## Intro

The purpose of this ‘proof of concept’ web site is to demonstrate showing and hiding various language elements.

If the user selects Afrikaans, then hide all language elements and only show the Afrikaans labels.

In this example I used a javascript library called JQuery, to assist with element selection. It has awesome built in group functions, that get applied to all elements in a selected group.

It is also possible to use vanilla javascript, although you would need to do additional looping to iterate through an array of elements.

## Step 1: Identify all the text paragraphs and labels that require additional language tags

It might be best to record all of the labels and paragraphs in an excel sheet to keep track and register which ones have been completed with translations etc.

**Example text:** This is a paragraph (It could be surrounded by html tags like '<span>' or '<p>' etc)

## Step 2: Enclose the inner most text with 'span' tags

Note that the span html tag is an inline tag, meaning it does not affect the default structure of the web page.

We need to encase the text in a tag in order to programmatically manipulate them.

Also give them 2 relevant class labels. An element can belong to many classes, but may only have 1 ID value.

The first 'generic' class label is used to hide all text labels in our code.

The second 'language specific' class label is used to show a specific language.

**Example:**

<span class="lang-caption langtype-eng">This is a paragraph</span>

## Step 3: Create additional span tags for other languages

Give them an additional attribute called 'hidden'. The hidden attribute will initially hide the other language tags.

Note that each language label/paragraph needs a tag, as shown below.

Example:

<span class="lang-caption langtype-eng">This is a paragraph</span>

<span class="lang-caption langtype-afr" hidden>Hierdie is 'n paragraaf</span>

<span class="lang-caption langtype-ger" hidden>Das ist ein Absatz</span>

## Step 4: The javascript file will handle all the langauge changing functionality

The file language.js contains the relevant javascript functions.

**There are 3 main functions to be aware of:**

1) $( document ).ready -> used to trigger the initiallizing code only once the document has been loaded. It also registers a click even for the language dropdown menu.

2) loadLanguageFunction -> This function is only called once every time a page is loaded. It tries to read local storage and get the saved language option - if available

3) myLanguageFunction -> used to set and show the relevant langauge tags and hide all the other langauge tags. This is called for when a user clicks a language option AND when a page loads (loadLanguageFunction).

## Language change mechanism

-> triggered by page load OR selecting a new language (click event)

-> when changing the language, hide all language tags (select all elements with class "lang-caption" and hide them, hence the purpose of all language elements having the same generic class name)

-> select all specific language tags and show them (select all classes of type Afrikaans "langtype-afr " and show them, hence the importance of the elements all having a specific language class)

$('.lang-caption').hide(); //hide all languages

$('.langtype-eng').show(); //show specific language class

## Getting the locally saved language

The browser built in localStorage variables are unique per web domain and stored on local device. So Facebook use etc would not interfere with the saved values for the CyberTigers website.

The value is used in the if block to run the correct language block of code.

var savedLang = localStorage.getItem('lang');

//gets the saved language ID (represents langauge dropdown item)

## Storing the locally saved language

This command stores the selected language in local cache storage.

localStorage.setItem('lang', savedLangID);