

# 스마트 시대의

멀티미디어 기본 이론부터  
최신 기술 동향까지



# 멀티미디어

## Chapter 04. 텍스트

# 목차

1. 텍스트의 개요
2. 텍스트의 표현
3. 텍스트의 저장 형식
4. 문자 인식 기술
5. 전자책

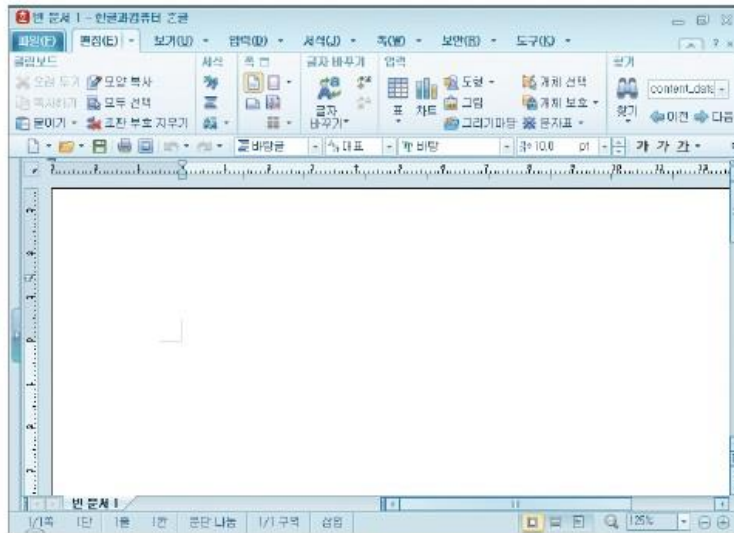
## ■ 개념과 특징

- 텍스트(Text) : 여러 문장이 모여 만들어진 문장의 집합
- 멀티미디어에서의 텍스트 : 사람이 이해할 수 있게 인공적으로 만든 2차원 형태의 미디어
  - 문자, 기호, 단어, 구, 문장, 다이어그램(Diagram), 도표, 인터넷 주소 등과 같이 문자의 배열 형태로 나타남
  - 멀티미디어 데이터 중 가장 많이 사용되며 가장 수월하게 조작할 수 있는 데이터
  - 다른 종류의 멀티미디어 데이터보다 기억 용량을 적게 차지
  - 사용하는 언어에 제한을 받는다는 특징이 있음
  - 대표적인 텍스트 편집 프로그램 : 한글과 컴퓨터사의 '한글', 마이크로 소프트사의 'MS 워드', 워드패드, 메모장 등

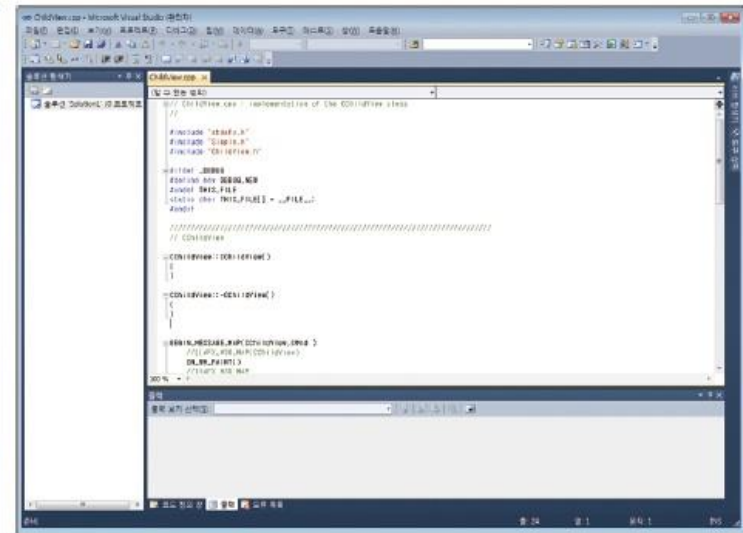
## 1.2 문서편집기와 워드프로세서

### ■ 문서편집기(Text Editor)

- 단순히 텍스트 파일을 읽고 간단히 편집하고 저장하는 데 사용하는 프로그램
- 대표적인 문서편집기는 윈도우의 **메모장**(Notepad)
- 확장자는 .txt이며, **태그(Tag)**나 **스타일(Style)** 같은 포맷을 가지지 **않음**
- 주로 프로그래밍 소스코드를 작성할 때 많이 사용됨



(a) 한글 2010



(b) 비주얼 스튜디오 2010

그림 4-1 텍스트 편집 프로그램

## 1.2 문서편집기와 워드프로세서

### ■ 워드프로세서

- 문서를 작성, 편집, 저장, 인쇄할 때 사용하는 프로그램
- 대표적인 워드프로세서 프로그램은 MS 워드와 한글
- 기본 텍스트 파일을 저장할 수 있을 뿐만 아니라 필요에 따라 다양한 저장 형식을 지정할 수 있음
- 구체적인 파일 포맷과 제어문자도 포함

### ■ 문서편집기와 워드프로세서의 진화

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



그림 4-3 에스메모

## 1.2 문서편집기와 워드프로세서

### ■ 문서편집기와 워드프로세서 비교

표 4-1 문서편집기와 워드프로세서 비교

	문서편집기	워드프로세서
특징	<ul style="list-style-type: none"><li>• 어떠한 종류의 저장 형식도 지원하지 않음</li><li>• 워드프로세서에 비해 문서 작성 기능이 제한적임 (글자 크기, 폰트, 글자 색깔 등을 바꿀 수 없음)</li><li>• 대부분의 문서편집기와 호환이 잘됨 → 불특정 다수를 대상으로 하는 문서를 작성할 때 사용</li><li>• 프로그래밍 소스코드를 작성할 때 사용</li></ul>	<ul style="list-style-type: none"><li>• 필요에 따라 다양한 형식으로 저장 가능</li><li>• 문서편집기보다 다양한 기능을 가짐(글자 속성, 문단 속성, 다양한 개체 삽입, 사전 기능, 문법 교정 기능 등)</li><li>• 다른 워드프로세서와 호환성이 낮음(한글로 작성한 파일을 MS 워드로 불러오지 못하거나 MS 워드로 작성한 파일을 한글로 불러오지 못함)</li></ul>
프로그램 예	윈도우의 메모장, 매킨토시의 텍스트에디터	한글, MS 워드

