는 도구를 사용해야 함 (오피스 프로그램)

▪ 글자와 기록 매체의 기원

- **글자**
  - 말을 일정한 체계로 표현한 기호
  - 약 6,000년 전 메소포타미아에서 사용한 쐐기문자가 글자의 기원
  - 최고의 글자 : 한글 https://youtu.be/dtXK0z0BuL0?si=fRA-X5XW29c5ak0C

- **종이**
  - AD105년 중국 후한의 채륜이 발명
  - 인쇄술의 발명과 활용을 기반으로 지식과 정보가 공유되고 확산 (최초의 인쇄 : 751년 무구정광다나리경, 목판본, 불국사 석가탑에서 발견)
  - 컴퓨터, 스마트폰 등 스마트 디바이스가 우리 생활 의 필수품으로 자리를 잡으면서 종이 소비가 크게 감소

01. 텍스트의 개요

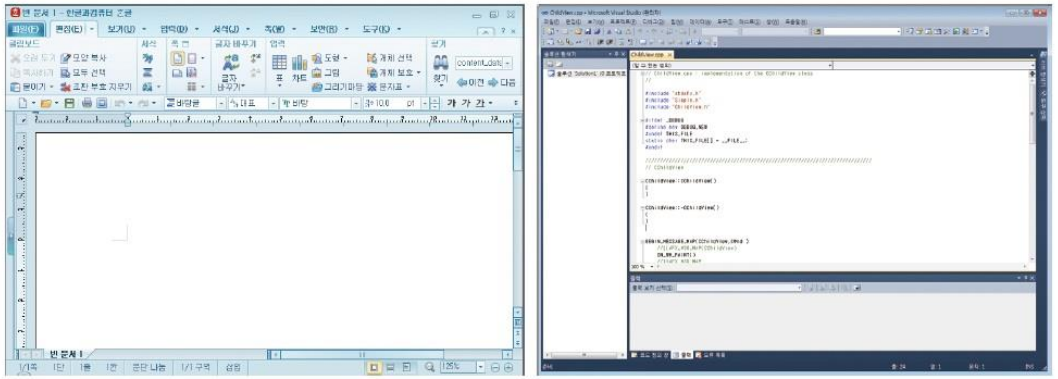
■ 문서편집기와 워드프로세서

▪ 문서 편집기

- 문서를 작성할 때 사용하는 프로그램
- 입력과 수정 기능이 있지만 단순하여 문서 작성에 많은 제한
- 프로그램을 작성하거나 웹 문서 제작에 필요한 소스코드를 입력할 때 주로 사용

▪ 워드 프로세서

- 문서의 작성, 편집, 저장 및 인쇄할 때 사용하는 하드웨어, 소프트웨어를 정의하는 용어 (대부분 소프트웨어)
- 문서를 보다 손쉽게 작성할 수 있도록 다양한 편의 기능을 제공
- 문서편집기보다 기능이 다양 (글자의 속성 변경, 이미지나 도표 등을 추가)
- 문서의 호환이 완벽하게 이루어지지 않는다는 단점이 있었으나 개방형 문서 표준(ODF)의 도입으로 해결됨
- 국내 문서 작성 시장은 MS 워드와 한글이 양분



(a) 한글 2010　　(b) 비주얼 스튜디오 2010

그림 4-1 텍스트 편집 프로그램

■ 문서편집기와 워드프로세서의 진화

- 네이버 워드, 구글 문서 같은 온라인 워드프로세서 프로그램
- 무료로 사용할 수 있는 공개형 워드프로세서인 오픈오피스 : 에스메모(SMemo)

그림 4-3 에스메모

문서편집기와 워드프로세서

■ 문서편집기와 워드프로세서 비교

	문서편집기	워드프로세서
특징	• 어떠한 종류의 저장 형식도 지원하지 않음 • 워드프로세서에 비해 문서 작성 기능이 제한적임 (글자 크기, 폰트, 글자 색깔 등을 바꿀 수 없음) • 대부분의 문서편집기와 호환이 잘됨 → 불특정 다수를 대상으로 하는 문서를 작성할 때 사용 • 프로그래밍 소스코드를 작성할 때 사용	• 필요에 따라 다양한 형식으로 저장 가능 • 문서편집기보다 다양한 기능을 가짐(글자 속성, 문단 속성, 다양한 개체 삽입, 사전 기능, 문법 교정 기능 등) • 다른 워드프로세서와 호환성이 낮음(한글로 작성한 파일을 MS 워드로 불러오지 못하거나 MS 워드로 작성한 파일을 한글로 불러오지 못함)
프로그램 예	윈도우의 메모장, 매킨토시의 텍스트에디터	한글, MS 워드

■ 음성 입력 ➔ 문자, 문서 작성

참고 : Google WorkSpace 음성으로 문서 작성하기 : https://youtu.be/G6AfB4FuYl4?si=n5u_Q8UI9Cd2hBNi

02. 텍스트의 표현

■ 코드 시스템 (Code System)

- 코드시스템
  - **컴퓨터에서 문자를 사용하기 위해 약속된 이진코드를 일정한 규칙에 따라 각 문자에 할당하는 것**
  - 각각의 문자를 컴퓨터에 저장하고 컴퓨터 내부에서 문자들을 구분하여 사용하기 위해 사용
  - 컴퓨터 내부에서 모든 문자는 이진코드로 인코딩 (Encoding) ➔ 디코딩 (Decoding) 하여 출력

    — 인코딩: 컴퓨터에 저장하거나 통신에 사용할 목적으로 문자나 기호를 다른 형식으로 변환하는 방식
  - 아스키 (ASCII) 코드 : 'A' : 이진코드 = '1000001' / 코드 값 = 65
  - 코드 시스템은 언어에 따라 다름
    - ➢ 알파벳 사용권에서는 8비트(1바이트) 코드, 한자를 사용하는 동양권에서는 16비트(2바이트) 코드 사용

■ 표준 코드 시스템

- 1963년 아스키 코드(ASCII, American Standard Code for Information Interchange Code)
- 1964년 EBCDIC(Extended Binary Coded Decimal for Interchange Code)
- 1995년 유니코드(Unicode) : 각 나라의 코드 시스템을 하나로 통합하고 표준으로 제정되어 현재 사용되고 있음

02. 텍스트의 표현

■ 코드 시스템

▪ 아스키 코드(ASCII)

- 세계적으로 널리 사용되고 있는 코드 체계, 데이터 처리 및 통신시스템에서 상호간의 정보 교환용으로 사용
- 한 문자를 표현하기 위해 7비트를 사용하기 때문에 표현할 수 있는 문자의 수는 2^7으로 총 128자를 표현함

