

1 Técnicas y consejos para crear una web multi-idioma

1.1 Sumario

- 1 Introducción a la programación de webs multi-idioma
- 2 Códigos de idioma
- 3 Ajustar el idioma primario de la página web
- 4 Cómo indicar el idioma del contenido
- 5 Google y el reconocimiento de idioma
- 6 Dirección del idioma
- 7 Codificación de caracteres
- 8 Tamaños de letra
- 9 Longitud de las palabras
- 10 Implementación de Multi-Idioma con PHP

1.2 Introducción a la programación de webs multi-idioma

Si alguna vez has trabajado o creado algún sitio web en varios idiomas, hay que tener en cuenta un millón de cosas. Algunas cosas como la clasificación de los nombres de dominio, configuración del servidor web, la estructura de direcciones URL, el diseño de página y la traducción de los contenidos son susceptibles de agregar en la lista con cosas por hacer.

Además de todo esto tenemos que tener en cuenta la accesibilidad al sitio web, la cuál se puede conseguir de manera más o menos fácil.

Un sitio web multi-idioma es una web cuyo contenido está escrito en más de un idioma.

La información se muestra en diferentes idiomas y es exactamente la misma salvo pequeñas variaciones. Un ejemplo puede ser la web de <http://www.booking.com/> la cuál permite cambiar de idioma en el menú superior.

1.3 Códigos de idioma

Lo primero que se debe hacer cuando se trabaja con varios idiomas en un sitio web es asegurarse de que el lenguaje está identificado correctamente en el código de la página. Además se deben cumplir los criterios de accesibilidad como:

- El idioma por defecto de cada página web se puede determinar mediante programación.
- Las ayudas técnicas, tales como lectores de pantalla y dispositivos Braille no pueden identificar automáticamente el idioma que se utiliza en un sitio web desde el texto solamente.

El lenguaje debe ser identificado en el código de la página a fin de que estas tecnologías de asistencia puedan interpretarla correctamente. Una vez reconocido el idioma, estas tecnologías pueden cambiar automáticamente a ese idioma, ajustar el acento, el tono y la velocidad de habla de los contenidos en función de la lengua en cuestión.

Lectores de pantalla modernos tales como JAWS y Window Eyes son capaces de hablar múltiples idiomas con un acento apropiado.

1.4 Ajustar el idioma primario de la página web

Lo primero que se necesita es escoger el idioma principal de las páginas. Éste va a ser el de la mayoría del contenido escrito en la web. Por ejemplo, si en tu página lo que predomina es el idioma Inglés, entonces la lengua materna deberá ajustarse en Inglés.

Una vez que hemos elegido la lengua materna de las páginas, entonces tendremos que ajustar el idioma correspondiente a nuestro contenido. Los códigos de idioma consisten generalmente en dos letras, sin embargo puede utilizarse también cuatro letras para definir dialectos. Un lenguaje de código de dos letras 'en' podría ser utilizado para definir el 'Inglés', mientras que el código de idioma de cuatro letras "en-GB" podría utilizarse para distinguir Inglés Británico de Inglés Americano "en-US". Tenga en cuenta «en-UK» no es un código de idioma de cuatro letras válido.

A continuación tendremos que aplicar el código de idioma a nuestra página. Para configurar el idioma principal de la página en Inglés usamos el atributo 'lang' junto con nuestro código de lenguaje 'en' y aplicaremos esto al elemento HTML al comienzo de cada página.

```
<html lang="en">
```

Para otros idiomas tendríamos que resetear el idioma indicado, por ejemplo para Francés, español y alemán:

```
<html lang="fr">
```

```
<html lang="es">
```

```
<html lang="de">
```

Si especificamos un código incorrecto de idioma, puede ser que la aplicación encargada de leer el texto, no pueda hacerlo, o que pronuncie un texto en español con acento francés, por ejemplo.