## 1.2 문서편집기와 워드프로세서

### ■ 음성으로 텍스트 입력하기

▶ 동영상 보기 : [iPhone 6 and iPhone 6 Plus – Voice Text](#)

▶ 동영상 보기 : [Voice, Text Input on Android](#)

▶ 동영상 보기 : [text to voice android](#)

#### :: 여기서 잠깐 음성으로 텍스트 입력하기

텍스트를 입력할 때 키보드나 스캐너를 사용하는 방법 외에도 음성 인식(Voice/Speech Recognition) 기능을 이용하는 방법이 있다. 아이폰의 메모장에서 가상 키보드의 마이크 버튼을 누른 후 입력할 내용을 말하고 [완료] 버튼을 눌러보자. 말한 내용이 텍스트로 입력될 것이다. 안드로이드폰의 경우 가상 키보드에서 설정 버튼을 길게 누르면 나타나는 메뉴에서 마이크 아이콘을 선택한 후 말하면 된다.

텍스트를 음성으로 읽어주는 기능도 있다. 아이폰의 경우는 [설정]-[손쉬운 사용] 메뉴에서 [자동 텍스트 말하기]와 [선택 항목 말하기] 항목을 활성화시킨 후 텍스트를 선택하고 [말하기] 버튼을 누르면 음성으로 읽어준다. 안드로이드폰의 경우는 [환경설정]-[접근성] 메뉴에서 [Talkback] 항목을 활성화하면 명령과 텍스트를 모두 음성으로 읽어준다.



» 음성 메시지를 텍스트로 변환하는 기능

### ■ 코드 시스템(Code System)

- 컴퓨터에서 문자를 사용하기 위해 **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)되며, 이것을 복호화하여 본래의 문자나 기호로 변환하여 화면과 프린터에 출력함
- 아스키(ASCII) 코드의 경우 **문자 'A'는 이진코드 '1000001'로 인코딩되며 코드값은 65**
- 코드 시스템은 언어에 따라 다름
  - 알파벳 사용권에서는 8비트(1바이트) 코드, 한자를 사용하는 동양권에서는 16비트(2바이트) 코드를 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



## 2.1 텍스트의 표현

### ■ 아스키 코드

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

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	Wn	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	Wr	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

그림 4-4 아스키 코드표

## 2.1 텍스트의 표현

### ■ 아스키 코드의 패리티 비트

- 컴퓨터 환경에서는 일반적으로 8비트를 사용하기 때문에 아스키 코드를 8비트로 구성하여 사용함
- 공백으로 남는 나머지 1비트는 오류 검출을 목적으로 하는 패리티(Parity) 비트로 활용함
- 짝수(Even) 패리티 비트와 홀수(Odd) 패리티 비트가 있음
- 패리티 비트는 오류는 쉽게 검출할 수 있지만 해결은 불가능함

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

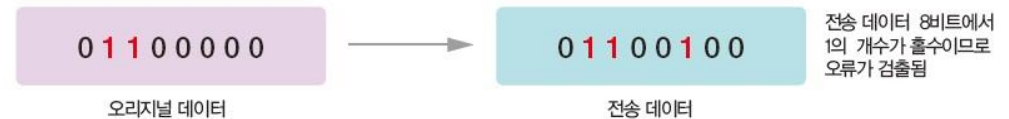


그림 4-6 짝수 패리티 비트 사용 시 오류 검출 방법

그림 4-5 짝수 패리티 비트와 홀수 패리티 비트

### ■ 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 사용하므로  $2^8$ , 256개의 문자를 표현할 수 있음

## 2.1 텍스트의 표현

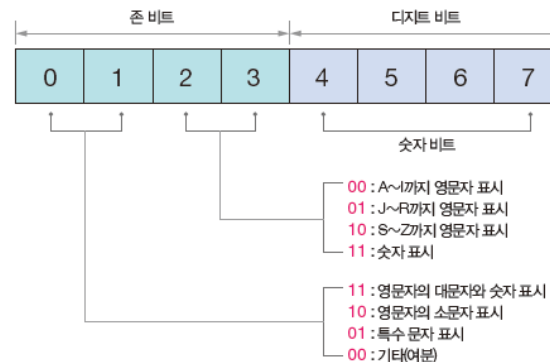
### ■ EBCDIC

- 8비트를 사용하므로 표현할 수 있는 문자 수는  $2^8 = 256$ 개
- 8비트는 상위 4비트(존 비트, Zone Bit)와 하위 4비트(디지트 비트, Digit Bit 또는 뉴메릭 비트, Numeric Bit)로 구분
- 각 영역에서 읽은 비트는 코드 테이블에 대응시켜 해당 코드가 정의하는 문자를 알아냄
- 근래에 아스키 코드가 주로 사용됨에 따라 많이 사용되지 않음