표 4-1 아스키 코드표

10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자
0	0x00	NULL	16	0x10	DLE	32	0x20	sp	48	0x30	0	64	0x40	@	80	0x50	P	96	0x60		112	0x70	p
1	0x01	SOH	17	0x11	DC1	33	0x21	!	49	0x31	1	65	0x41	A	81	0x51	Q	97	0x61	a	113	0x71	q
2	0x02	STX	18	0x12	DC2	34	0x22	"	50	0x32	2	66	0x42	B	82	0x52	R	98	0x62	b	114	0x72	r
3	0x03	ETX	19	0x13	DC3	35	0x23	#	51	0x33	3	67	0x43	C	83	0x53	S	99	0x63	c	115	0x73	s
4	0x04	EOT	20	0x14	DC4	36	0x24	$	52	0x34	4	68	0x44	D	84	0x54	T	100	0x64	d	116	0x74	t
5	0x05	ENQ	21	0x15	NAK	37	0x25	%	53	0x35	5	69	0x45	E	85	0x55	U	101	0x65	e	117	0x75	u
6	0x06	ACK	22	0x16	SYN	38	0x26	&	54	0x36	6	70	0x46	F	86	0x56	V	102	0x66	f	118	0x76	v
7	0x07	BEL	23	0x17	ETB	39	0x27	'	55	0x37	7	71	0x47	G	87	0x57	W	103	0x67	g	119	0x77	w
8	0x08	BS	24	0x18	CAN	40	0x28	(	56	0x38	8	72	0x48	H	88	0x58	X	104	0x68	h	120	0x78	x
9	0x09	HT	25	0x19	EM	41	0x29	)	57	0x39	9	73	0x49	I	89	0x59	Y	105	0x69	i	121	0x79	y
10	0x0A	₩n	26	0x1A	SUB	42	0x2A	*	58	0x3A	:	74	0x4A	J	90	0x5A	Z	106	0x6A	j	122	0x7A	z
11	0x0B	VT	27	0x1B	ESC	43	0x2B	+	59	0x3B	;	75	0x4B	K	91	0x5B	[	107	0x6B	k	123	0x7B	{
12	0x0C	FF	28	0x1C	FS	44	0x2C	,	60	0x3C	<	76	0x4C	L	92	0x5C	₩	108	0x6C	l	124	0x7C	|
13	0x0D	₩r	29	0x1D	GS	45	0x2D	-	61	0x3D	=	77	0x4D	M	93	0x5D	]	109	0x6D	m	125	0x7D	}
14	0x0E	SO	30	0x1E	RS	46	0x2E	.	62	0x3E	>	78	0x4E	N	94	0x5E	^	110	0x6E	n	126	0x7E	~
15	0x0F	SI	31	0x1F	US	47	0x2F	/	63	0x3F	?	79	0x4F	O	95	0x5F	_	111	0x6F	o	127	0x7F	DEL

02. 텍스트의 표현

■ 코드 시스템

▪ 아스키 코드(ASCII)

- 패리티(Parity) 비트
  - 컴퓨터 환경에서 일반적으로 사용하는 데이터 단위는 8비트이므로 아스키 코드를 8비트로 구성하여 사용
  - 공백으로 남는 나머지 1비트는 패리티 비트로 활용
    - 패리티 비트: 오류 검출을 목적으로 사용하는 비트, 짝수 패리티 비트와 홀수 패리티 비트가 있음

표 4-2 짝수 패리티 비트와 홀수 패리티 비트

오리지널 데이터	짝수 패리티	홀수 패리티
00000000	0	1
01011011	1	0
01010101	0	1
11111111	0	1
10000000	1	0
01001001	1	0

  - 오류검출 예) 짝수 패리티비트에서 아스키 코드가 '1100000'인 경우
    - 7비트 코드 안에 있는 1의 개수가 짝수이므로 패리티 비트는 '0'이 되어 전체 코드는 '01100000'으로 구성
    - 한 비트의 에러가 발생하면 오류 검출이 가능

그림 4-4 짝수 패리티 비트 사용 시 오류 검출 방법

  - 패리티 비트를 이용하면 오류를 쉽게 검출할 수 있지만 해결은 불가능
  - 한 비트 오류에 대해서만 검출할 수 있고 두 비트 이상은 해결할 수 없음

02. 텍스트의 표현

코드 시스템

확장 아스키 코드(Extended ASCII)

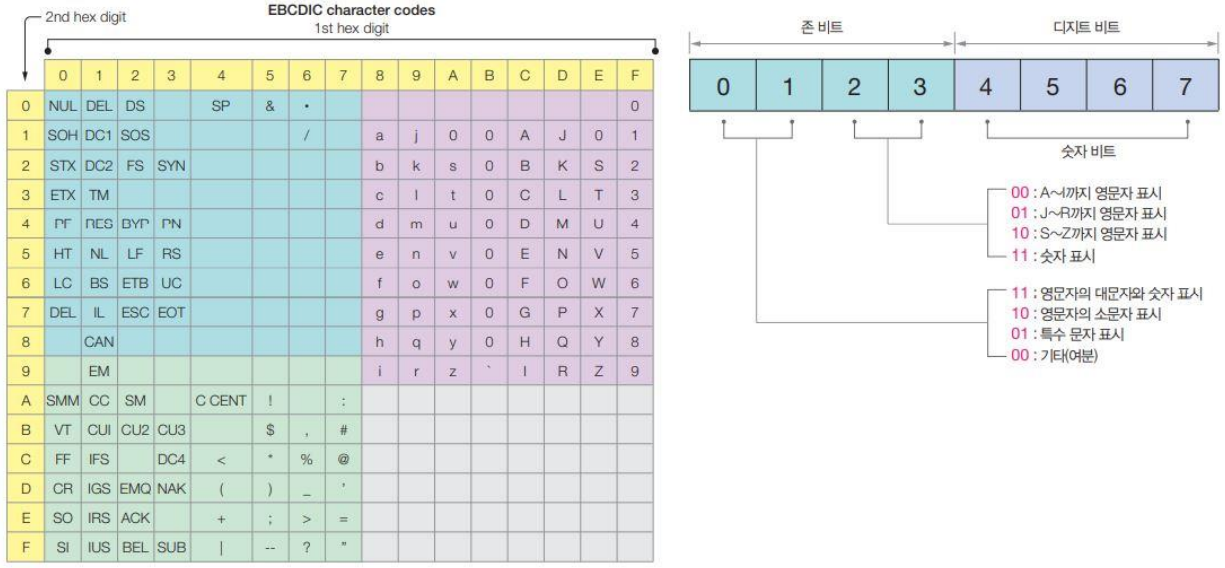
- 독일어, 불어 같이 영어의 알파벳 외에 점이 있는 문자, 물결 표시 문자를 사용하는 나라 아스키 코드 대신 사용
- 패리티 비트를 사용하지 않고 8비트를 모두 문자를 표현하는 데 사용
- 2^8 = 256개의 문자를 표현할 수 있음

02. 텍스트의 표현

■ 코드 시스템

▪ EBCDIC

- IBM에서 서버급의 중대형 컴퓨터에 사용하기 위해 개발
- 코드를 구성하기 위하여 8비트를 사용하지만 아스키 코드와는 전혀 다름 (2^8인 265개의 문자)
- 8개의 비트는 2개의 영역으로 나누어 상위 4비트와 하위 4비트로 구분
- 상위 4비트는 존 비트 (Zone Bit) / 하위 4비트는 디지트 비트 (Digit Bit) 또는 뉴메릭 비트 (Numeric Bit)
- 영역에서 읽은 비트는 코드 테이블에 대응시켜 해당 코드가 정의하는 문자를 알아냄
- EBCDIC에서는 실제 256개의 문자를 모두 사용하지 않고 150개 정도의 코드만 사용
- ASCII 코드가 주로 사용되며, 많이 사용하지 않음.



