**Character Set and Encoding System**

Character Set and Encoding System are different concepts, but often confused together.

* A char set is just a standardized set of characters.
* An encoding system is a standardized way to transform a sequence of characters (of a given char set) into a sequence of 0 and 1.

In the early days of computing, these two concepts are not clearly made distinct, and are just called a char set or encoding system. For example, ASCII does not really separate the concepts, since it's very simple, dealing with only 128 chars (including invisible “control characters” (code sequence)). Another example: HTML has<meta http-equiv="Content-Type" content="text/html;charset=utf-8">; the syntax contains the word “charset”, but it's actually about encoding, not charset. 〔➤ [HTML: Character Sets and Encoding](http://xahlee.info/js/html_chars.html)〕

An encoding system defines a character set implicitly. Because it needs to define what characters it is designed to handle.

**What's Character Encoding?**

Any file has to go thru encoding/decoding in order to be properly stored as file or displayed on screen. Your computer needs a way to translate the character set of your language's writing system into a sequence of 1s and 0s. This transformation is called [Character encoding](http://en.wikipedia.org/wiki/Character_encoding).

There are many encoding systems. The most popular encoding systems used today are:

* ASCII. For English. Most widely used before year 2000.
* UTF-8 of Unicode (used in Linux by default, and much of the Internet)
* UTF-16 of Unicode (used by Microsoft Windows and Mac OS X's file systems, [Java programing language](http://xahlee.info/java-a-day/java.html), …)
* [GB 18030](http://en.wikipedia.org/wiki/GB_18030) (Used in China, contains all Unicode chars).
* [EUC](http://en.wikipedia.org/wiki/Extended_Unix_Code) (Extended Unix Code). Used in Japan.
* [IEC 8859](http://en.wikipedia.org/wiki/ISO/IEC_8859) series (used for most European langs)

## Most popular -- Unicode's Character Set and Encoding Systems

Unicode is a standard created by the Unicode Consortium.

Unicode primarily defines 2 things:

1. a character set. (which includes the characters needed for all world's languages.)
2. Several encoding systems. (most popular are UTF-8, UTF-16)

### Unicode's Character Set

Unicode's character set includes ALL human language's written symbols. It includes the tens of thousands Chinese characters, math symbols, as well as characters of dead languages, such as [Egyptian Hieroglyph](http://xahlee.info/comp/unicode_egyptian_hieroglyph.html). 〔➤ [Sample Characters of Unicode](http://xahlee.info/comp/unicode_common_symbols.html)〕

Unicode Search   
Type star or #97 or #x61 or paste ♥¥α©

### Unicode Character's Code Point

Each character in Unicode is given a unique ID. This id is a number (integer), and is called the char's **code point**.

For example, the code point for the greek alpha α char is 945. In hexadecimal it's “3b1”. In the standard Unicode notation it is written as “U+03B1”.

### Unicode's Encoding System: UTF-8, UTF-16, …

Then, Unicode defines several encoding system. UTF-8 and UTF-16 are the two most popular Unicode encoding systems. Each encoding system has advantages and disadvantages.

UTF-8 is suitable for texts that are mostly Latin alphabet letters. For example, English, Spanish, French, and most web technology such as [HTML](http://xahlee.info/js/index.html), [CSS](http://xahlee.info/js/css_index.html), [JavaScript](http://xahlee.info/js/javascript_basics.html). Most Linux's files are in UTF-8 by default. UTF-8 encoding system is backwards compatible with ASCII. (meaning: If a file only contain characters in ASCII, then encoding the file using UTF-8 results the same byte sequence as using ASCII as encoding scheme.)

UTF-16 is another coding system from Unicode. With UTF-16, every char is encoded into least 2 bytes, and commonly used characters in Unicode are exactly 2 bytes. For Asian languages containing lots of Chinese characters, such as Chinese ＆ Japanese, UTF-16 creates smaller file size.

There's also UTF-32, which always uses 4 bytes per character. It creates larger file size, but is simpler to parse. Currently, UTF-32 is not being used much.