문자 집합, 인코딩 그리고 유니코드

SPARCS

박용수 <pcpenpal@sparcs.org>

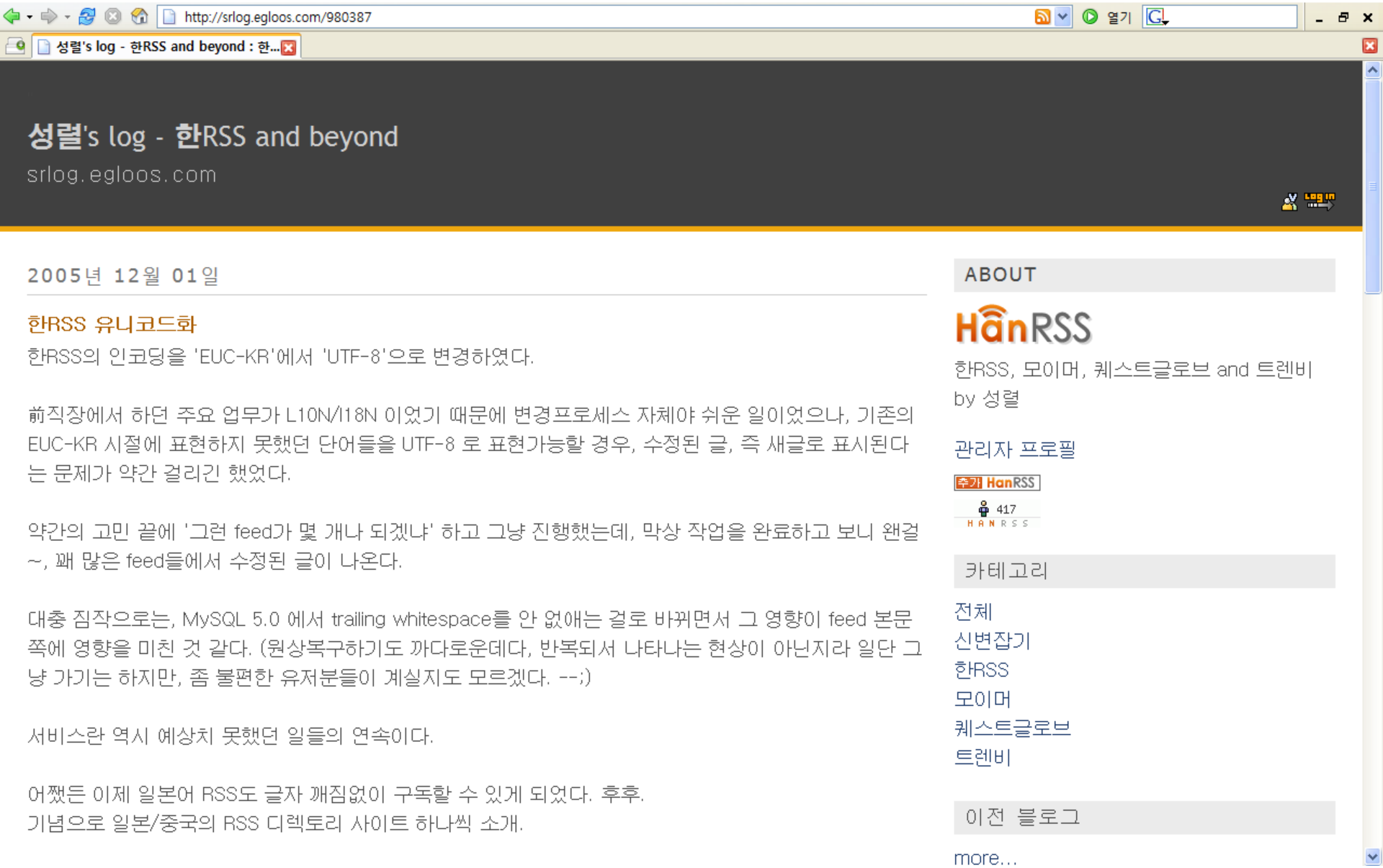
목차

- 도입
- 문자 집합
- 인코딩
- 유니코드
- 유니코드의 인코딩
- UTF-8

도입 (1)