1.5 Cómo indicar el idioma del contenido

Para indicar el idioma de los contenidos de la página se puede hacer de la siguiente forma:

```
<blockquote lang="fr">
<p>Le plus grand faible des hommes, c'est l'amour qu'ils ont de la vie.</p>
</blockquote>
```

También es necesario indicar el idioma de los enlaces, por ejemplo:

```
Aquí estamos enlazando a una página escrita en francés, desde una página escrita en Inglés.
<a href="http://www.xxxx.yyy" hreflang="fr">French</a>
```

Si necesitamos identificar tanto el texto del enlace, como el idioma de destino, tendremos que usar tanto 'lang' como 'hreflang':

```
<a href="http://www.xxxx.yyy" lang="fr" hreflang="fr">Francais</a>
```

El atributo 'hreflang' debería usarse solamente con enlaces.

1.6 Google y el reconocimiento de idioma

A diferencia de algunas tecnologías como los screen readers, Google no reconoce los atributos 'lang' en el código fuente de la página. Google trata de identificar el idioma de la página principal por sí mismo. Para facilitar el reconocimiento de idioma, Google recomienda usar solamente un idioma por página.

Información muy interesante para adaptar las páginas correctamente a Google en:

<http://googlewebmastercentral.blogspot.com.es/2010/03/working-with-multilingual-websites.html>

1.7 Dirección del idioma

Si vamos a crear un sitio web en varios idiomas, y que atiende a lenguas escritas de derecha a izquierda en lugar de izquierda a derecha, tendremos que asegurarnos de que la dirección del texto se especifica correctamente en el código de la página. Se puede configurar la dirección del texto con el atributo 'dir' en el elemento HTML. Para idiomas como el Árabe, Persa y Urdu el atributo 'dir' debe establecerse en RTL (de derecha a izquierda):

```
<html dir="rtl">
```

Un atributo 'dir' no es necesario para las páginas escritas utilizando lenguajes de izquierda a derecha, como Español, ya que es la dirección por defecto del texto.

Los diferentes diseños de página a menudo se requieren para idiomas de derecha a izquierda, como la mayoría de idiomas de derecha a izquierda deben alineados a la derecha en vez de izquierda alineado. Esto significa que será necesario adaptar el diseño de la página a estas lenguas, reflejando fundamentalmente la disposición de la izquierda para las páginas de idiomas adecuados. Por ejemplo, el sitio web de las Naciones Unidas adapta su diseño de la lengua árabe que se escribe de derecha a izquierda. Todo el diseño de la página se invierte si se compara con el idioma inglés. Estos cambios de diseño se pueden lograr utilizando CSS.

1.8 Codificación de caracteres

La codificación de caracteres se utiliza para decodificar la colección de letras y símbolos usados en el sistema. La codificación más utilizada es "Unicode". Unicode contiene caracteres para la mayoría de los idiomas y scripts en el mundo y está soportado por un número muy grande de sistemas operativos.

Ésto quiere decir que Unicode puede mostrar múltiples idiomas dentro de la misma página, lo cuál lo hace una opción excelente para programar este tipo de páginas.

```
Con HTML5 pondríamos dentro de la cabecera del documento HTML:  
<meta charset="UTF-8">
```

```
Con HTML4:  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

Para más información visitar: <http://www.w3.org/International/questions/qa-html-encoding-declarations>

1.9 Tamaños de letra

Cuando se diseña una web multi-idioma, es importante tener en cuenta el tamaño de letra a utilizar por defecto, ya que ésta en algunos casos puede que no sea aconsejable para todos los idiomas. Diferentes idiomas como Chino, Japonés o Árabe pueden ser difíciles de leer con un tamaño pequeño de letra.

Por lo tanto para algunos idiomas tendremos que incrementar el tamaño de letra en pantalla. Para ello hay dos formas de hacerlo:

- Por un lado podemos usar la pseudo clase CSS lang para poner diferentes tamaños de letra dependiendo del atributo lang:

```
<html lang="en"> o bien <html lang="zh">  
  
:lang(en) {  
    font-size: 85%;  
    font-family: arial, verdana, sans-serif;  
}  
  
:lang(zh) {  
    font-size: 125%;  
    font-family: helvetica, verdana, sans-serif;  
}
```

Esta técnica está soportada hoy en día en prácticamente todos los navegadores.

- Por otro lado podríamos hacerlo creando una clase css para cada idioma:

```
<body class="english"> o bien <body class="chinese">  
  
.english {  
    font-size: 85%;  
    font-family: arial, verdana, sans-serif;  
}  
  
.chinese {  
    font-size: 125%;  
    font-family: helvetica, verdana, sans-serif;  
}
```

1.10 Longitud de las palabras

La longitud de las palabras varía entre diferentes idiomas, por lo que deberemos ser cuidadosos con estos tamaños ya que por ejemplo los menús se pueden ver afectados por este tipo de cambios de idioma.

Por lo tanto tendremos que tener en cuenta nuestro diseño, para que pueda crecer automáticamente sin modificar la estructura de la página, y adaptándose a las diferentes resoluciones de los dispositivos.

1.11 Implementación de Multi-Idioma con PHP

Si queremos dar soporte multi-idioma a nuestra aplicación PHP una forma sencilla de hacerlo sería la siguiente.

- Podemos estructurar la web para que anteponiendo el código de idioma antes de la página, indique que idioma cargar:

Ejemplos:

- <http://www.veiga.local/amadeus/es/index.html>
- <http://www.veiga.local/amadeus/en/index.html>
- <http://www.veiga.local/amadeus/fr/index.html>

Mediante un fichero **.htaccess** podemos pasar ese parámetro del idioma a un fichero controlador encargado de cargar el idioma utilizado para la web.

- Las diferentes traducciones las podemos almacenar en ficheros del estilo:

es_traducccion.php, en_traducccion.php, fr_traducccion.php

O también dependiendo del tipo de contenido, se puede almacenar en una base de datos, que contenga las diferentes traducciones de los contenidos de la web. Por ejemplo para almacenar una noticia en diferentes idiomas, lo normal sería almacenarla en una base de datos que contenga todas las traducciones, mientras que para almacenar unas opciones de menú de una web que prácticamente no va a cambiar, lo más cómodo sería hacerlo en unos ficheros de texto como los mostrados a continuación.

- Dentro de cada fichero definiríamos las traducciones por ejemplo de la siguiente forma:

es_traducccion.php

```
<?php
// Fichero de idioma español.
define('LABEL_BIENVENIDA','Bienvenidos a la web de Amadeus.');
```

```
define('LABEL_MENU','Menú Principal');
define('LABEL_NOMBRE','Nombre');
define('LABEL_APELLIDOS','Apellidos');
.....
?>
```

en_traducccion.php

```
<?php
// Fichero de idioma inglés.
define('LABEL_BIENVENIDA','Welcome to Amadeus website.');
```

```
define('LABEL_MENU','Main Menu');
define('LABEL_NOMBRE','Name');
define('LABEL_APELLIDOS','Surname');
.....
?>
```

fr_traducccion.php

```
<?php
// Fichero de idioma francés.
define('LABEL_BIENVENIDA','Bienvenu à la web Amadeus.');
```

```
define('LABEL_MENU','Menu Principal');
define('LABEL_NOMBRE','Nom');
define('LABEL_APELLIDOS','Prénom');
.....
?>
```

- El fichero controlador (al que nos redireccionaría el .htaccess) se encargaría de cargar el fichero de idioma en base al idioma seleccionado. Un ejemplo de uso podría ser:

El fichero **.htaccess** podría tener la siguiente estructura:

.htaccess

```
RewriteEngine on
DirectoryIndex index.html
Options -Indexes
RewriteRule ^(.+)/(.+).html$ controlador.php?inc=$2&idioma=$1 [QSA,L]
RewriteRule ^(.+).html$ controlador.php?inc=$1&idioma=es [QSA,L]
```

controlador.php

```
<?php
// Primero averiguamos el idioma seleccionado.
if (isset($_GET['idioma']) && $_GET['idioma']!='')
    $idioma=strtolower($_GET['idioma']);
else
    $idioma='es';

// Cargamos el fichero de idioma. Por defecto español.
include_once "{$idioma}_traduccion.php";

//..... Resto del código.

// Ejemplos de uso.
echo LABEL_MENU;

echo LABEL_BIENVENIDA;
echo "Hola: ".LABEL_MENU;
?>
```

--Veiga (discusión) 20:22 13 may 2013 (CEST)