EBCDIC character codes

2nd hex digit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1st hex digit	0	NUL	DEL	DS		SP	&	.								0
1	SOH	DC1	SOS				/		a	j	0	0	A	J	0	1
2	STX	DC2	FS	SYN					b	k	s	0	B	K	S	2
3	ETX	TM							c	l	t	0	C	L	T	3
4	PF	RES	BYP	PN					d	m	u	0	D	M	U	4
5	HT	NL	LF	RS					e	n	v	0	E	N	V	5
6	LC	BS	ETB	UC					f	o	w	0	F	O	W	6
7	DEL	IL	ESC	EOT					g	p	x	0	G	P	X	7
8		CAN							h	q	y	0	H	Q	Y	8
9		EM							i	r	z	.	I	R	Z	9
A	SMM	CC	SM		C	CENT	!	:								
B	VT	CUI	CU2	CU3		\$	,	#								
C	FF	IFS		DC4	<	*	%	@								
D	CR	IGS	EMQ	NAK	(	)	_	'								
E	SO	IRS	ACK		+	:	>	=								
F	SI	IUS	BEL	SUB		--	?	*								

(a) 코드 테이블



(b) 코드 구조

그림 4-7 EBCDIC의 코드 테이블과 코드 구조

- 한글은 8비트로 코드 체계를 표현할 수 없어 별도의 코드 시스템이 필요
- 글자를 초성, 중성, 종성으로 구분하여 표현하면 조합이 가능한 전체 글자수는  $19 \times 21 \times 28 = 11,179$ 가 됨
- 이 방식으로 만들어지는 한글 코드는 조합형 코드와 완성형 코드가 있음
  - 조합형 코드 : 초성, 중성, 종성에 대한 각각의 코드를 저장하고 실제로 출력할 때 조합함
  - 완성형 코드 : 초성, 중성, 종성의 낱글자들을 미리 조합하여 코드로 완성시켜 놓은 것으로 사용의 편리성이 높음



## 2.1 텍스트의 표현

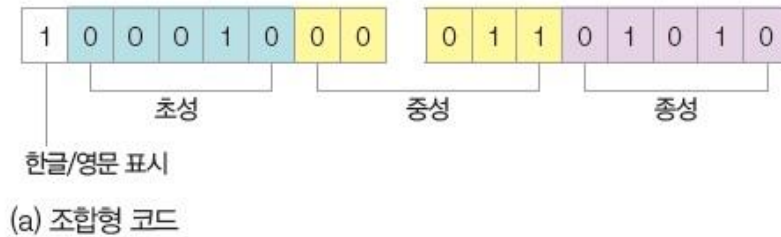


그림 4-9 조합형 코드와 완성형 코드의 원리

### ■ 유니코드

- 각 나라별 코드 체계는 다양하고 복잡
- 인터넷 환경에서는 모든 나라에서 공통으로 사용할 수 있는 코드 체계가 필요
- 각 나라의 코드 체계를 통합하여 개발
- 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

## ■ 폰트(Font)

- 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

▶ 동영상 보기 : [독창적 '디지털 서체' 예술품으로 인정](#)

▶ 동영상 보기 : [font editor](#)

## ■ 폰트의 속성

- 크기 : 포인트(Point)라는 단위로 부르며 pt로 크기를 나타냄
- 장평 : 한 글자의 가로 길이와 세로 길이의 비율(일반적으로 가로 길이와 세로 길이의 기준은 100으로 설정되어 있음)
- 자간 : 글자와 글자 사이의 간격

MD이슈체	그래픽	느낌체	사랑체	파도체
HY중고딕	HY견고딕	HY신명조	HY전명조	HY그래픽M
HY궁서B	HY크리스탈M	HY책송B	HY에드라인M	HY목각파임B
HY목판B	HY염서L	HY염서M	HY염서M	HY올림도M
휴먼마미체	휴먼모음T	휴먼엑스포	휴먼매직체	휴먼고딕
문체부 바탕체	문체부 돋움체	문체부 제목 돋움체	문체부 제목 바탕체	각진제목체
가난상체	가난상체	바탕한체	굵은바탕한체	가늌바탕한체
굵은돋움한체	양재 소슬	양재 트른B	양재 참숯B	양재 둥기

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

## ■ 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	IIIIIIIIII	HHHHHH
	바탕	IIIIIIIIII	HHHHHH
	굴림	IIIIIIIIII	HHHHHH
고정간격 폰트	바탕체	IIIIIIIIII	HHHHHH
	굴림체	IIIIIIIIII	HHHHHH



## ■ 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)	<ul style="list-style-type: none"> <li>• 문자의 끝부분을 빠치게 만든 것</li> <li>• 텍스트의 본문에 적합 예 Times, Bookman, Palatino</li> </ul>
산세리프(Sans Serif)	<ul style="list-style-type: none"> <li>• 글자의 끝부분에 빠침이 없는 것</li> <li>• 제목, 강조될 문장에 적합 예 Helvetica, Arial, Optima, Avant Garde</li> </ul>
타입 스타일	<ul style="list-style-type: none"> <li>• 굵음(boldface), 이탤릭(italic) 등</li> </ul>
타입 스타일 속성	<ul style="list-style-type: none"> <li>• 밑줄(underlining), 외곽선(outlining) 등</li> </ul>
타입 크기	<ul style="list-style-type: none"> <li>• 일반적으로 포인트(pt) 단위로 표현하며, 1pt=1/72인치</li> <li>• 문자의 상단에서 하단까지의 거리를 말함(줄 간격을 위한 약간의 여백 포함)</li> </ul>