(a) 코드 테이블 (b) 코드 구조

그림 4-5 EBCDIC의 코드 테이블과 코드 구조

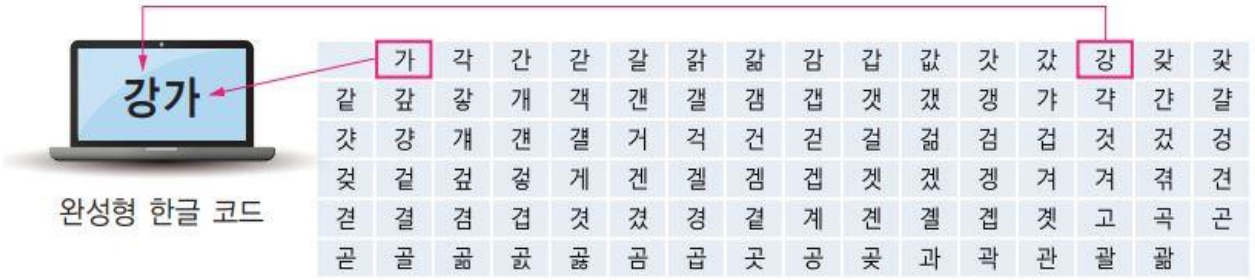
02. 텍스트의 표현

■ 코드 시스템

▪ 한글 코드

- 한글의 경우 8비트 코드 체계로 표현할 수 없으므로 별도의 코드시스템이 필요
- 글자를 초성, 중성, 종성으로 구분하여 표현하는 방법이 개발 ➔ 조합이 가능한 전체 글자수 19×21×28=11,179
  - 조합형 코드: 초성, 중성, 종성에 대한 각각의 코드를 저장하고 실제로 출력할 때 조합함
  - 완성형 코드: 초성, 중성, 종성의 낱글자들을 미리 조합하여 코드로 완성시켜 놓은 것

조합형 한글 코드



완성형 한글 코드

그림 4-6 조합형 코드와 완성형 코드의 글자 표현 방식

02. 텍스트의 표현

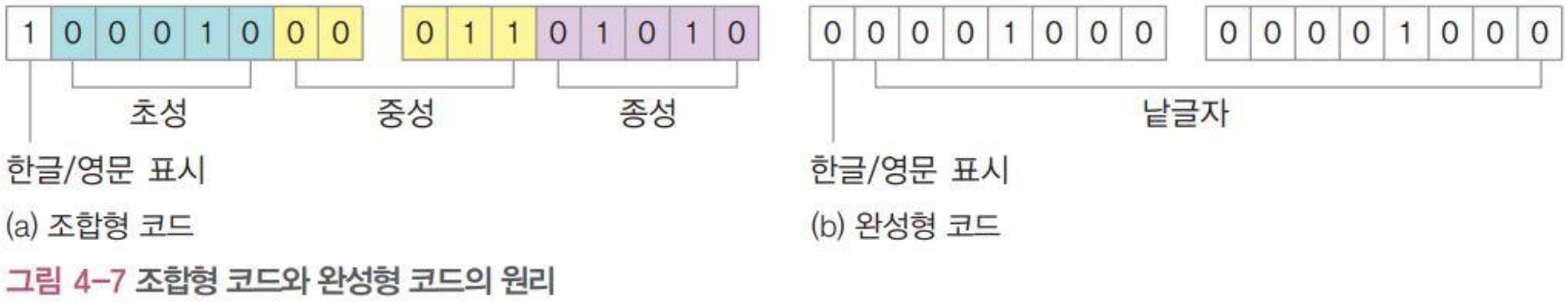


그림 4-7 조합형 코드와 완성형 코드의 원리

- 최상위비트 (MSB)가 1인 경우는 한글 코드이므로 2바이트를 가지고 해석
- MSB가 0인 경우는 아스키 코드이므로 1바이트씩 해석

02. 텍스트의 표현

■ 코드 시스템

▪ 유니코드

- 모든 나라에서 공통으로 사용할 수 있는 코드 체계가 필요함에 따라 개발된 코드
- 1990년에 제록스 외 몇 개 업체의 연구자들이 모여서 개발한 산업체 표준
- 1991년에 유니코드 버전 1.0이 발표되었고 세계 표준으로 채택
- 컴퓨터에서 다양한 언어 표현이 가능해짐에 따라 소프트웨어의 국제화에 기여

- 전 세계의 모든 문자를 8비트 단위인 옥텟 (Octec) 으로 표현
- 기본적으로 하나의 문자를 4개의 옥텟으로 표현하나, 2개의 옥텟만을 사용하는 코드 세트가 정의되어 있어 주로 이 코드 체계를 사용

- 데이터 용량을 많이 차지하기 때문에 문서 표현이나 처리보다는 문서 교환이나 통신 분야에 알맞음
- 최대 수용 문자 수는 216=65,536

02. 텍스트의 표현

폰트

폰트의 개요

- 인쇄 환경에서 사용하던 용어로 글자의 모양을 나타냄(글꼴이라고 부름)
- 컴퓨터에서 사용되는 모든 문자의 모양과 크기에 대한 정보를 가지고 있음
- 구성 방법에 따라 비트맵(Bitmap) 폰트와 벡터(Vector) 폰트로 구분됨

폰트의 속성

- 크기 : 포인트(Point)라는 단위로 부르며 pt로 크기를 나타냄
- 장평 : 한 글자의 가로 길이와 세로 길이의 비율
- 자간 : 글자와 글자 사이의 간격

MD이솝체	그래픽	**느낌체**	사랑체	파도체
HY중고딕	**HY견고딕**	HY신명조	HY견명조	HY그래픽M
HY궁서B	HY크리스탈M	**HY백송B**	**HY헤드라인M**	HY목각파임B
HY목판B	HY엽서L	HY엽서M	HY얕은샘물M	**HY울릉도M**
휴먼아미체	**휴먼모음T**	**휴먼엑스포**	**휴먼매직체**	**휴먼고딕**
문체부 바탕체	문체부 돋음체	문체부 제목 돋움체	문체부 제목 바탕체	**각진제목체**
가는안상수체	**굵은안상수체**	바탕한체	굵은바탕한체	가는바탕한체
굵은돋움한체	**양재 소슬**	**양재 튼튼B**	**양재 참숯B**	양재 둘기

