What is Character Sets & Character Encodings & Character Reference ?

a. Character Set

Karakterlere karşılık teorik, soyut sayılar eşleşmesinin yapıldığı karakterler kümesine character set (charset) adı verilir.

Örneğin "Unicode (Universal Character Set)" karakterlerin benzersiz decimal sayılarla ifade edildiği karakterler setine denir. Mesela A = 65, B = 66, C = 67 ,... olarak karakter setinde yer alır. Bu şekilde örneğin “hello” string'i Unicode karakter setinde

104 101 108 108 111

h e l l o

ile karşılık görür. Unicode tüm dünyada kullanılmakta olan hemen hemen her bir harfi içeren karakter setine bir örnektir.

ASCII, internet tarihindeki ilk karakter seti standardıdır. Bu karakter seti tam olarak 128 farklı alphanumeric karakter tanımlar: Numbers (0-9), English Letters (A-Z) ve special Characters (örn; ! $+-()@<>).

ANSI (Windows-1252) Windows tarihindeki ilk karakter setidir. Tam olarak 256 farklı karakter tanımlar.

ISO-8859-1 (Latin-1) HTML 4 için ilk default karakter setidir. İngiliz alfabesi dışında "i", "ü", "ç",... gibi karakterleri de içerir. Tam olarak 256 farklı karakter tanımlar.

b. Character Encodings

Karakter setlerindeki harflerin karşılık geldiği teorik, soyut sayıların binary olarak nasıl tutulacağı yönündeki tekniğe / yönteme character encoding adı verilir.

Character Encoding karakterleri benzersiz binary sayılarla ifade eder. Örneğin UTF-8, UTF-16, UTF-32, ... birer character encoding'tirler.

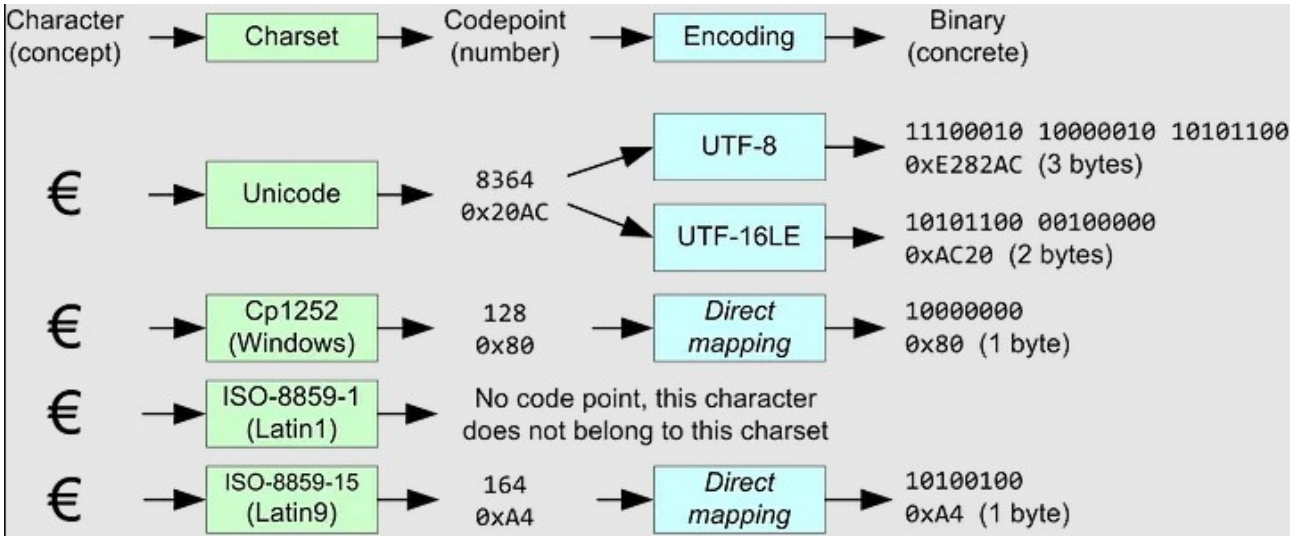
c. "Character Sets" vs. "Character Encodings"

İnternetteki ilk karakter seti olan ASCII karakter seti icat edildiğinde ve sonrasında charset ve character encoding arasında bir fark bulunmamaktaydı. İki kavram da aynı şeyi ifade ediyordu. Her ikisi de karakterlerin nasıl binary olarak tutulacağını ifade ediyordu. Fakat Unicode charset’i icad edildikten sonra arada bir fark oluştu. Unicode charset’inin icadı ile karakterlerin binary halde tutulması 2 adımdan oluşur hale geldi. Bu adımlar;

- Karakterler ve karşılık geldikleri teorik, soyut sayılar
- Karakterlerin karşılık geldikleri teorik, soyut sayıların binary olarak nasıl tutulacağı

şeklindendir.

