Encoding, Decoding 학습

스크래핑 센터

최진영

|  |  |
| --- | --- |
| Encoding/Decoding | |
| *Encoding* | 컴퓨터는 문자를 인식할 수 없으므로 숫자로 변환되어 저장됨, 변환해주기 위해서는 기준이 있어야함, 그것을 문자코드라고 하며 대표적으로 ASCII코드 또는 유니코드가 있다. |
| *Decoding* | 복호화 또는 디코딩(decoding)은 [부호화](https://ko.wikipedia.org/wiki/%EB%B6%80%ED%98%B8%ED%99%94)(encoding)된 정보를 [부호](https://ko.wikipedia.org/wiki/%EB%B6%80%ED%98%B8)(code)화되기 전으로 되돌리는 [처리](https://ko.wikipedia.org/wiki/%EC%A0%95%EB%B3%B4_%EC%B2%98%EB%A6%AC) 혹은 그 처리 방식을 말한다. 보통은 부호화의 절차를 역으로 수행하면 복호화가 된다. |
| *아스키(ASCII)코드* | 미국 ANSI에서 표준화한 정보교환용 7비트 부호체계입니다.  000(0X00)부터 127(0X7F)까지 총 128개의 문자 집합을 제공하며, 처음 32개는 인쇄와 전송 제어용으로 사용되는 제어 문자(Control Characters)이고 이를 제외한 33번째 이후의 문자들에 대해 숫자 및 영문 대소문자가 배치되어 사용됩니다. (127번째 마지막 문자 DEL은 제외)  일반적으로 데이터는 byte 단위로 다뤄지는데 ASCII는 1바이트를 구성하는 8비트 중에서 7비트만 사용하고 나머지 1비트는 통신 에러 검출을 위해 사용합니다. 이 비트를 Parity Bit 라고 하는데 현재는 더 이상 사용되지 않습니다.  ASCII코드는 매우 단순하고 간단하여 어느 시스템에서도 적용 가능하다는 장점이 있으나 2바이트 이상의 코드를 표현할 수 없다는 단점이 있습니다. |
| *유니코드* | 확장 ASCII와 같이 영어 이외의 언어를 사용하기 위해 국가별 인코딩 표준이 만들어져 사용됐지만, 인터넷의 발달로 인한 다른 지역간의 다른 언어를 사용한 문서 교환이 활발해지면서 문제점이 나타나게 되었습니다. 이러한 문제를 해결하기 위해 모든 문자 체계를 하나의 문자 집합으로 만든 것이 **유니코드**입니다.  유니코드는 먼저 포함시키고자 하는 문자 집합을 정의하였는데 이것을 문자 셋 또는 캐릭터 셋(character set)이라고 합니다. 또한 이것을 UTF(Unicode Transformation Format) 이라고 하며 여기에 번호를 붙인 것이 유니코드 인코딩입니다.  **UTF-8 :** 하나의 문자를 1~4바이트의 가변길이로 표현. 1바이트 영역은 ASCII코드와 하위 호환되며 ASCII코드의 128개 문자 집합은 UTF-8과 동일하게 호환됨. 현재 인터넷에서 가장 많이 쓰이는 인코딩이며 뛰어난 크로스플랫폼 호환성도 갖고 있음.  **UTF-16 :** 모든 문자를 2바이트의 고정크기로 표현하고 UTF-8과 마찬가지로 ACII코드의 128개 문자 집합에 대해 호환성을 가짐. 바이트 순서가 정해지지 않아 리틀/빅 엔디안 문제가 발생하기 때문에 인터넷 상에서의 사용을 권고하지 않음.  Java와 .NET Framework의 기본 인코딩.  인터넷에서는 UTF-8 인코딩을 많이 사용합니다. 인터넷에서는 전송속도가 중요한 기준이 되는데 이에 맞춰서 문서의 크기를 작게 변환해주고, 크로스플랫폼 호환성을 통해 엔디안에 상관없이 사용 가능하다는 점이 많이 사용되는 이유입니다. |
| *조합형* | [한글 인코딩](https://namu.wiki/w/%ED%95%9C%EA%B8%80%20%EC%9D%B8%EC%BD%94%EB%94%A9) 방식 중 하나. 한글의 제자원리에 부합하여 표현이 자유롭다는 장점이 있어 한때 많이 쓰였고 또 많은 사용자의 지지를 받았던 방식이지만 지금은 [유니코드](https://namu.wiki/w/%EC%9C%A0%EB%8B%88%EC%BD%94%EB%93%9C)의 등장으로 필요성이 크게 줄어 [옛한글](https://namu.wiki/w/%EC%98%9B%ED%95%9C%EA%B8%80)의 표기 이외에는 거의 쓰이지 않는다.  코드에는 자모만을 배당해 두고, 자모의 조합으로 한글을 표현한다. 글자 하나를 만들 때 초성+중성+종성을 조합해서 만드는 것. 가령 '만'이라는 글자를 입력하면 ㅁ+ㅏ+ㄴ 해서 '만'이 되는 식이다. 이 때 자모 단위로 특정한 값을 가지고 이러한 값들의 나열로부터 글자를 조합하기 때문에 조합형이라고 한다.  완성형 글자로는 표현할 수 없는, 160만개가 넘는 [한글 자모](https://namu.wiki/w/%ED%95%9C%EA%B8%80%20%EC%9E%90%EB%AA%A8)로 결합한 모든 글자를 표현할 수 있다는 장점을 지닌다. 단점은 사용 빈도와 용량 효율성. 조합형 한글자보다 완성형 한글자가 용량을 적게 차지하는건 둘째치고, 현대 한글은 1만자 정도에서 해결이 되며 의성어 등을 빼면 실제 3천여 자 정도, 고문헌에 사용된 글자 수는 5천여 자 정도밖에 안된다. 그리고 엄연한 표준 코드이다. . |
| *완성형* | [한글 인코딩](https://namu.wiki/w/%ED%95%9C%EA%B8%80%20%EC%9D%B8%EC%BD%94%EB%94%A9) 방식 중 하나. 현재 산업 표준이기도 하며, 표준 번호는 KS X 1001(옛 번호 KS C 5601)이다. 1974년에 처음으로 제정되었고, 1987년 버전(KS C 5601-1987)이 널리 알려져있다. 인코딩은 EUC-KR(Code Page 51949)로 이루어진다. 이 문서에서는 이 KS X 1001 표준에 대해 주로 서술한다.  글자 코드 세트에 쓸 문자를 현대 한국어에서 빈도가 높은 한글만 추려내어, '가'부터 '힝'[[1]](https://namu.wiki/w/%EC%99%84%EC%84%B1%ED%98%95#fn-1)까지 죄다 배당한 후 문자를 기록하는 방식으로, 가령 '속'이라는 글자를 기록한다면 가, 각, 간, 갇, 갈, 갉, 갊…으로 쭉 찾아가다가 0xBCD3로 표현하는 방식이다.  장점 : 외국어나 특수문자와 같이 이용하는 것을 미리 가정하고 만들었기 때문에 [조합형](https://namu.wiki/w/%EC%A1%B0%ED%95%A9%ED%98%95)에 비해 국제 표준과 충돌이 적다는 장점이 있다.  단점 : 미리 조합되어 있는 글자 외의 문자는 어떻게 해도 표시할 수 없다 |