스마트시대의 멀티미디어 1234567890
스마트시대의 멀티미디어 1234567890
스마트시대의 멀티미디어 1234567890
스마트시대의 멀티미디어 1234567890
스마트시대의 멀티미디어 1234567890
스마트시대의 멀티미디어 1234567890
스마트시대의 멀티미디어 1234567890

그림 4-11 다양한 글자의 모양과 크기

폰트

■ 고정간격 폰트 VS 비례간격 폰트

- 모니터에 출력되는 형태에 따라 고정간격 폰트(고정 폰트)와 비례간격 폰트(가변 폰트)로 구분
  - ➢ 고정간격 폰트 : 가로 길이가 일정한 폰트
  - ➢ 비례간격 폰트 : 선택하는 글꼴에 따라 가로 길이가 일정하지 않은 폰트

표 4-2 글꼴별 다양한 변화

구분	글꼴	I 글자의 경우	H 글자의 경우
비례간격 폰트	신명조	IIIIIIIIIII	HHHHH
	바탕	IIIIIIIIIII	HHHHH
	굴림	IIIIIIIIIII	HHHHH
고정간격 폰트	바탕체	IIIIIIIIIII	HHHHH
	굴림체	IIIIIIIIIII	HHHHH

■ 타입페이스(Typeface)

- 글씨를 써 놓은 모양을 나타내는 용어로 일반적으로 글꼴이라고 부름
- 크게 사각형글꼴과 비사각형글꼴로 구분함
  - ➢ 사각형글꼴 : 동일한 넓이의 사각형 틀을 기본으로 함, 대표적으로 맑은고딕체
  - ➢ 비사각형글꼴 : 사각형 틀을 벗어난 글씨체를 의미, 대표적으로 안상수체

표 4-3 타입페이스의 종류와 정의

타입페이스	정의
세리프(Serif)	• 문자의 끝부분을 삐치게 만든 것 • 텍스트의 본문에 적합 예 Times, Bookman, Palatino
산세리프(Sans Serif)	• 글자의 끝부분에 삐침이 없는 것 • 제목, 강조될 문장에 적합 예 Helvetica, Arial, Optima, Avant Garde
타입 스타일	• 굵음(boldface), 이탤릭(italic) 등
타입 스타일 속성	• 밑줄(underlining), 외곽선(outlining) 등
타입 크기	• 일반적으로 포인트(pt) 단위로 표현하며, 1pt=1/72인치 • 문자의 상단에서 하단까지의 거리를 말함(줄 간격을 위한 약간의 여백 포함)

■ 타입페이스(Typeface)

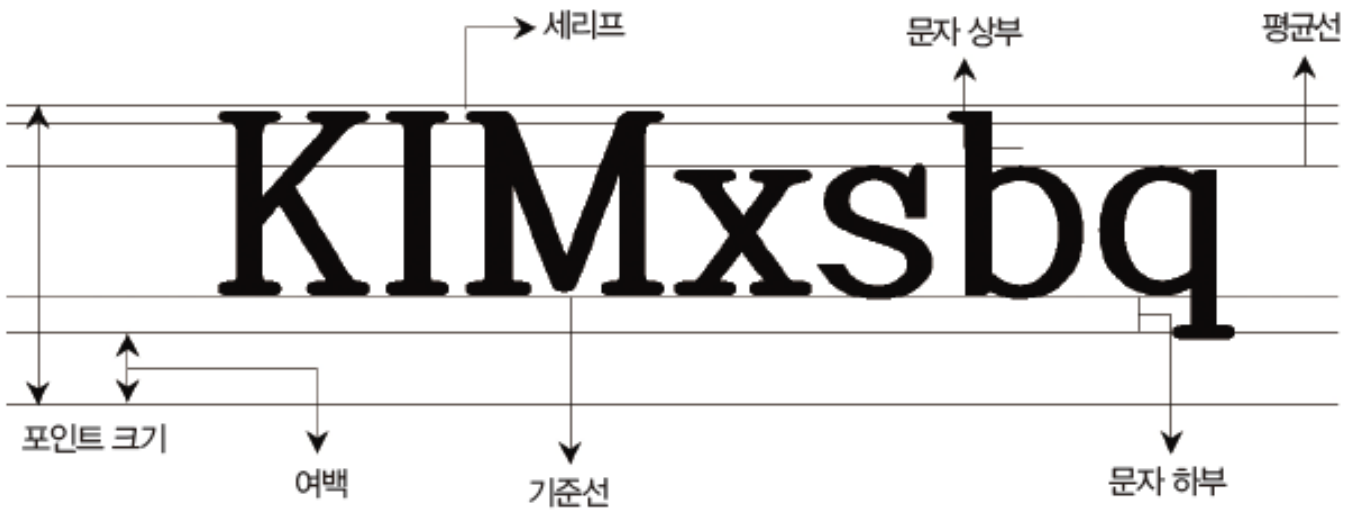


그림 4-12 타입페이스의 구성 요소

비트맵 폰트와 벡터 폰트

■ 비트맵 폰트

- 2차원의 사각 평면을 작은 픽셀(Pixel) 단위로 분할하여 그 위에 글자나 이미지의 형상을 그대로 표현
- 각 픽셀은 비트(Bit)에 해당하는 0 또는 1이 저장되는데 이러한, 여러 개의 점들이 조합하여 정보가 표현됨

■ 비트맵 이미지

- 비트맵 방식으로 이미지를 저장하고 관리하는 것
- 대표적으로 GIF, JPG, PNG, BMP, TIFF, PCT, PCX 등이 있음
- 비트맵 폰트와 비트맵 이미지를 다른 용어로 래스터 폰트와 래스터 이미지라고도 함