Örnek vermek gerekirse € karakteri karakter setlerinde bir teorik, soyut sayı karşılığına (code point sayısına) sahiptir ve bu teorik, soyut sayı da farklı encoding yöntemleri ile farklı farklı binary'ler halinde somut olarak tutulmaktadır.



Yani € karakteri Unicode karakter setinde varmış ve UTF-8 ile encode'landığında farklı, UTF-16LE ile encode'landığında farklı binary'ler halinde tutulmaktaymış veya € karakteri ISO-8859-1 karakter setinde yer almamaktaymış ve bu nedenle bu karakter setinde herhangi bir teorik, soyut sayı karşılığına sahip değilmiş.

Sonuç olarak karakter setleri (charset'ler) karakterlerin bilgisayarda hangi teorik / soyut sayısal değerlerle depolanacağını belirler. Character Encoding'ler ise karakterlerin bilgisayarda hangi binary değerlerle depolanacağını belirler. Dolayısıyla ASCII bir karakter setidir ve karakterlerin hangi decimal değerlerle bilgisayarda depolanacağını tanımlar. UTF-8 ise bir character encoding'tir ve karakterlerin hangi binary değerlerle bilgisayarda depolanacağını tanımlar.

Character Sets ve Encodings Üzerine

Charset ve character encoding kavramları önceleri aynı, sonradan farklı anlamlar ifade eder olduklarından dolayı günümüzde internet dili html tasarlanırken tasarımda eski ifade kalmıştır ve günümüze kadar gelmiştir.

```
// HTML 4'de charset belirleme
<meta http-equiv="Content-Type" content="text/html; charset="utf-8">

// HTML5'de charset belirleme
<meta charset="UTF-8">

// Includekarabuk 'deki <meta etiketi
<meta http-equiv="Content-Type" content="text/html; charset="utf-8">
```

Bu html ifadelerdeki utf-8 bir charset değildir. Bir character encoding'tir. Fakat html sayfalarda character encoding'i belirtirken charset keyword'ü halen kullanılmaktadır.

Character Encoding ve Encryption Üzerine

Encoding, karakterlerin binary olarak 0 ve 1'ler ile nasıl depolanacağını belirler.

Encoding = characters -> binary

Encryption, karakterlerin binary olarak 0 ve 1'ler ile depolanması sonrası bu 0 ve 1'lerin nasıl başka 0 ve 1'lere dönüştürüleceğini belirler.

Encryption = understandable binary -> unintelligible binary

(anlaşılmaz binary)

Encoding ile oluşan kaynak 0 ve 1'ler başkalarınca anlaşılabilir. Dışarıdan bu 0 ve 1'lerin korunması ve anlaşılamaması amacıyla encryption kullanılır.

Not:

Encryption framework'ler bazen hem encoding hem encryption arka arkaya uygulayabilmektedirler. Bu ise bu iki kavramın birbirine karışmasına neden olabilmektedir. Fakat bu iki kavram ayrı iş yapmaktadırlar.

d. Character Reference

Karakter referansı Unicode karakter setlerindeki karakterleri referans yoluyla çağırmamızı sağlayan kodlamalardır. HTML'deki numerik "karakter referans"ları Unicode (Universal Character Set) 'daki karşılık gelen bir karakteri gösterirler. Numerik karakter referanslarının formatı şu şekildedir:

&#nnnn;

ya da

&#xhhhh;

nnnn olan format decimal form'dur. hhhh olan ise hexadecimal form'dur. nnnn ve hhhh herhangi bir sayı alabilir. Karakter referansı mevcut karakter setinde tanımlı karakterleri örneğin html dökümanına referans yoluyla dahil edebilmemizi sağlar. Mesela Türkçe klavye kullanan bir kimse klavyesinden matematiksel sembolleri normal şartlarda çıkaramaz. Çünkü karakter setinde tanımlı o matematiksel sembolleri klavyeden çıkarmak uzun ve karmaşık tuş kombinasyonları gerektirir. Bu komplike yöntem yerine ilgili karakterin karakter referansı kullanılabilir ve istenilen semboller böylece ekrana verilebilir. Aşağıda bir html dökümanına klavyeden girilebilmesi mümkün olan karakterlerin "karakter referansları" verilmiştir:

Unicode character	Character Reference (decimal)	Character Reference (hexadecimal)	Effect
U+0020	&#32;	&#x20;	(space)
U+0021	&#33;	&#x21;	!
U+0022	&#34;	&#x22;	"
U+0023	&#35;	&#x23;	#
U+0024	&#36;	&#x24;	$
U+0025	&#37;	&#x25;	%
U+0026	&#38;	&#x26;	&
U+0027	&#39;	&#x27;	'
U+0028	&#40;	&#x28;	(
U+0029	&#41;	&#x29;	)
U+002A	&#42;	&#x2A;	*
U+002B	&#43;	&#x2B;	+
U+002C	&#44;	&#x2C;	,
U+002D	&#45;	&#x2D;	-
U+002E	&#46;	&#x2E;	.
U+002F	&#47;	&#x2F;	/
U+0030	&#48;	&#x30;	0
U+0031	&#49;	&#x31;	1
U+0032	&#50;	&#x32;	2
U+0033	&#51;	&#x33;	3
U+0034	&#52;	&#x34;	4
U+0035	&#53;	&#x35;	5
U+0036	&#54;	&#x36;	6
U+0037	&#55;	&#x37;	7
U+0038	&#56;	&#x38;	8
U+0039	&#57;	&#x39;	9
U+003A	&#58;	&#x3A;	:
U+003B	&#59;	&#x3B;	;
U+003C	&#60;	&#x3C;	<
U+003D	&#61;	&#x3D;	=
U+003E	&#62;	&#x3E;	>
U+003F	&#63;	&#x3F;	?
U+0040	&#64;	&#x40;	@
U+0041	&#65;	&#x41;	A
U+0042	&#66;	&#x42;	B
U+0043	&#67;	&#x43;	C
U+0044	&#68;	&#x44;	D
U+0045	&#69;	&#x45;	E
U+0046	&#70;	&#x46;	F
U+0047	&#71;	&#x47;	G
U+0048	&#72;	&#x48;	H
U+0049	&#73;	&#x49;	I
U+004A	&#74;	&#x4A;	J
U+004B	&#75;	&#x4B;	K
U+004C	&#76;	&#x4C;	L

U+004D	&#77;	&#x4D;	M
U+004E	&#78;	&#x4E;	N
U+004F	&#79;	&#x4F;	O
U+0050	&#80;	&#x50;	P
U+0051	&#81;	&#x51;	Q
U+0052	&#82;	&#x52;	R
U+0053	&#83;	&#x53;	S
U+0054	&#84;	&#x54;	T
U+0055	&#85;	&#x55;	U
U+0056	&#86;	&#x56;	V
U+0057	&#87;	&#x57;	W
U+0058	&#88;	&#x58;	X
U+0059	&#89;	&#x59;	Y
U+005A	&#90;	&#x5A;	Z
U+005B	&#91;	&#x5B;	[
U+005C	&#92;	&#x5C;	\
U+005D	&#93;	&#x5D;	]
U+005E	&#94;	&#x5E;	^
U+005F	&#95;	&#x5F;	_
U+0060	&#96;	&#x60;	`
U+0061	&#97;	&#x61;	a
U+0062	&#98;	&#x62;	b
U+0063	&#99;	&#x63;	c
U+0064	&#100;	&#x64;	d
U+0065	&#101;	&#x65;	e
U+0066	&#102;	&#x66;	f
U+0067	&#103;	&#x67;	g
U+0068	&#104;	&#x68;	h
U+0069	&#105;	&#x69;	i
U+006A	&#106;	&#x6A;	j
U+006B	&#107;	&#x6B;	k
U+006C	&#108;	&#x6C;	l
U+006D	&#109;	&#x6D;	m
U+006E	&#110;	&#x6E;	n
U+006F	&#111;	&#x6F;	o
U+0070	&#112;	&#x70;	p
U+0071	&#113;	&#x71;	q
U+0072	&#114;	&#x72;	r
U+0073	&#115;	&#x73;	s
U+0074	&#116;	&#x74;	t
U+0075	&#117;	&#x75;	u
U+0076	&#118;	&#x76;	v
U+0077	&#119;	&#x77;	w
U+0078	&#120;	&#x78;	x
U+0079	&#121;	&#x79;	y
U+007A	&#122;	&#x7A;	z
U+007B	&#123;	&#x7B;	{
U+007C	&#124;	&#x7C;	|
U+007D	&#125;	&#x7D;	}
U+007E	&#126;	&#x7E;	~