### ■ 타입페이스(Typeface)

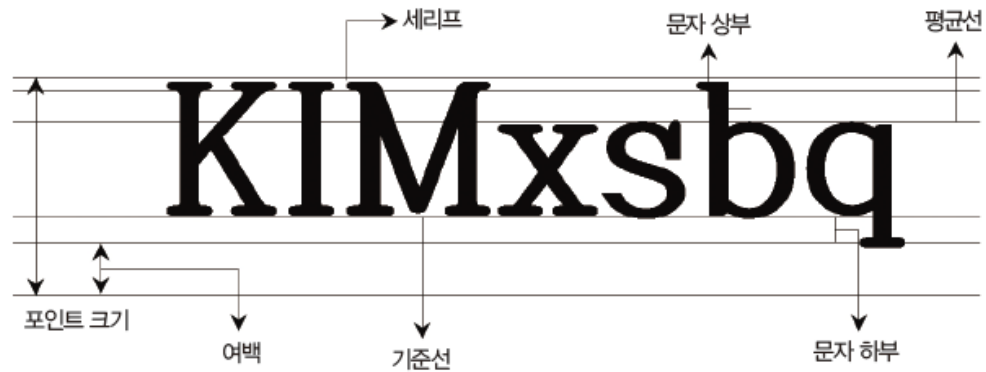


그림 4-12 타입페이스의 구성 요소

## 2.3 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
- 색체를 표현할 경우에는 더 많은 정보가 필요함

### ■ 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

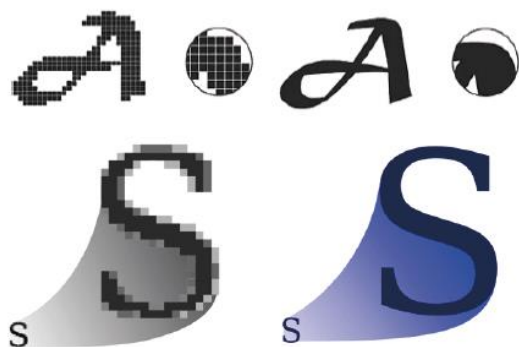


그림 4-13 비트맵과 벡터의 개념

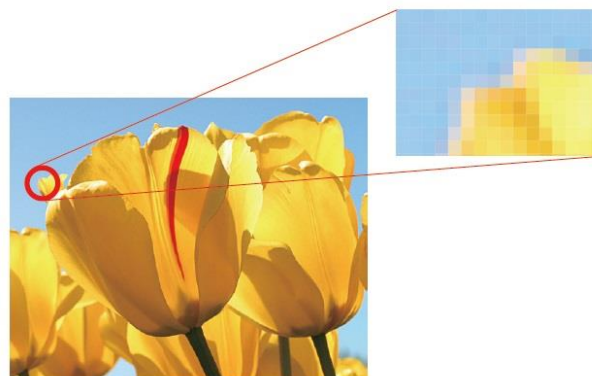


그림 4-14 계단현상

## 2.3 비트맵 폰트와 벡터 폰트

### ■ 비트맵 방식의 장단점

표 4-4 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 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)

## 2.3 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

표 4-5 벡터 방식의 장단점

장점	<ul style="list-style-type: none"><li>• 크기나 해상도에 영향을 받지 않음</li><li>• 글자나 이미지를 확대하거나 축소해도 일그러짐 없이 깨끗한 모양을 유지함</li><li>• 글자나 이미지를 표현하기 위한 좌표만 저장하기 때문에 메모리 용량을 적게 차지함</li></ul>
단점	<ul style="list-style-type: none"><li>• 복잡한 수학적 알고리즘을 가지고 있기 때문에 크기에 상관없이 알고리즘을 저장할 메모리 공간이 필요함</li><li>• 글자 또는 이미지를 표현하는 속도는 비트맵에 비하여 떨어짐</li></ul>

## 3.1 텍스트의 저장 형식

### ■ 메모장(txt)

- 1985년 마이크로소프트사의 윈도우 1.0부터 시작하여 현재까지 사용되고 있음
- 확장자는 \*.txt
- 저장 문서에 대한 어떠한 정보나 포맷도 가지고 있지 않아 거의 대부분의 문서 파일 포맷에 대하여 편집 가능
- 문서 파일 형식 중에서 가장 호환성이 높음

### ■ 워드프로세서(hwp, docx)

- 가장 많이 사용하는 것은 마이크로소프트사의 MS워드와 한글과컴퓨터사의 한글
- 워드프로세서 파일의 포맷은 본문의 내용뿐만 아니라 폰트나 서식 정보 등도 함께 포함하고 있음
- 하나의 워드프로세서 파일을 다른 워드프로세서에서 사용하려면 해당 포맷으로 변경해야 함

표 4-6 MS 워드와 한글의 주요 특징

MS 워드	<ul style="list-style-type: none"><li>• 확장자 *.docx로 저장됨</li><li>• 작성한 문서를 PDF 또는 XPS(XML Paper Specification) 형식으로 저장 가능</li><li>• MS 오피스의 MS 워드, 엑셀, 파워포인트 등이 문서 표준을 공개하여 모든 워드프로세서 및 문서편집기에서 호환이 가능함</li><li>• 전 세계적으로 시장 점유율 높음</li></ul>
한글	<ul style="list-style-type: none"><li>• 확장자 *.hwp로 저장됨</li><li>• 한글 워드프로세서의 시초는 1989년에 개발한 '아래아한글 1.0'으로 고어를 포함한 완벽한 한글을 구현함</li><li>• 작성한 문서를 PDF, XML 형식으로 저장 가능</li><li>• MS 오피스가 점령하지 못한 유일한 나라가 될 정도로 토종 워드프로세서로서 성능과 안정성이 뛰어남</li></ul>

## 3.2 워드프로세서

### ■ 워드프로세서

#### :: 여기서 잠깐 워드프로세서의 진화

초기 워드프로세서는 1964년 IBM이 마그네틱 테이프를 탑재한 셀렉트릭(Selectric) 타자기를 워드프로세싱 머신으로 부른 것이 시초이다. 워드프로세싱 머신은 타자기에 저장 장치를 탑재하여 편집 기능과 인쇄 기능을 제공하였다. 이후 비슷한 기능을 가진 타자기들이 등장하면서 기존의 타자기와 구분하기 위해 '워드프로세서'라는 용어를 사용하기 시작했다. 일반에게 컴퓨터가 보급되기 시작하면서 1976년 기존의 타자기 대신 소프트웨어 형태의 워드프로세서가 등장했다. 이후 다양한 워드프로세서가 출시되었으며 1983년 마이크로소프트사의 MS 워드가 출시되었다. MS 워드는 도스 운영체제 환경에서는 고전했지만 윈도우로 전환되면서 시장을 장악하여 현재 전 세계적인 사용자를 보유하고 있다.