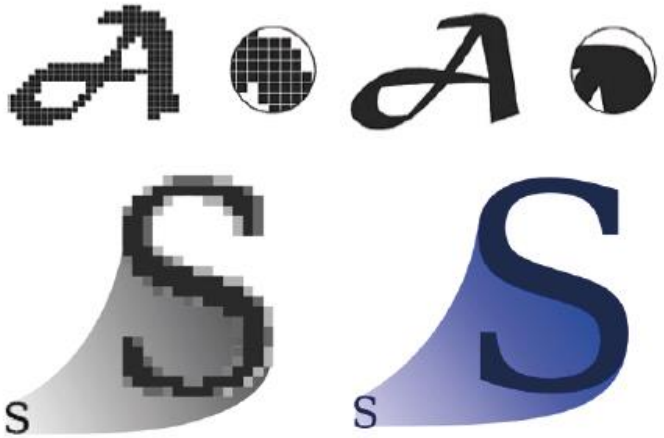


그림 4-13 비트맵과 벡터의 개념

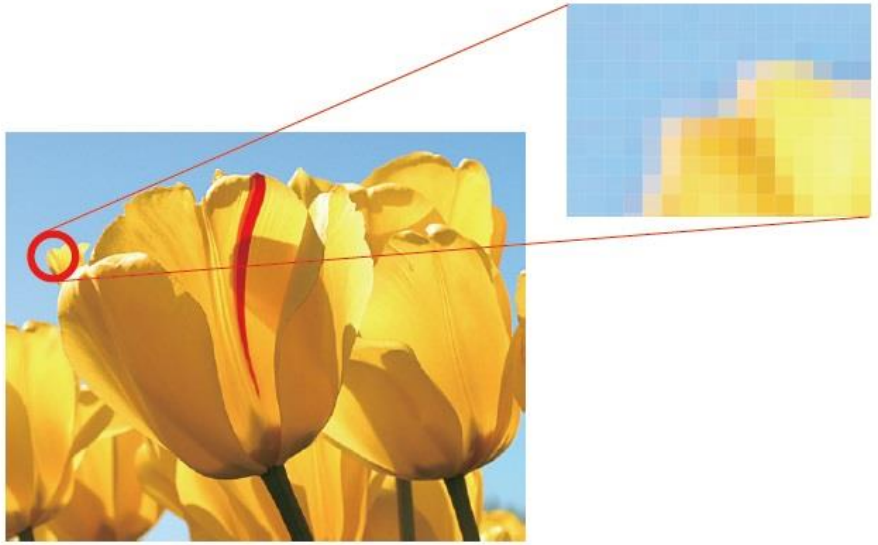


그림 4-14 계단현상

비트맵 폰트와 벡터 폰트

■ 비트맵 방식의 장단점

장점	• 출력하기 위해 복잡한 연산을 거치지 않기 때문에 빠름 → 주로 화면표시용으로 사용
단점	• 글자나 이미지의 크기가 커지면 필요한 메모리 용량도 증가함 • 확대, 축소를 대비하여 가능한 많은 사이즈를 등록해 놓아야하기 때문에 데이터 양이 증가함 • 글씨나 이미지를 확대하거나 축소하면 계단현상이 발생하여 모자이크처럼 깨짐 → 앨리어싱(Aliasing) 현상

■ 앨리어싱 (Aliasing) 현상을 제거하기 위한 방법

- 모니터의 해상도를 높여 이미지를 좀 더 조밀하게 점으로 표현
- 경계면의 밝기를 조절하여 중간 색조로 부드럽게 보이도록 처리하는 안티앨리어싱(anti-aliasing)
- 계단현상을 제거하는 기술인 RET(Resolution Enhancement Technology)

비트맵 폰트와 벡터 폰트

■ 벡터 폰트와 벡터 이미지

- 문자나 이미지의 모양을 윤곽선의 방향과 길이로 기억하는 방식
- 글자와 이미지 모양을 나타내기 위해 시작하는 좌표의 위치(X1, Y1)와 끝나는 좌표의 위치(X2, Y2)를 지정하는 방식

▶ 동영상 보기 : Create a vector font from images

■ 벡터 방식의 장단점

장점	• 크기나 해상도에 영향을 받지 않음 • 글자나 이미지를 확대하거나 축소해도 일그러짐 없이 깨끗한 모양을 유지함 • 글자나 이미지를 표현하기 위한 좌표만 저장하기 때문에 메모리 용량을 적게 차지함
단점	• 복잡한 수학적 알고리즘을 가지고 있기 때문에 크기에 상관없이 알고리즘을 저장할 메모리 공간이 필요함 • 글자 또는 이미지를 표현하는 속도는 비트맵에 비하여 떨어짐

02. 텍스트의 표현

■ 폰트

▪ 한글 폰트의 생성과 진화

- 최 정순, 최 정호라는 글씨 장인 두 명의 평생에 걸친 노력에 힘입어 현재의 한글 폰트가 등장
- 한글은 디자이너 관점에서 글자의 제작 방법과 글자 모양의 변화에 연계돼 정리
- 굴림체 폰트: 1970년대에 일본의 인기 글꼴인 나루체(둥근고딕)를 모방 하여 급히 제작
- 맑은고딕 폰트: 한국적인 아름다움과 조형미를 현대적으로 표현하고, 가독성을 극대화한 폰트로 제작
- 윈도우에 탑재된 한글 시스템의 폰트의 용량이 큰 경우 기본 서체를 변경하여 사용할 수 있음

그림 4-10 윈도우 기본 글꼴

- 캘리그래피
  - 글씨를 쓰는 사람의 의도를 표현하기 위해 마치 그림을 그리듯이 글씨를 쓰는 것
  - 불규칙함, 동적인 선, 조형적인 효과, 독창성을 특징으로 하는 아름다운 손글씨

02. 텍스트의 표현

폰트

디지털 시대의 텍스트 위기

- 텍스트에 대한 이해력 저하
  - 젊은 세대는 텍스트를 훑듯이 읽고 지나가면서 자신은 이해한 것으로 착각
  - 요즘 학생들은 과제를 해결하기 위하여 검색엔진 대신 유튜브에 접속해 과제를 해결
  - 짧은 문장에 익숙하고 긴 문장을 읽기 힘든 현상은 텍스트에 대한 이해력 저하로 이어짐

- 위기의 한글 사용
  - 인터넷의 발달로 언어 파괴 현상이 심각해짐
  - 파괴 현상은 PC 통신이 보급된 시기부터 나타남
  - 젊은 세대를 중심으로 축약된 말이나 다른 세대와 소통 되지 않는 외래어 같은 말을 쓰는 부정적인 측면

- 디지털 시대와 손글씨의 가치
  - 지금은 손글씨보다 키보드와 스마트폰 자판이 더 익숙한 시대이지만 손글씨는 여전히 가치가
  - 손글씨로 문장을 작성하면 자판을 사용할 때보다 사용하는 단어와 어휘가 더 풍부해 짐
  - 손글씨를 예쁘게 쓰기 위해 집중하는 힘도 길러짐
  - 수업 시간에 필기를 병행하면 내용이 오랫동안 기억되고, 학습 내용을 재구성하는 능력도 향상

폰트 디자인, 제작

폰트 디자인은 어떻게 만들어지는가?
https://youtu.be/S2kC7BLYXZE?si=lo_1kIB-GFQfDjNN

한글 폰트 개발이 중요한 이유는 ?
https://youtu.be/rLKk4fXgjzY?si=jxqGviZuibVi1dcl

03. 텍스트 관련 기술의 변화

■ 문서 시장의 변화

▪ 문서 표준

- 예전엔 문서편집기가 MS워드, 아래한글, 엑셀, 파워포인트 등을 각각의 문서 파일로 인식하여 여러 문제 발생

1. 각각의 프로그램에서 향상된 편집 기능을 제공하기 위하여 지속적으로 새로운 버전의 소프트웨어를 제공해야 함
   새로운 버전과 이전 버전 사이에 작성된 문서의 호환성 문제가 발생