- 블로고스피어의 수많은 이야기들


http://channy.creation.net/blog/?p=350
열기
Channy's Blog » Daum 유니코드(U...
Channy's Weblog
"웹과 일상에 대한 이야기들"
차니의 주요 관심사인 인터넷과 웹기술
신앙과 창조론, 지질학 등의 이야기입니다
1st Blog
2nd Blog
KoreaCrunch
Daum 유니코드(UTF-8)에 도전하다
Tags: Daum , Unicode
지난 주 목요일에 Daum의 커뮤니티 섹션(카페, 블로그, 플래닛)의 개편이 있었습니다. 그냥 보기엔 일상적인 디자인과 콘텐츠 개편으로 보입니다만 실제 중요한 이슈 한 가지가 숨겨져 있습니다. 그것은 다름 아닌 모든 페이지가 유니코드(UTF-8) 인코딩 지원으로 변경된 것입니다. 국내 웹 페이지들과 데이터 대부분은 지금까지 한글 완성형 코드(KSC5601)를 표현하는 EUC-KR 인코딩 방식을 주로 사용해 왔습니다. 따라서 UTF-8 지원이 대규모 웹 서비스에서 이루어진 것은 꽤 고무적이라고 볼 수 있습니다.
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ko">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="autocomplete" content="off">
<title>Daum 카페</title>
<link rel="shortcut icon" href="http://www.daum.net/daum2.ico">
물론 Daum에서 UTF-8을 사용한 첫 서비스는 아닙니다. 협력 사진 공유 서비스인 Daum Pie가 신규 서비스로 개발될 작년 처음 UTF-8을 도입했습니다. 이번 커뮤니티 섹션 전체 적용에서는 이 사례와 경험을 기초로 작업해서 얻은 성과 입니다. 물론 개별 카페, 블로그, 플래닛과 데이터 전체가 전환된 것은 아니고 탑 섹션만 바뀌었지만 향후 UTF-8으로 가기 위한 초석을 쌓았다고 생각합니다.
UTF-8과 EUC-KR에 대한 장단점 문제는 오랫동안 논란이 되어 온 문제입니다. UTF-8 인코딩이한국어만 주로 표시하는 국내 웹 사이트에는 부적절하다는 이유도 있기도 합니다. 그러나, EUC-KR을 사용하면 '아햏햏', '뛔' 같은 단어나 'ᄉᆞ랑ᄒᆞ다' 같은 조선 시대 고어(古語) 등 우리말인데도 표시할 수 없다는 문제점이 있습니다. 이런 이유로 네이버 국어 사전은 UTF-8을 인코딩을 사용하고 있습니다. 뿐만 아니라 세계화
Google 검색
1st Blog
2nd Blog
Web
이메일로 구독하기:
구독하기
Delivered by FeedBurner
본 사이트는 Google AdSense를 통한 타겟팅된 광고 게재를 권해드립니다
Google 툴바가 탑재된 Firefox로 더욱 향상된 웹 브라우징을 경험하세요.
Tags
Accessbility ActiveX Ajax Apple Ask Biology Blog Book Bookmark Browser Business Canvas Christian Christmas Congnamul Daum del.icio.us Desktop Developer DRM EbayDevCon Essay Etech Evangelist Event Family Flash


http://srlog.egloos.com/980387
열기
성렬's log - 한RSS and beyond : 한...
성렬's log - 한RSS and beyond
srlog.egloos.com
2005년 12월 01일
한RSS 유니코드화
한RSS의 인코딩을 'EUC-KR'에서 'UTF-8'으로 변경하였다.
前직장에서 하던 주요 업무가 L10N/I18N 이었기 때문에 변경프로세스 자체야 쉬운 일이었으나, 기존의 EUC-KR 시절에 표현하지 못했던 단어들을 UTF-8 로 표현가능할 경우, 수정된 글, 즉 새글로 표시된다는 문제가 약간 걸리긴 했었다.
약간의 고민 끝에 '그런 feed가 몇 개나 되겠냐' 하고 그냥 진행했는데, 막상 작업을 완료하고 보니 완걸~, 꽤 많은 feed들에서 수정된 글이 나온다.
대충 짐작으로는, MySQL 5.0 에서 trailing whitespace를 안 없애는 걸로 바뀌면서 그 영향이 feed 본문 쪽에 영향을 미친 것 같다. (원상복구하기도 까다로운데다, 반복되서 나타나는 현상이 아닌지라 일단 그냥 가기는 하지만, 좀 불편한 유저분들이 계실지도 모르겠다. --;)
서비스란 역시 예상치 못했던 일들의 연속이다.
어쨌든 이제 일본어 RSS도 글자 깨짐없이 구독할 수 있게 되었다. 후후.
기념으로 일본/중국의 RSS 디렉토리 사이트 하나씩 소개.
ABOUT
HanRSS
한RSS, 모이머, 퀘스트글로브 and 트렌비
by 성렬
관리자 프로필
추가 HanRSS
417
HANRSS
카테고리
전체
신변잡기
한RSS
모이머
퀘스트글로브
트렌비
이전 블로그
more...

http://neoocean.net/blog/index.php?pl=1043

밀피유의 주말한정 이야기☆ :: 유니...