#### >> 워드프로세서의 진화



## 3.3 PDF(pdf)

### ■ PDF

- 어도비사(Adobe Systems)에서 만든 문서파일 포맷
- 대부분의 운영체제에서 호환되어 읽기가 가능
- 확대나 축소를 해도 해상도가 변하지 않음
- 자체적인 압축 기능을 포함하고 있어 온라인 및 오프라인 환경에서 문서를 쉽게 공유할 수 있음
- 보안성이 높음

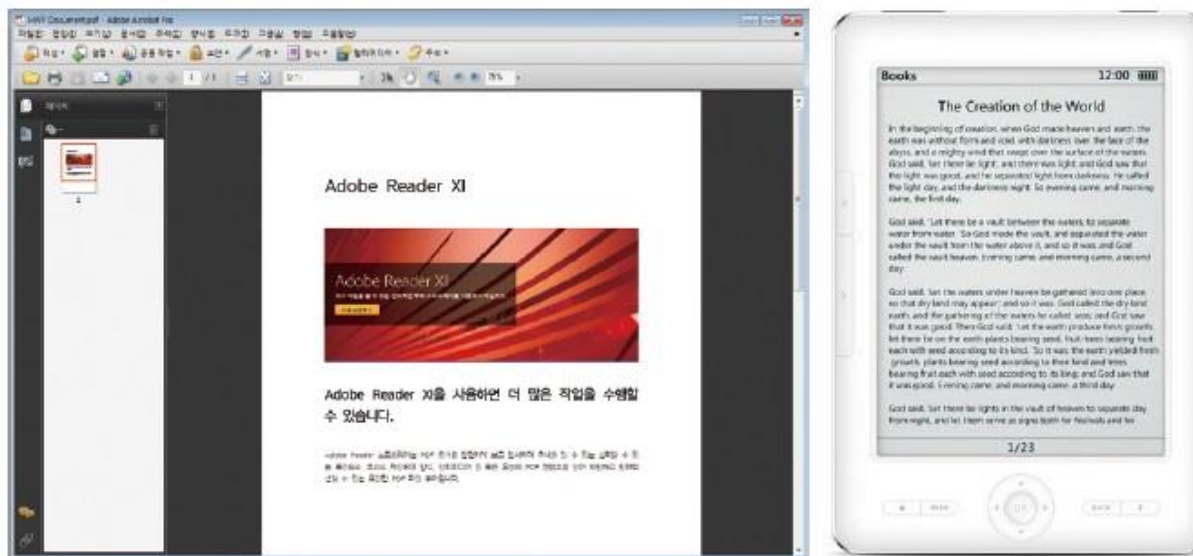


그림 4-16 PDF의 사용 예

## 3.3 PDF(pdf)

### ■ PDF의 장점

- 문서의 호환성이 높고 파일의 무결성을 가진다
  - 어떤 프로그램으로 PDF를 작성하더라도 파일 정보가 그대로 유지됨
- 문서의 사용과 관리가 편리하다
  - 자체 압축 기능이 있어 다른 파일에 비해서 상대적으로 용량이 작음
  - 이미지, 그래픽, 텍스트 등과 같은 정보를 하나의 파일에 포함(임베딩, Embedding)함
  - 책갈피 및 링크 기능을 첨가하여 원하는 부분을 쉽게 찾을 수 있음
- 문서 보안이 뛰어나다
  - 문서에 암호를 설정하는 기능을 제공



그림 4-17 PDF 문서의 변환

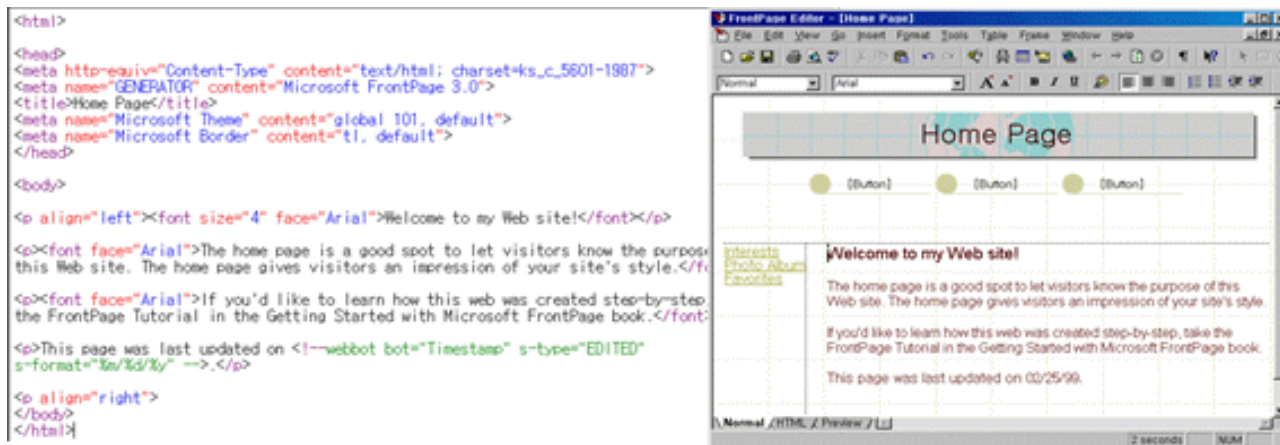
### ■ PDF의 단점

- 문서의 제작 방식이 동일하지 않고 다양한 파일 형식이 존재한다
  - 제작 방식에 따라 텍스트 PDF 파일과 이미지형 PDF 파일이 존재함
- 편집이 불편하다
  - 장점인 보안성 때문에 편집이 불가능하거나 힘들
  - PDF 파일은 하나의 커다란 이미지 형태이기 때문에 일부를 수정하는 데 제한이 따름
- 모니터에서 가독성이 낮다
  - 대부분의 PDF 파일은 인쇄하기 쉽게 A4나 A3 규격에 맞춘 것이 대부분이라 모니터에서는 화면에 꽉 차지 않음