2. 일반적으로 많이 사용되는 HWP, DOC, XLS, PPT 도구는 특정 기업에 종속된 폐쇄형 전자문서임
   새로운 버전의 문서 개발도 이들 기업에 종속

▪ ODF

- 앞서 기술한 문제를 해결하기 위해 개방형 문서 표준인 ODF가 탄생
- ODF 전자문서가 도입되면서 그동안 사회·경제적으로 비효율적으로 인식된 종이 문서 사용량이 감소
- 개방형 표준 전자문서에서는 대표적으로 개방형 문서 표준 형식(ODF), XML(HTML), PDF 등을 사용
- 개발 주체가 불명확하여 주기적으로 소프트웨어의 성능을 개선하는 폐쇄형 전자문서보다 품질이 떨어지는 문제

그림 4-13 개방형 문서 표준인 ODF의 역할

03. 텍스트 관련 기술의 변화

■ 문서 시장의 변화

▪ 클라우드 환경의 오피스 서비스

- 최근의 오피스 소프트웨어는 PC 중심의 설치 방식에서 모바일과 클라우드 환경으로 이동
- 별도의 프로그램 설치 없이 PC, 웹, 모바일 환경에서 서로 연동해 사용
- 클라우드 서비스 사용자는 제작한 문서를 클라우드 저장소에 저장하고 모든 플랫폼에서 공유,편집이 가능

- **폴라리스 오피스**
  - 클라우드 환경에서 문서를 작성하고, 공유를 통해 협업하고, 결과물을 생산하는 최근 트렌드를 반영한 서비스
  - 워드, 엑셀, 파워포인트 등 다양한 형태의 문서를 PDF 파일로 제작하고 공유
  - 반대로 PDF 문서를 워드, 엑셀, 파워포인트 문서로 변경·편집한 뒤 다시 PDF로 저장하는 재편집도 가능

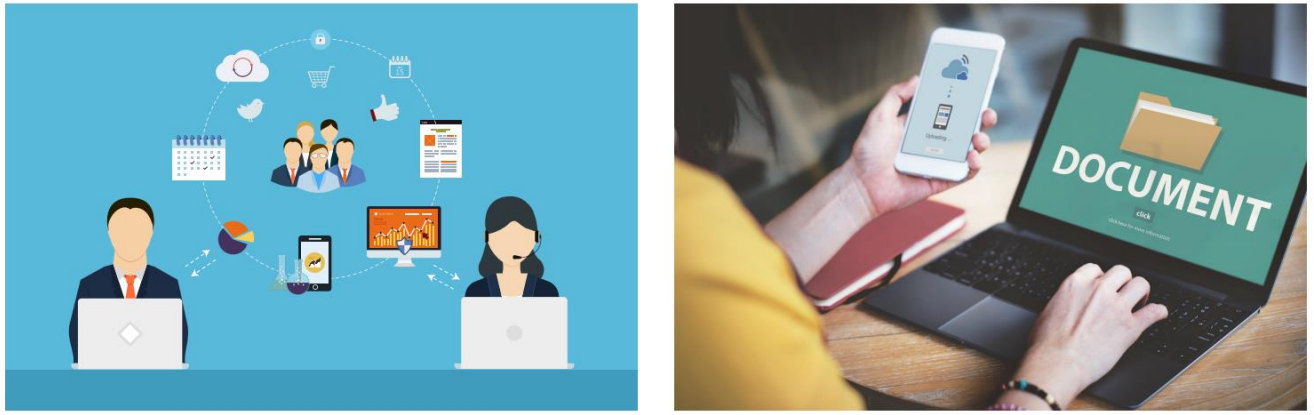


그림 4-14 사용자 편의성을 제공하는 클라우드 환경의 오피스 서비스

03. 텍스트 관련 기술의 변화

PDF

PDF의 개념

- 개방형 문서 표준인 ODF기반의 전자문서
- 원래 어도비의 소유였으나 월드와이드 웹 컨소시엄(W3C)에서 누구나 이용할 수 있는 개방형 문서 표준으로 공개
- 문서 위·변조 방지 기능이 있기 때문에 여러 개방형 표준 가운데서도 열람, 보관용으로 가장 적합한 문서
- PDF 작성 프로그램인 아크로뱃은 유료이나 뷰어 프로그램인 아크로뱃 리더는 무료
- 전자책 시장에서도 EPUB과 더불어 주요 전자책 포맷으로 각광 받음

그림 4-15 PDF 문서의 변환

03. 텍스트 관련 기술의 변화

PDF

PDF의 장점

- **문서의 호환성이 높고 파일의 무결성을 가짐**
  - PDF 파일은 대부분의 컴퓨터에서 읽기와 인쇄가 가능하여 인터넷, 인트라넷에서 정보를 공유할 때 적합한 형식
  - 파일의 무결성: 온·오프라인으로 전송된 파일이 원본 문 서와 동일하다는 의미
  - 어떤 프로그램으로 PDF를 작성하더라도 텍스트, 도면, 이미지, 그래픽 등 소스파일 정보가 그대로 유지
- **문서 사용과 관리가 편리**
  - PDF 파일은 자체에 압축 기능이 들어 있어 다른 파일에 비해서 상대적으로 용량이 적음
  - 문서의 깨짐을 방지하기 위해 폰트, 이미지, 그래픽, 표 등과 같은 정보를 하나의 파일에 자유롭게 포함(임베딩) 하여 저장
  - PDF를 사용하여 책 한 권을 하나의 파일로 만들 수 있으며 책갈피 및 링크 기능을 첨가하여 원하는 부분을 쉽게 찾을 수 있음
- **문서 보안이 뛰어남**
  - PDF 파일은 문서에 암호를 설정하는 기능을 제공하기 때문에 보안이 뛰남
  - 단순히 파일을 읽는 경우에만 보안 기능이 수행되는 것이 아니라 인쇄, 복사, 편집 등 각각의 과정에 대하여 제한을 설정할 수 있음
- **쌍방향으로 인터페이스 삽입이 가능**
  - 일반 워드프로세서에는 없는 쌍방향 인터페이스(체크박스, 글상자, 멀티미디어 등)를 삽입하여 효율적으로 공동 작업을 진행할 수 있게 함

03. 텍스트 관련 기술의 변화

PDF

PDF의 단점

- **문서의 제작 방식이 동일하지 않고, 다양한 파일 형식이 존재함**
  - PDF 문서는 제작 방식에 따라 크게 텍스트 PDF 파일과 이미지형 PDF 파일이 있음
  - 텍스트 PDF 파일은 대부분의 텍스트와 이미지를 수정하거나 편집할 수 있음
  - 이미지형 PDF 파일은 스캐너로 문서를 입력하기 때 문에 편집 범위가 제한