밀피유의 주말한정 이야기☆

태그 | 방명록

유니코드 페이지가 갑자기 한가득 늘어났어요. :)

2006/01/01 15:24

사실 유니코드라고 표현하면 조금 이상하기는 하지만, 일단 UTF-8을 유니코드라고 표현하기로 해 보겠습니다. 그러니까, 한 1년쯤 전만 해도 분명 유니코드로 개인 페이지들이 만들어지려면 꽤 오랜 시간이 걸릴 것이고, 꽤 오랜 시간이 걸려도 제대로 변환이 이루어지지 않으리라고 생각했는데, 뭔가 2006년을 기점으로 하루밤 사이에 수많은 페이지들이 한방에 유니코드로 바뀌는 모습을 보면서, 놀라기도 하고, 신기하기도 합니다.

태터툴즈 클래식은 1.0으로 가는 마이그레이션 단계 중에서 가장 골때리는 문제라고 생각되던 유니코드 변환 문제를 중간에 해결해 버린다는 의미가 가장 크다는 느낌입니다. 1.0으로 가는 마이그레이션 단계 중에서 유니코드로 변환하는 부분이 가장 크다고 생각하는데, 클래식에서 이렇게 한바탕 마이그레이션을 지나고 나면 앞으로 1.0으로 마이그레이션 하는 부담을 엄청나게 덜 수 있다고 생각합니다.

개인적으로 걱정한 부분은 일단 변환할 글 갯수도 많았지만, 방명록이나 RSS리더에 합해서 50메가에 달하는 텍스트가 들어가 있어서 '괜찮을까' 하는 점이었습니다. 백업을 하기는 했

by Miffy

공지 프로필.

검색

달력

« 2007/02 »

일	월	화	수	목	금	토
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

태그목록

서울 태그 UAC 환기

도입 (2)

- 요점은 무엇인가?
  - 컴퓨터에서 문자열을 어떻게 표현하고 다룰 것인가!
  - 모두가 만족스러운 솔루션은 무엇인가!

도입 (3)

- 컴퓨터란?
  - 정보 처리 기기
- (컴퓨터에서의) 정보란?
  - 비트들이 모여 나타내는 이산적 수치

도입 (4)

- 비트, bit
  - 0 혹은 1, 이진수 한 자리
  - 최소 단위 정보 크기
- 바이트, byte
  - 적당한 수의 비트를 선택해 만든 쓸만한 가장 작은 정보 크기
  - 컴퓨터가 바이트 단위로 일을 처리
  - 대세는 8비트
- 옥텟, octet: 8비트

도입 (5)

- 컴퓨터에서 수의 표현
  - 이진수 표현

$84_{(10)} = 01010100_{(2)}$

128	64	32	16	8	4	2	1
0	1	0	1	0	1	0	0

$= 54_{(16)}$

문자 집합 (1)

- 문자 집합, (coded) character set
  - 번호를 매긴 문자들의 집합
  - 어떤 범위의 수에 문자를 배정

- 문자 집합의 예
  - ASCII
  - KS X 1001 (KS C 5601)
  - Unicode

문자 집합 (2)

- ASCII
  - 1963, 'American Standard Code for Information Interchange'
  - 역사와 전통, 대세
  - 영어 알파벳, 숫자, 구두점 등 여러 기호
  - 0번부터 127번까지의 문자들 (7bit)

2^7-1

문자 집합 (3)

- ASCII 표

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	₩	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

인코딩 (1)

- 인코딩, encoding – 넓은 의미
  - 어떤 형태의 정보를 다른 형태로 바꾸는 것
  - 문자 집합도 일종의 인코딩
- 인코딩 – 문자 처리 관련 의미
  - 문자의 번호를 컴퓨터 정보의 표현으로 나타내는 방법
    - 번호를 바이트의 나열로 표현하는 방법

- 디코딩, decoding – 인코딩의 역과정

문자 집합 (4)

- ASCII 다시 보기
  - 이진수 7자리, 즉 7비트 내 표현 범위
  - 1바이트(1옥텟)에 저장하기 용이

- 문자 집합과 인코딩의 관계
  - 인코딩을 고려해서 문자 집합을 만든다

인코딩 (2)

- 적절한 문자 집합과 인코딩을 선택하면 컴퓨터를 이용해 문자열을 처리 및 저장할 수 있다

인코딩 (3)

ABCDEFG

10진수	16진수	문자
65	0x41	A
66	0x42	B
67	0x43	C
68	0x44	D
69	0x45	E
70	0x46	F
71	0x47	G