## 3.4 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
- 웹 브라우저를 통해 볼 수 있는 대부분의 웹페이지들은 HTML로 작성
- 확장자는 \*.htm 또는 \*.html
- 웹 브라우저상에 정보를 표시하기 위해 마크업 심볼의 집합으로 구성됨
  - 마크업 : 특정 위치에 삽입되는 문자(명령어)나 기호를 의미, 태그라고도 함



## 3.3 웹 기반 문서(html, xml)

### ■ 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	 	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		

## 3.4 웹 기반 문서(html, xml)

### ■ 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

## 3.4 웹 기반 문서(html, xml)

### ■ 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, 재가공이 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

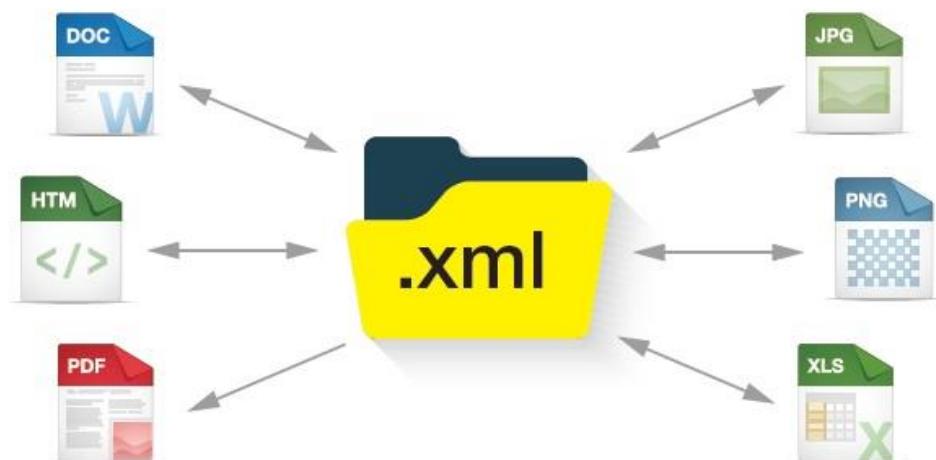


그림 4-19 XML 문서의 변환

## 3.5 스마트폰에서 지원하는 텍스트 파일 형식

### ■ 스마트폰에서 문서 보기

- 텍스트, 웹페이지, 워드, 파워포인트, 엑셀, PDF 등은 별도의 애플리케이션 없이 iOS, 안드로이드 모두 미리보기 가능
- 한글 문서는 안드로이드는 미리보기 가능하지만, iOS에서는 폴라리스 오피스나 한컴오피스 등을 이용해야 미리보기 가능

### ■ 스마트폰에서 문서 편집하기

- iOS : iWorks(Pages, Keynote, Numbers), 폴라리스 오피스, 오피스2 플러스 같은 애플리케이션을 이용해 편집
- 안드로이드 : 씽크프리(Thinkfree), 폴라리스 오피스 같은 애플리케이션을 이용해 편집
- 한글 문서는 한컴오피스, 폴라리스 오피스, 씽크프리 등을 이용해 편집

### ■ 스마트폰에서 전자책 보기

- 어도비 뷰어(Adobe Viewer)를 사용하면 인디자인(Adobe InDesign) 프로그램으로 만든 전자책을 볼 수 있음



## ■ 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

▶ 동영상 보기 : [생활 속에 들어온 최첨단 기술 - 문자인식기술](#)

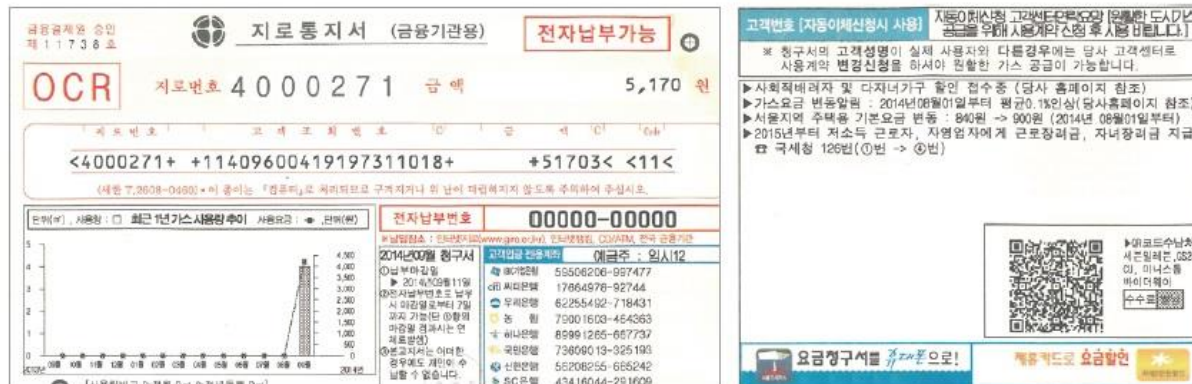


그림 4-20 OCR 문자와 2D 바코드 사용의 예

# 4.1 문자 인식 기술

## ■ 문자 인식(Character Recognition) 기술의 개요



▶ 동영상 보기 : [OCR\(Optical Character Recognition\)](#)

## ■ 광학 문자를 표현하는 폰트의 종류

### :: 여기서 잠깐 광학 문자를 표현하는 폰트의 종류

광학 문자 인식(OCR) 기술은 컴퓨터 판독과 육안 판독을 동시에 할 수 있는 텍스트를 사용한다. 1960년대부터 시작된 OCR 기술은 현재에는 필기체 글씨를 인식할 수 있는 정도까지 발전했지만 초기에는 컴퓨터가 인식할 수 있는 폰트를 사용해야 했다. 이러한 필요성에 의해 OCR-A와 OCR-B가 탄생하였고 이후 MICR E-13B(Magnetic Ink Character Recognition E-13B), SEMI M12 등이 나타났다.

