

컴퓨터가 문자를 이해하는 방법

# 컴퓨터는 한글을 어떻게 인지할까?

가



# 컴퓨터는 한글을 어떻게 인지할까?

가



B0A1

# 비트 (bit)

4bit

1	0	1	0
---	---	---	---

1bit   1bit   1bit   1bit

1이나 0을 저장할 수 있는 단위

# 바이트 (byte)

8bit = 1byte

1	0	1	0	1	0	1	0
---	---	---	---	---	---	---	---

1bit   1bit   1bit   1bit   1bit   1bit   1bit   1bit

8개의 bit를 묶어 256개 까지 표현하는 단위

컴퓨터는 한글을 Byte코드로 인지한다.

가



B0A1

2byte

## 2진수와 16진수

1	0	1	1	0	0	0	0
---	---	---	---	---	---	---	---

=

B0

1	0	1	0	0	0	0	1
---	---	---	---	---	---	---	---

=

A1

“가”

인코딩



# 인코딩의 방법

EUC-KR

MS949

UTF-8

# EUC-KR

## 한글을 2byte로 표현

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
b0a0	가	각	간	강	갈	감	갸	갇	갸	갸	갸	갸	갸	갸	갸	갸
b0b0	갈	갸	갸	개	객	갸	갸	갸	갸	갸	갸	갸	갸	갸	갸	갸
b0c0	갸	갸	개	갸	갸	거	격	건	견	겔	겔	겔	겔	겔	겔	겔
b0d0	겔	겔	겔	겔	게	겐	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔
b0e0	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔
b0f0	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔	겔
b1a0		괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘
b1b0	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘
b1c0	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘
b1d0	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘
b1e0	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘	괘

EUC-KR 코드표 일부

# 2350개

MS949

EUC-KR의 확장판

EUC-KR  $\Rightarrow$  MS949

“힉뵐”

MS949

EUC-KR의 확장판

EUC-KR  $\Rightarrow$  MS949

“??”

MS949

MS949  $\Rightarrow$  EUC-KR

“한글”

MS949

MS949  $\Rightarrow$  EUC-KR

“??”

MS949

확장이라면서 왜?

MS949

형, 이나 뵤은 EUC-KR은 표현할 수 없다



# MS949

1. **헝 or 뷔크** ⇒ EUC-KR ⇒ “???” ⇒ MS949

이미 byte배열이 깨져있다

2. **헝 or 뷔크** ⇒ MS949 ⇒ **헝, 뷔크** ⇒ EUC-KR

결국 EUC-KR이 읽지 못함

# 인코딩 방식의 중요성

IF...

레거시 DB ⇒ US7ASCII, EUC-KR

백엔드 ⇒ UTF-8, MS949

웹 ⇒ UTF-8

제대로 읽어올 수 있을까?

UTF-8

“유니코드 문자열(Charset)”

# UTF-8

	AC0	AC1	AC2	AC3	AC4	AC5	AC6	AC7	AC8	AC9	ACA	ACB	ACC	ACD	ACE	ACF
0	가 AC00	감 AC10	갸 AC20	갯 AC30	갈 AC40	각 AC50	갬 AC60	거 AC70	검 AC80	겐 AC90	갯 ACA0	결 ACB0	격 ACC0	겟 ACD0	고 ACE0	곰 ACF0
1	각 AC01	갑 AC11	갬 AC21	갯 AC31	갯 AC41	갯 AC51	겟 AC61	걱 AC71	겁 AC81	갯 AC91	겟 ACA1	겟 ACB1	겟 ACC1	겟 ACD1	곡 ACE1	곱 ACF1
2	갯 AC02	갯 AC12	갯 AC22	갯 AC32	갯 AC42	갯 AC52	갯 AC62	갯 AC72	갯 AC82	갯 AC92	갯 ACA2	갯 ACB2	갯 ACC2	갯 ACD2	곡 ACE2	곶 ACF2
3	갯 AC03	갯 AC13	갯 AC23	갯 AC33	갯 AC43	갯 AC53	갯 AC63	갯 AC73	갯 AC83	갯 AC93	갯 ACA3	갯 ACB3	갯 ACC3	갯 ACD3	곶 ACE3	곶 ACF3
4	간 AC04	갯 AC14	갯 AC24	갯 AC34	갯 AC44	갯 AC54	갯 AC64	갯 AC74	갯 AC84	갯 AC94	갯 ACA4	갯 ACB4	갯 ACC4	갯 ACD4	곶 ACE4	곶 ACF4
5	갯 AC05	강 AC15	갯 AC25	갯 AC35	갯 AC45	갯 AC55	갯 AC65	갯 AC75	갯 AC85	갯 AC95	갯 ACA5	갯 ACB5	갯 ACC5	갯 ACD5	곶 ACE5	곶 ACF5
6	갯 AC06	갯 AC16	갯 AC26	갯 AC36	갯 AC46	갯 AC56	갯 AC66	갯 AC76	갯 AC86	갯 AC96	갯 ACA6	갯 ACB6	갯 ACC6	갯 ACD6	곶 ACE6	곶 ACF6

유니코드 표

코드 포인트에 U+ (lg 아님 ㅋ)붙여서 사용

# UTF-8

UTF-8

UTF-16

UTF-32

저장하는 byte 크기의 차이

# UTF-8

2~4byte

2, 4byte

4byte

UTF-8

UTF-16

UTF-32

저장하는 byte 크기의 차이

# UTF-8

코드포인트 범위	인코딩 방법	크기
U+0000 ~ U+007F	그대로 인코딩	1 byte
U+0080 ~ U+07FF	110xxxxx 10xxxxxx	2 byte
U+0800 ~ U+FFFF	1110xxxx 10xxxxxx 10xxxxxx	3 byte
U+10000 ~ U+1FFFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx	4 byte

[ 표 ] 코드포인트 범위에 따른 UTF-8 인코딩 방법

가 ⇒ U+AC00    1010110000000000

1110101010110000100000000(3byte)

끝