

Creating a Dzongkha monolingual and bilingual Website

A website can be monolingual or multi-lingual. A monolingual website is a website with content in one language and a multilingual website is any website that offers contents in more than one language.

For both monolingual and multilingual Dzongkha website, following are the some of the important technical features to be kept in mind.

1. **HTML Encoding (Character Sets)**
2. **Using Language Attribute**
3. **Styling the Dzongkha content using :lang() selector**
4. **Using Dzongkha fonts**
5. **Storing Dzongkha characters in MySQL database.**

Additionally, following aspects should be considered for multilingual websites.

6. **Domains and URL Structure**
7. **Switching language**

1. HTML Encoding (Character Sets)

To display an HTML page correctly, a web browser must know the character set (character encoding) to use. The charset attribute specifies the character encoding for the HTML document. The charset is specified in the <meta> tag.

Syntax: <meta charset="character_set">

Sample code 1: Specifying the character set.

```
<html>
<head>
<meta charset="UTF-8">
</head>
</html>
```

Character set (charset): A set of characters recognized by the computer. (ASCII, Unicode, etc.)

ASCII was the first character encoding standard (also called character set). It defines 127 different alphanumeric characters that could be used on the internet.

ANSI (Windows-1252) was the original Windows character set. It supported 256 different character codes.

ISO-8859-1 was the default character set for HTML 4. It also supported 256 different character codes.

Unicode covers almost all of the characters and symbols in the world.

Unicode: Unicode provides a unique number for every character in the world.

The default character encoding was changed to UTF-8 in HTML5.

UTF-8: Unicode Transformation Format. 8 means it uses 8 bit blocks to represent a character. It is a character encoding capable of encoding all possible characters, or code points, defined by Unicode.

UTF-8 encoding should be used for Dzongkha webpages.

Difference between Unicode and UTF-8

Unicode is a character set. UTF-8 is encoding.

Unicode is a list of characters with unique decimal numbers (code points). A = 41, B = 42,

Encoding is how these numbers are translated into binary numbers to be stored in a computer:

Encoding translates numbers into binary. **Character sets** translates characters to numbers.

2. Using Language Attribute

Language of the content in the web page can be programmatically determined. The lang attribute specifies the language of the element's content.

Always use a language attribute on the html element. This is inherited by all other elements.

The dir attribute specifies the text direction of the element's content. ltr (left to right) is the default value for dir attribute.

Syntax: <html lang="language_code">

Sample code 2: Specifying language of the HTML page .

```
<html lang="dz" dir="ltr">
<head>
<meta charset="UTF-8">
</head>
</html>
```

If you have any content on the page that is in a different language from that declared in the html element, use language attributes on elements surrounding that content. This allows you to style it differently.

Syntax: <element lang="language_code">

Sample code 3: Specifying language of the element's content .

```
<html lang="dz" dir="ltr">
<head>
<meta charset="UTF-8">
</head>
<body>
<h1>འདྲི་ཚོང་ཁའི་ཡོངས་འབྲེལ་འཆར་ཤོག་ཡིན།</h1>
<p>ཚོང་ཁའི་འབྲེལ་གྱི་རྒྱལ་ཡོངས་སྐད་ཡིག་ཡིན།<p>
<p lang="en">Dzongkha is the national language of Bhutan</p>
```

```
</body>
</html>
```

3. Styling the content using lang attribute.

Styles are commonly used to control changes in fonts, font sizes and line heights when language changes occur in the document. The best way to style content of different languages in HTML is to use the :lang() selector in your CSS style sheet. The :lang() selector is used to select elements with a lang attribute with the specified value.

Syntax: :lang (languagecode) {css declarations;}

Sample code 4: using :lang() selector for sample code 3
:lang(dz) { font-family:dzongkha; font-size:50px; color:#006}
p:lang(en){ font-family:Arial, Helvetica, sans-serif; font-size:24px; font-style:italic; color:#0F0}

4. Using Dzongkha fonts

For the proper display for Dzongkha, it is important that we use Dzongkha fonts. Not all the devices used to access internet may have the Dzongkha fonts installed. However, CSS3 Web Fonts allows Web designers to use fonts that are not installed on the user's computer.

You can just include the desired font file on your web server, and it will be automatically loaded to the user's computers when needed.

Your "own" fonts are defined within the CSS3 @font-face rule.

Syntax:

```
@font-face {
    font-family: name;
    src: url();
}
```

Sample code 5: defining our own font
@font-face
{
font-family: 'dzongkha';
src: url('fonts/DDC_Uchen.ttf') format('truetype');
}

Note: in above code, fonts are supposed to be in the folder named fonts in the same directory as the css file. Though, the TrueType Font file is supported by all the web browsers, Embedded Opentype Font file is still used for the older versions of IE which some users with Windows XP still uses it.

