



HTML encoding - charset



- SGML
 - Standard Generalized Markup Language
 - "Lenguaje de Marcado Generalizado"
 - Aplicaciones como HTML, XML, XHTML, etc
 - Requiere que toda aplicación informe el conjunto de caracteres para sus documentos
 - Además cada caracter tiene asociado su “code position”
 - Número entero que identifica la posición dentro del conjunto
 - Cada documento SGML es una secuencia de caracteres de ese conjunto
 - Sistemas computacionales identifican cada caracter de acuerdo a su posición
 - ASCII: 65: A, 66: B, 67: C



- El conjunto de caracteres ASCII no es suficiente
 - En HTML se utiliza un conjunto de miles de caracteres
 - Universal Character Set (UCS), definido en ISO10646
 - Define miles de caracteres, usados por distintas comunidades en el mundo
 - Se actualiza periódicamente con nuevos caracteres
 - El conjunto de caracteres de un documento no es suficiente para interpretarlo bien
 - Es necesario especificar el “character encoding” que se usó para transformar
 - Document character stream → byte stream



- “Character encoding” y “charset” son lo mismo
 - IANA mantiene una lista completa de encodings definidos
- El parámetro “charset” identifica una codificación de caracteres
 - Es el método de conversión de una secuencia de bytes a caracteres
 - Servidores WEB envían HTML a los browser
 - Como una secuencia de bytes
 - Browser interpreta como una secuencia de caracteres
 - El método de conversión puede ser un algoritmo simple (correspondencia 1 a 1) o un esquema complejo



- Seleccionando un encoding
 - Editores de texto generan documentos con un encoding predeterminado
 - Ese encoding depende exclusivamente del sistema operativo en uso
 - Servidores y proxies también pueden cambiar el encoding de los documentos
 - Encoding más usados
 - ISO-8859-1, también llamado Latin1
 - ISO-8859-5, con soporte a cirílico
 - SHIFT_JIS, EUC-JP, encoding Japonés
 - UTF-8, encoding de ISO10646, usa diferente número de bytes para diferentes caracteres



- ¿Cómo sabe un servidor que encoding usar para los documentos que sirve?
 - Algunos examinan los primeros bytes del documento
 - Podrían chequear contra un grupo de encoding conocidos
 - Servidores modernos le dan el control al “webmaster”
 - Se debe tener cuidado para no identificar un documento con el encoding equivocado
- ¿Cómo sabe el browser que encoding se usó para los datos?
 - El servidor web provee esta información
 - Parámetro “charset” del header HTTP Content-Type
Content-Type: text/html; charset=EUC-JP

HTML



- El protocolo HTTP menciona un encoding por “default”
 - Se usa cuando el servidor no especifica el encoding
 - Los browser podrían no asumir uno por “default”
- Documentos HTML incluyen información explícita sobre su encoding
 - Se utiliza el elemento META para informar a los browser del encoding del documento

```
<META http-equiv="Content-Type" content="text/html;  
charset=EUC-JP">
```



- Hay veces que ni el protocolo HTTP ni el HTML especifica un encoding
 - Elementos particulares del documento pueden especificar su encoding
 - Los browser pueden usar la siguiente prioridad para determinar el encoding a usar:
 - Parámetro “charset” en “Content-Type” de HTTP
 - Declaración META en HTML, con “http-equiv” y “charset”
 - Atributo charset asignado a un elemento que referencia a un recurso externo:

```
<A href="http://www.w3.org/" charset="ISO-8859-1">W3C  
Web site</A>
```




- Un browser también considera
 - Configuraciones del usuario
 - Mecanismos para sobre-escribir charset erróneos
- Referencias a caracteres
 - Un encoding podría no alcanzar a representar todo el charset de un documento
 - Para estos casos se utiliza referencias a caracteres
 - Es un mecanismo independiente de encoding de caracteres
 - Se utilizan de dos formas
 - Numéricas
 - Entidades



- Referencias a caracteres
 - Numéricas: Especifica un conjunto de enteros que hacen referencia al carácter, en el charset del documento
 - Ejemplos
 - `&#D;` donde D es un número decimal, se refiere al carácter número decimal “D” de ISO 10646
 - `&#xH;` donde H es un número hexadecimal, se refiere al carácter en la posición hexadecimal “H” de ISO 10646
 - `å` `å` representan al mismo carácter (letra å)



- Referencias a caracteres
 - Entidades: Una forma más intuitiva de hacer referencia a los caracteres
 - No abarcan todo el conjunto de caracteres
 - Se utilizan nombres simbólicos
 - No obligan a aprenderse los números que hacen referencia al carácter (code position)
 - Ejemplo
 - `å` representa a “å” (es más fácil de recordar que `å`)
 - Son case-sensitive
 - `Å` es la misma anterior pero en mayúscula
 - Otras famosas: `<` (<) `>` (>) `&` (&) `"` (“)



- Entidades
 - Desarrolladores deberían utilizar `<` en vez de `<`
 - Evitar posibles confusiones con el inicio de un TAG HTML
 - Lo mismo con `>` en vez de `>`
 - Se debería utilizar `&` para evitar la confusión con el inicio de una referencia a un carácter

Consideraciones



- En un sistema web, debe considerar
 - Encoding sistema operativo
 - Ver configuración de “locale”
 - Encoding de servidor web
 - Ver configuración de Apache
 - Encoding de base de datos
 - Ver opciones de creación de bases de datos
 - Encoding de CGI
 - Header HTML
- Todos los anteriores deberían tener un encoding acorde
 - Como convertir de un encoding a otro? → iconv