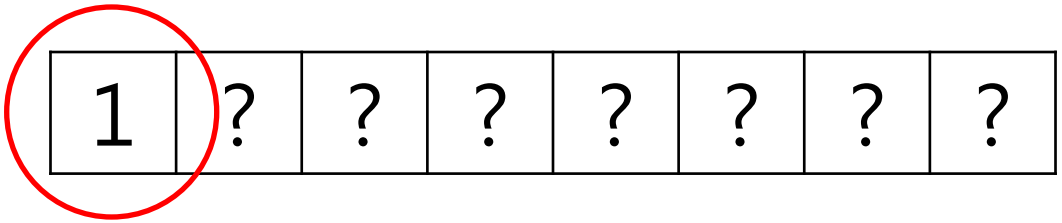
0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	1								

문자 집합 (5)

- 한글 및 한자는?
- KS X 1001
  - 1974, 한국 산업 규격, '정보 교환용 부호계'
  - KS C 5601
  - 94x94 행렬에 한글 2,350자, 한자 4,888자 등

인코딩 (4)

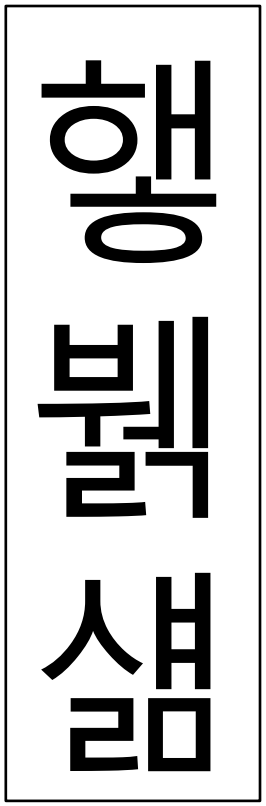
- EUC-KR
  - KS X 1001과 ASCII를 표현하는 인코딩
  - 멀티바이트 인코딩
    - 1바이트면 ASCII, 2바이트면 KS X 1001의 문자




```
if (c & 0x80) {
    /* KS X 1001 */
}
```

문자 집합 (6)

- KS X 1001 다시 보기
  - 한글 2,350자
    - 모든 한글 음절을 표현할 수 없다



유니코드 (1)

- 국제화 시대에 발맞춘 세상의 모든 문자를 다루는 문자 집합은 없을까!

- 유니코드, Unicode
  - 바로 그 것!

유니코드 (2)

- 유니코드 용어 - code point
  - (앞서 이야기하던) 문자의 번호
  - code point 표기법: U+AC00 – '가'

- 조합 가능한 모든 한글 음절 11,172자 전부 수록!
  - 11,172 = 19 * 21 * 28

유니코드 (3)

- code point의 범위
  - 0~1,114,112
  - U+0000~U+10FFFF

- BMP, basic multilingual plane
  - U+0000~U+FFFF
  - 현대 언어 및 잘 사용되는 모든 기호들 포함
  - U+0000~U+007F는 ASCII와 동일

유니코드의 인코딩 (1)

- UCS-4/UTF-32
- UCS-2
  - BMP를 2바이트로 표현
  - BMP 외의 문자는 UCS-2에서 표현 불가
- UTF-16
  - BMP는 2바이트로 표현
  - BMP 외의 문자는 두 개의 2바이트 표현으로
    - 각 2바이트 표현은 BMP에 있는 것처럼 보임

UTF-8 (1)

- 유니코드의 인코딩에 대한 생각
  - UTF-16은 문자가 최소 2바이트이므로 ASCII 표현에 비해 크기가 2배
  - 기존 ASCII 문서와의 호환성 없음

- UTF-8은 위 두 문제를 효과적으로 해결하는 인코딩 방법!

UTF-8 (2)

- 인코딩 방법
  - U+0000~U+007F: 1바이트 (ASCII 인코딩)
  - U+0080~U+07FF
    - 위 범위를 이진수로 풀면 0000 0yyy yyzz zzzz
    - 인코딩 결과는 [110y yyyy] [10zz zzzz]
  - U+0800~U+FFFF
    - 이진수 표현 xxxx yyyy yyzz zzzz
    - 인코딩 [1110 xxxx] [10yy yyyy] [10zz zzzz]
  - U+010000~U+10FFFF: 마찬가지

UTF-8 (3)

- 절묘한 비트 분배
  - 첫 바이트를 살펴보면 한 문자가 몇 바이트로 표현되었는지 알 수 있음
  - ASCII 코드는 ASCII 인코딩과 똑같이 표현됨
    - ASCII 문서는 UTF-8 문서
- 한글은 UTF-8로는 3바이트로 인코딩 됨
  - 한글 위주의 문서는 UTF-16에 비해 용량 면에서 불리

마무리

- 지금은 유니코드의 세상
  - Windows의 모든 내부 문자열은 UTF-16으로 처리
  - Java 등의 언어는 처음부터 유니코드를 고려하여 설계됨
  - 웹에 관련해서는 UTF-8 사용이 대세

참조

- 유니코드
  - http://www.unicode.org/
- 위키백과
  - Character Encoding

Q & A