- OCR-A : 1965년에 컴퓨터가 식별할 수 있도록 최대한 단순한 형태로 알파벳을 만들기 위하여 가로 4, 세로 7의 일정한 격자에 의해 표현된 매우 부자연스러운 모양의 폰트이다. 영문자, 숫자와 20여 개 특수기호의 형상과 크기 등이 규정되어 있다. 문자의 형태가 꽤 부자연스런 경우가 있으며 글자 형태의 특징이 과장되어 있다.
- OCR-B : OCR-A보다 4배 조밀한 격자를 기반으로 만들어져 직선적 OCR-A에 비해 기존의 활자에 가까운 균형적 형태를 나타낸다.
- MICR E-13B : 자성을 지닌 특수 잉크로 기록된 문자를 읽어서 전기 신호로 변환하는 폰트를 말한다. MICR 문자는 판독 직전에 문자화시켜 자기 헤드로 문자를 판독한다. MICR 기술은 은행에서 주로 사용되는데 수표 아랫부분의 숫자와 문자들을 말한다.
- SEMI M12 : 반도체 제조 산업에서 반도체를 만들기 위한 얇은 판인 웨이퍼(Wafer)와 반도체 칩과 외부 회로를 연결시켜 주는 전선(Lead)을 고정시키는 리드 프레임(Lead Frame)에 표기하기 위해 사용된다.

OCR-A	OCR-B
ABCDEFGHIJKLMNPQRSTUVWXYZ 0123456789 #&()*+,-./<>@\`£¥	ABCDEFGHIJKLMNPQRSTUVWXYZ 0123456789 #\$%&()*+,-./<>@\`£¥
MICR E-13B	SEMI M12
0123456789 * ~ ¢ ¤ ¥	ABCDEFGHIJKLMN PQRSTUVWXYZ- 0123456789.

### >> 광학 문자를 표현하는 폰트의 종류

## 4.2 문자 인식 장치와 문자 인식 소프트웨어

### ■ 문자 인식 장치

- OCR 판독기와 소프트웨어로 문자를 판독하고 식별하여 컴퓨터가 이해할 수 있는 코드로 변환시키는 장치
- 인식의 대상이 되는 문자는 보통 식별하기 유리한 형태로 규격화되어 있음
- 광전변환장치, 인식처리장치, 기억장치, 출력장치로 구성됨



그림 4-22 문자 인식 절차

그림 4-23 네이버랩의 문자 인식 서비스

## 4.3 문자 인식 응용 기술

### ■ 필기체 문자를 인식하는 과정

- ⊖ 문자의 변형 형태를 일정한 형태로 정규화하는 전처리 작업
- ⊖ 기계 학습 알고리즘을 사용하여 문자의 특징을 추출하고 형태를 분류함
- ⊗ 문자나 단어의 사용 빈도, 언어의 형태학적 정보를 이용하여 문자를 인식함

### ■ 필기체 문자의 정확도를 향상시키기 위한 지표

- 문자의 특징적인 선을 명확하게 구별하여 씀
- 자연스러운 글자 형태를 사용함
- 문자 기입란의 크기와 모양을 동시에 제한하는데, 여기에 맞추어 씀

## 4.3 문자 인식 응용 기술

### ■ C-Pen

- 아라미디어(Aramedia.com)에서 만든 손바닥 크기의 스캐너로 블루투스와 USB 연결이 가능함
- 스캔한 문자들을 내장된 OCR 기능을 사용하여 PC의 소프트웨어로 전송함

▶ 동영상 보기 : [C-Pen](#)

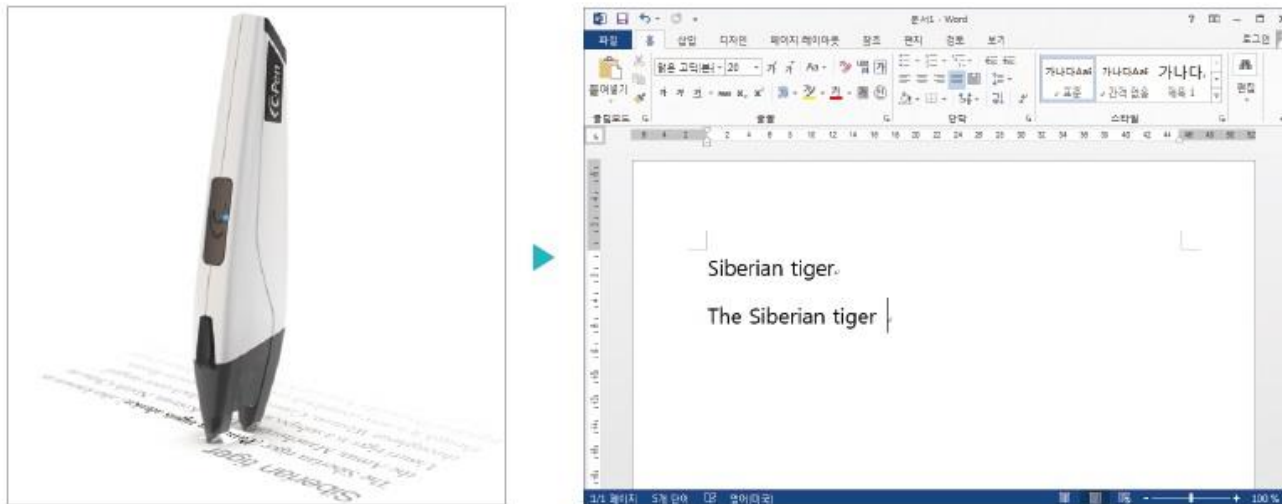


그림 4-24 C-Pen의 문자 인식 가능



## 4.3 문자 인식 응용 기술

### ■ 모바일 문자 인식

- 문자 인식 애플리케이션을 이용하여 카메라로 원하는 대상을 촬영한 후 영역을 지정하는 방식

▶ 동영상 보기 : [OCR Test App : OCR on Android](#)