- **편집이 불편함**
  - PDF의 장점 중 하나인 '보안성'은 편집이 불가능 하거나 힘들다는 데에서 나온 것
  - PDF 파일은 하나의 커다란 이미지 형태이기 때문에 워 드 파일처럼 일부를 수정하는 데 제한이 따름
  - PDF 파일을 편집하려면 별도의 프로 그램을 사용해 MS 워드나 엑셀 같은 다른 문서 포맷으로 변환해야 함

03. 텍스트 관련 기술의 변화

PDF

DRM기술

- PDF는 문서에 대한 권한이 없는 사용자가 문서를 열람하고 읽는 것이 가능하기 때문에 개인 정보 유출이 불가피
- 최근에는 PDF 문서에 디지털 저작물 보호 및 관리를 의미하는 DRM 기술이 도입

- DRM 기술
  - 디지털 콘텐츠의 무단 사용을 방지하는 기술
  - 콘텐츠에 특정 인물만 접근하거나 정해진 시간 동안만 접근할 수 있게 제약하는 기술
  - 내용 복사나 화면 캡처도 불가능
  - RM이 설정된 콘텐츠는 지정된 PC와 스마트폰 등에서만 제한적으로 사용할 수 있음

  - PDF 문서에 DRM 기술이 도입됨에 따라 기밀 서류를 보관하거나 인터넷에 데이터베이스를 구축 하는데 PDF 문서가 많이 이용

웹 기반 문서(html, xml)

■HTML

- 인터넷에 웹사이트를 만들기 위한 프로그램 언어
- 1990년대 팀 버너스 리(Timothy John Berners-Lee)에 의해 창안됨
- 웹 문서를 작성하는 보편적으로 방법으로 단순하고 사용하기 편리함
- 웹브라우저를 통해 볼 수 있는 대부분의 웹페이지들은 HTML로 작성
- 확장자는 *.htm 또는 *.html
- 웹브라우저상에 정보를 표시하기 위해 마크업 심볼의 집합으로 구성됨
  - ➢ 마크업 : 특정 위치에 삽입되는 문자(명령어)나 기호를 의미, 태그라고도 함

■ HTML 태그

표 4-7 기본적인 HTML 태그

태그	설명	태그	설명
〈HTML〉 … 〈/HTML〉	HTML 형식의 웹페이지를 선언	〈MENU〉 … 〈/MENU〉	〈LI〉 리스트 아이템 처음과 끝부분에 씀
〈HEAD〉 … 〈/HEAD〉	페이지의 헤드(Head)부를 지정	〈LI〉	리스트 아이템 항목들
〈TITLE〉 … 〈/TITLE〉	페이지의 타이틀을 선언	〈BR〉	줄을 바꿈
〈BODY〉 … 〈/BODY〉	페이지의 몸체(Body) 부분을 지정	〈P〉	문단을 바꿈
〈HN〉 … 〈/HN〉	… 부분을 Hn 크기의 글자로 만듦	〈HR〉	수평선을 그음
〈B〉 … 〈/B〉	… 부분을 볼드체로 처리	〈PRE〉 … 〈/PRE〉	미리 지정된 형태의 텍스트
〈I〉 … 〈/I〉	… 부분을 이탤릭체로 처리	〈IMG SRC="…"〉	이미지를 불러옴
〈UL〉 … 〈/UL〉	순서 없는 리스트 항목을 만듦	〈A HREF=" "〉 … 〈/A〉	하이퍼링크 지정
〈OL〉 … 〈/OL〉	번호가 있는 리스트 항목을 만듦		

■XML(eXtensible Markup Language)

- HTML은 제한된 개수의 태그를 사용하기 때문에 문서의 형태를 충분히 표현할 수 없음
- HTML이 한계를 나타내면서 다양한 변종들이 등장함
  - ➢ 마크업 : 액티브엑스(ActiveX), 플래시(Flash) , 스크립트(Script) 언어, DHTML(Dynamic HTML), 채널
  - ➢ 브라우저 간의 호환성 문제 해결하지 못함
- HTML이 가지는 상호 호환성 문제를 해결하기 위해 XML이라는 새로운 마크업 언어를 표준화함
- HTML과 SGML의 장점을 기반으로 만들어진 웹페이지 기술 언어
- HTML은 태그의 종류가 한정적이지만 XML은 태그를 사용자가 직접 정의할 수 있고 그 태그를 다른 사람들도 사용 가능

■XML의 특징

- 문서를 내용, 구조, 서식으로 개별적인 파일로 생성함
- 문서를 구성하는 각 요소들의 독립성이 보장되기 때문에 문서의 호환성이 높고 내용이 독립적임
- 구성 요소별로 저장, 검색, 재 가공이 용이함
- 확장성이 뛰어나 데이터베이스나 스프레드시트 등과 같은 구조화된 데이터들을 XML로 쉽게 변환할 수 있음
- 데이터를 화면에 출력하기 위해서는 화면 표현용 언어인 XSL(eXtensible Stylesheet Language)이 필요
- 디지털 자료 보존에도 유용함