Örneğin A karakteri ASCII karakter setinde 65 decimal sayısıyla ifade edilmekteydi. Bu karakter setindeki A karakterini referans yoluyla çağırmak için

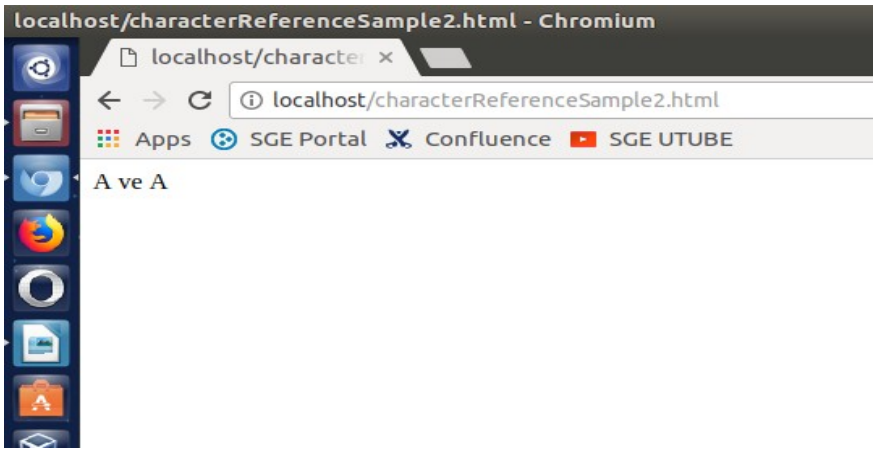
U+0041	&#65;	&#x41;	A

A decimal referansı ya da A hexadecimal referansı kullanılabilir.

/var/www/characterReferenceSample2.html

A ve A

Çıktı:



Böylece karakter referansları ile biz karakter setinde tanımlı karakterleri html dökümanına ekleyebiliriz.

Aşağıda ise karakter setinde yer alsa bile normal şartlarda klavyeden çıkarılamayacak bir matematiksel sembolün karakter referansı gösterilmiştir:

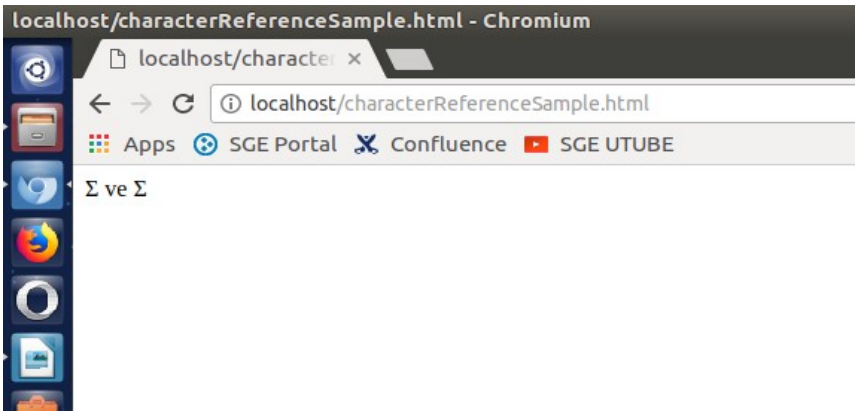
Unicode character	Character Reference (decimal)	Character Reference (hexadecimal)	Effect
U+03A3	&#931;	&#x3A3;	Σ

Karakter setindeki Σ sembolünü referans yoluyla çağırmak için Σ decimal referansı ya da Σ hexadecimal referansı kullanılabilir.

/var/www/characterReferenceSample.html

Σ ve Σ

Çıktı:



Character Reference'ı yanında bir de Character Entity Reference'ı vardır. Character Reference'ları decimal ya da hexadecimal sayıları kullanarak karakter setindeki bir karakteri göstermeye yararken "Character Entity Reference"ları ise isim kullanarak karakter setindeki bir karakteri göstermeye yarar. Örneğin HTML'de öntanımlı entity'ler (> , " , & , … v.b.) ilgili karakterleri gösterirken DTD'de öntanımlı entity'ler ve ayrıca explicitly olarak kendi tanımladığımız entity'ler ilgili karakterleri gösterir. Character entity referanslarının formatı şu şekildedir.

&name;

Kaynaklar

https://www.w3schools.com/html/html_charset.asp

https://www.w3schools.com/charsets/ref_html_utf8.asp

https://stackoverflow.com/questions/2281646/whats-the-difference-between-encoding-and-charset

https://tr.wikipedia.org/wiki/Unicode

https://medium.com/@joffrey.bion/charset-encoding-encryption-same-thing-6242c3f9da0c