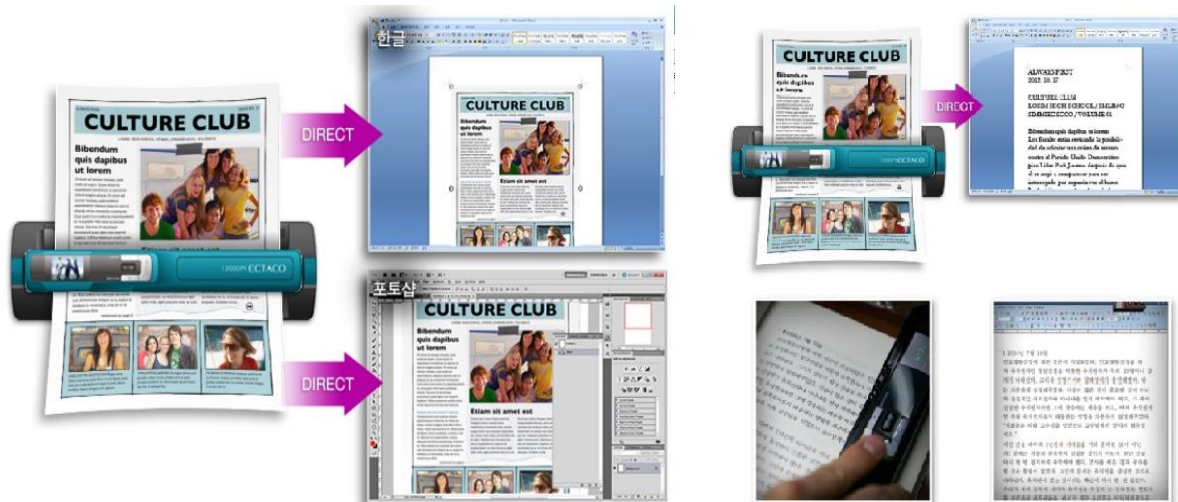
그림 4-25 모바일 문자 인식 사례



## 4.3 문자 인식 응용 기술

### ■ 휴대용 스캐너의 문자 인식

- 스캔과 동시에 이미지를 그림판, 포토샵, 한글 등에 삽입



▶ 동영상 보기 : [휴대용 스캐너](#)

### ■ 스마트 기기와 웨어러블 기기에서 문자 인식

- 스마트 기기에서 문자 인식 : 갤럭시 노트3 화면의 S펜 버튼을 누르면 나타나는 '액션 메모' 기능
- 웨어러블 기기에서 문자 인식 : 삼성의 갤럭시 기어와 구글의 구글 글래스에 탑재될 광학 문자 인식 기술

# 5.1 전자책

## ■ 종이책

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

## ■ 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

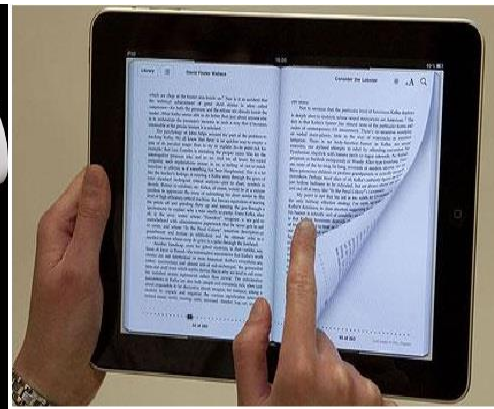
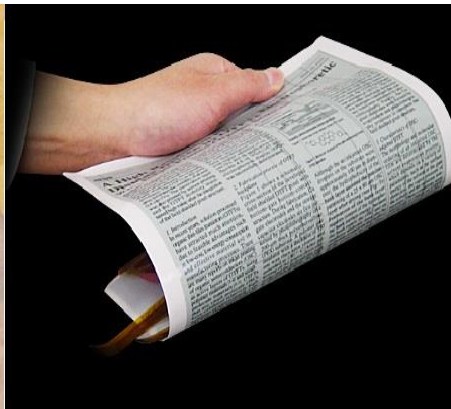
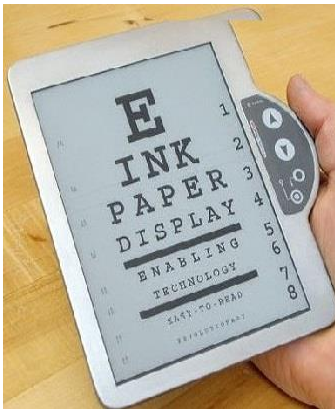


▶ 동영상 보기 : [E-Book vs Book](#)

▶ 동영상 보기 : [전자책과 종이책의 차이 무엇을 읽을](#)

## ■ 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



▶ 동영상 보기 : [E-Paper](#)

▶ 동영상 보기 : [E-Ink](#)

## ▲ 전자 잉크와 전자종이

## ■ 전자책이 미디어 환경에 미치는 영향

- 읽기의 부환과 출판 산업의 활성화
- 데스크톱, 태블릿, 스마트폰의 역할이 기능적으로 분업화
- 새로운 글쓰기 공간의 창출