Dzongkha web fonts can be downloaded from
http://www.dzongkha.gov.bt/IT/download/Dzongkha_web_fonts.zip

5. Storing Dzongkha characters in MySQL database.

The Unicode characters should be stored using utf8 encoding and utf8_general_ci collation.

Let's create a database and a table to hold our data. What is most important here is the "CHARACTER SET utf8" portion. This tells MySQL that all the text in this table will be encoded in UTF-8.

Sample code 6: creating database with utf8 character set and utf8_general_ci collation

```
$sql = "CREATE DATABASE dzongkha_web DEFAULT CHARACTER SET utf8
      DEFAULT COLLATE utf8_general_ci";
if (mysqli_query($conn, $sql)) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . mysqli_error($conn);
}
mysqli_close($conn);
```

Tables created in the database will use utf8 and utf8_general_ci by default for any character columns.

Sample code 7: creating table with utf8 character set and utf8_general_ci collation

```
$sql = "CREATE TABLE dzongkha_text (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
dzongkha_name VARCHAR(30) NOT NULL
) CHARSET=utf8 COLLATE utf8_general_ci ";

if (mysqli_query($conn, $sql)) {
    echo "Table dzongkha_text created successfully";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}
```

Note that MySQL uses a non-standard name "utf8" to mean UTF-8. The COLLATE command is used to tell how to sort the data when using the SORT BY command. Also note that you should always use VARCHAR instead of CHAR with UTF-8. (UTF-8 uses variable sized numbers for different characters. For instance, Latin letters use 1 byte codes, while Japanese characters are 3 bytes. Using CHAR(10) would force the database to reserve 30 bytes, because it doesn't know ahead of time which length will be used, so it reserves the maximum.)

6. Domains and URL structure

There are three commonly used approaches to creating multilingual URL/domain structures:

1. Multidomain – ccTLDs (country-code top level domain names)
2. one subdomain per language (www.en.yoursite.com; www.dz.yoursite.com; www.es.youresite.com...)
3. one sub-folder per language, with only one domain name (www.yoursite.com/en/); www.yoursite.com/dz/))

Since most of the Bhutanese website won't target audiences from other countries or have content primarily or only relevant to people in one country, ccTLDs is not required.

According to Google:

If your time and resources are limited, consider buying one non-country-specific domain, which hosts all the different versions of your website. In this case, we recommend either of these two options:

- Put the content of every language in a different subdomain. For our example, you would have en.example.com, de.example.com, and es.example.com.
- Put the content of every language in a different subdirectory. This is easier to handle when updating and maintaining your site. For our example, you would have example.com/en/, example.com/de/, and example.com/es/.

The fourth option to use URL parameters, for example mywebsite.com?lang=dz is not recommended as they are difficult for search engines to interpret.

As the multilingual websites are basically translations of the same content (most of the time), subdirectories are an obvious solution. This approach is the easiest and cheapest to implement. Simply add a language directory to your server.

www.mywebsite.com/en
www.mywebsite.com/dz

7. Switching language and linking pages

A multilingual website is useless without the ability to change languages. Oftentimes you'll find multilingual websites with language switcher placed on top right of the page.

Some multilingual website redirect to the home page while user switches the language which is actually not desirable. When the user switches the language while on particular page, the user should be redirected to the equivalent page of the language chosen. For example, if the user is on www.mysite.com/en/intro.php, he/she should be redirected to www.mysite.com/dz/intro.php when he clicks to switch the language.

The simplest way to achieve this page to page linking is to provide link of the equivalent pages in other languages on the page itself. But this will be tedious and very static. One can make it less tedious and dynamic by writing simple function.

Sample code 8: Simple function to switch language.

PHP: langchange.php

```
function langchange()
{
    $current_page_uri = $_SERVER['REQUEST_URI'];
    if($poss=strpos($current_page_uri,'/dz/'))
    {

        $redirectto= substr_replace($current_page_uri,"/en/", $poss,4);
        echo $redirectto;

    }
    else
    {
        $poss=strpos($current_page_uri,'/en/');
        $redirectto= $redirectto= substr_replace($current_page_uri,"/dz/", $poss,4);
        echo $redirectto;
    }
}
```

HTML

```
<div id="lang_switch" lang="en"><a href="<?php include('./langchange.php'); langchange();?>">English</a></div>
```

In above php function, we are simply determining the current uri and looking for the substring '/en/' and '/dz/' and replacing with each other. It is assumed that the subdirectory for Dzongkha is named 'dz' and English is named 'en'.