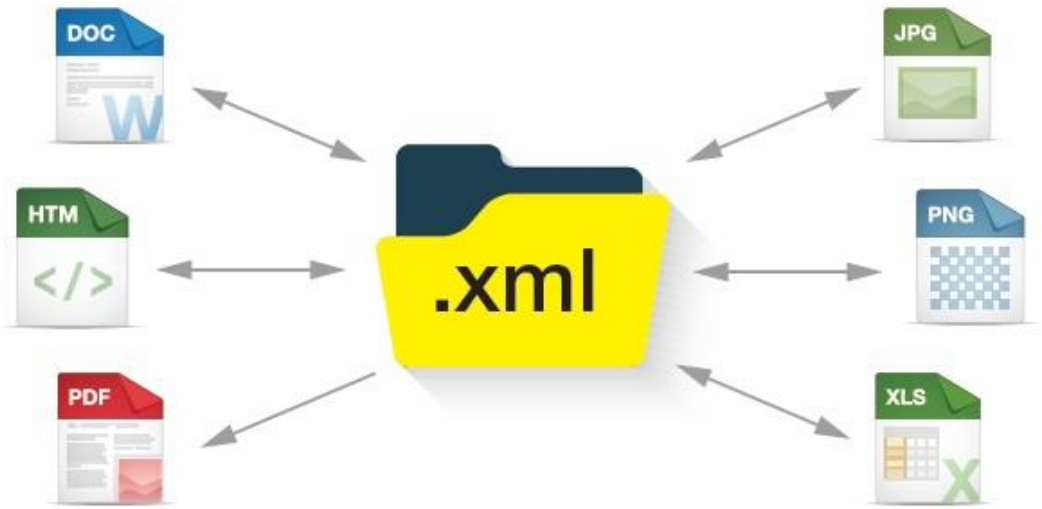


그림 4-19 XML 문서의 변환

문자 인식 기술

■ 문자 인식(Character Recognition) 기술의 개요

- 기존의 인쇄 자료나 손으로 기록한 문자, 기호, 마크 등을 컴퓨터가 자동으로 인식하는 기술
- 패턴 인식(Pattern Recognition)의 한 종류
- 광학적인 장치를 사용하여 인식하기 때문에 광학 문자 인식(OCR, Optical Character Recognition)이라고도 함
- 1970년대부터 상업적 용도로 널리 사용되기 시작함
- 문자 인식 방법으로는 패턴 정합(Pattern Matching)과 구조 분석(Structure Analysis)이 있음
  - ➢ 패턴 정합 : 문자의 유사성, 정합도에 의해 문자를 식별하는 방식, 주로 인쇄문자의 인식에 사용
  - ➢ 구조 분석 : 문자 고유의 특징적인 선의 형태와 특성에 의해 문자를 식별하는 방식, 주로 필기문자의 인식에 사용

https://youtu.be/k571PSUJRwY?si=JjI5OkEcKG340AiX

전자책

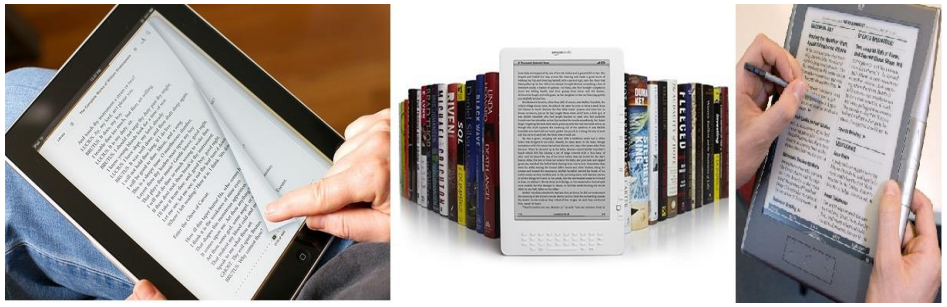
■ 종이책

- 배터리와 관계가 없음
- 눈이 덜 피곤하고 저렴함
- 전자책에 비해 훨씬 견고하고 콘텐츠의 종류도 훨씬 많음

▶ 동영상 보기 : E-Book vs Book

■ 전자책

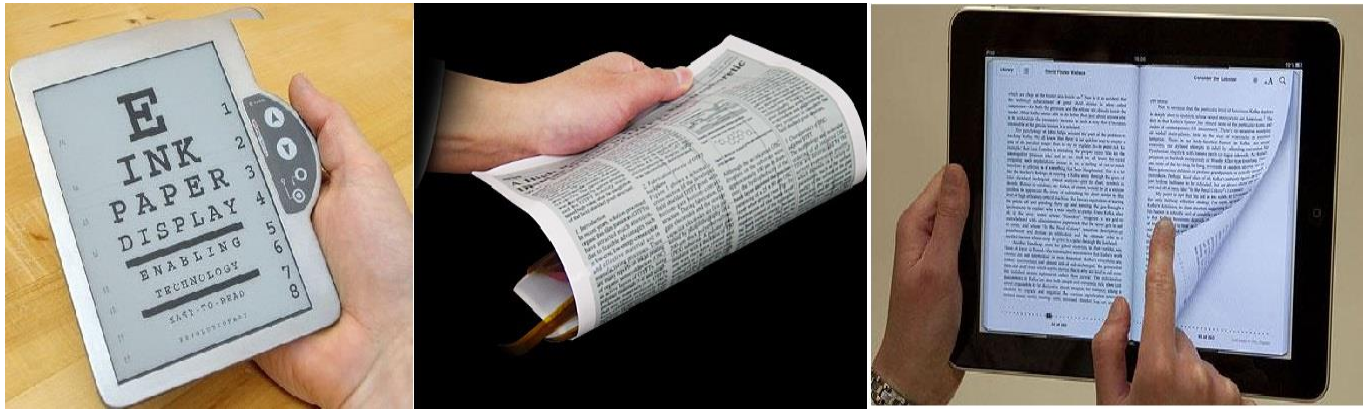
- 동영상을 담을 수 있음
- 전자책은 콘텐츠를 의미하는 디지털 책(Digital Book)과 전자책 리더(Reader)로 나뉨
  - ➢ 디지털 책 : CD-ROM과 같이 패키지 형태나 유무선 인터넷을 통해 유통됨
  - ➢ 전자책 리더 : 디지털 책을 읽을 수 있게 하는 하드웨어 또는 소프트웨어를 의미



전자책

■ 전자책의 특징

- 콘텐츠와 콘텐츠를 전달하는 매체가 분리
  - ➢ 하나의 단말기로 여러 개의 전자책을 볼 수 있지만, 콘텐츠를 각 단말기에 적합한 포맷으로 변환해야 함
- 스크린 미디어로 전환하면서 새로운 읽기 습관이 나타남
  - ➢ 다중 읽기(Multiple Reading) : 텍스트로부터 이탈하는 몰입 대 조작의 습관이 나타남
  - ➢ 소셜 읽기(Social Reading) : 다른 독자와의 교류가 텍스트의 세부적인 수준으로까지 심화됨
  - ➢ 증강 읽기(Augmented Reading) : 아이트래킹 기술을 활용해 독자의 안구움직임을 추적하고 낱말의 뜻을 알려줌



▲ 전자 잉크와 전자종이

전자노트 : https://youtu.be/agsO2jezIOg?si=9kfD8Lm4hsEALbm3

04. 텍스트 기반 멀티미디어 서비스

오디오북

- 스마트 디바이스의 사용 증가에 따른 디지털 피로도가 높아지면서 오디오북이 주목 받음
- 시간과 장소에 관계없이 편리하게 휴대할 수 있고, 책을 들으면서 다른 일을 할 수 있음
- 장애인, 노약자 등 정보 취약층에게도 유용하게 활용
- 책을 읽어주는 방식이 다양하기 때문에 2~3시간 만에 책 한 권을 읽을 수 있음
- 현재 어느 정도를 들었는지 확인하기 어려운 단점

- 최근에 스마트폰, AI 스피커를 통한 글로벌 오디오북 시장이 급성장
- 구글은 구글플레이에 오디오북을 출시
- 네이버는 오디오북을 제작하기 위하여 텍스트를 목소리로 바꿔주는 TTS 엔진을 개발

- 오디오북의 특성상 기계음보다 감정을 살려서 읽는 낭독자에 대한 수요는 줄지 않을 것으 로 예상

- 전자책 기능 : 책 읽어주는 기능 탑재
- 전문 오디오 북 : 윌라 오디오북
- 유투브 책 요약해서 읽어주는 컨텐츠 다수