Please see: Understanding Encoding as the info below is quite old.

Related: utf-8

Character Encoding settings allow various language character sets to be displayed properly in a browser.

For more on internationalization see: i18n

All the data in Tiki are supposed encoding with UTF-8 which can handle all characters defined by the Unicode standard while still being relatively compact. Problems can arise if you are looking at this site and you see some strange character when you choose a non english language.

One concern is that your browser has to support UTF-8 and further more figure out that the Tikiwiki pages are encoded in this encoding. If your browser interprets the UTF-8 encoded output as something else, for example Latin-1 which is the default in Western Europe, then the characters 'æ', 'ø', and 'å' in the Danish locale looks like 'à¦', 'à¸', and 'à¥'.

Redflo solved this in August 2004.
All releases since then are unaffected...

Background

The Unicode standard defines the Universal Character Set (UCS) which gives numbers to all the characters in all the alphabeths of the world. The UCS is a superset of Latin-1 (ISO-8859-1) which again is a superset of ASCII. ASCII defines the first 128 characters and Latin-1 defines another 128 characters and thereby exhusts all bits in a 8-bit byte.

The UCS defines many more characters, so 1 byte per character is not enough. Unicode uses 31 bit, so the logical size of each character would be 4 bytes (32 bit). The problem with those wide characters is that they're only needed if your use of the ~2 billion characters are evenly distributed --- most people use no more than 256 of those characters in their documents, so there's a lot wasted space.

The UTF-8 encoding is a way of transforming 4 byte wide characters into 1-6 byte wide characters. It's backwards compatible with ASCII meaning that texts encoded in ASCII automatically is in UTF-8 as well. Other encodings (including Latin-1) use two or more bytes to represent each character. That's why 'æ', 'ø', and 'å' turns into two-letter combinations when an UTF-8 encoded text is viewed as Latin-1.

All the above is dealt with in much more detail in the UTF-8 and Unicode FAQ for Unix/Linux which is usefull for a lot more than just Unix/Linux.

If Tiki on your server doesn't look fine:

I've added a line
header('Content-Type: text/html; charset=utf-8');
in tiki-setup.php. This should fix the "browser character encoding decision" problems.
redflo.

Some servers (like Apache with the default Debian config) adds a `charset=iso-8859-1` to the Content-Type header. The browser (ex: Mozilla) first looks for a charset value in the Content-Type header and then for META tags, so the header overrides the META tag inserted by TikiWiki.

To solve this with Apache then either check that there is no option `AddDefaultCharset iso-8859-1` in the `httpd.conf` file. Some distributions set `AddDefaultCharset on` in the `httpd.conf` for some security issues (see Apache Css Security), so keep in mind you can allways overset default settings in virtualhost directive in the `httpd.conf` file.

An other solution is to move the `doc/htaccess` in your Tikiwiki installation to `.htaccess` and uncomment `AddDefaultCharset utf-8` which will add the correct header for UTF-8 output.

Test on this page itself

(pl) Czy polski ogónki funkcjonujÄ… tutaj, pisany przez mozilli? Np, tu trochÄ™ treÅ›Ä‡. (ja) æ—¥æœ¬èªž. (en) Good, this looks OK .

Editing the translations

If you change the `language.php` file in Tikiwiki, keep in mind to save the file in UTF-8 encoding. If you're using Emacs then it's easy to change the encoding of a file. Simply open the file, and then type '-+C-x RET f+-' which runs the command `set-buffer-file-coding-system`. Now choose `utf-8` from the list.

original page