

How to Enable Auto-Toggle for Multi-Language

Brief Overview

The Auto-Toggle method is the recommended way to implement [WalkMe Multi-Language](#) on your website or platform. This method involves WalkMe detecting a language variable on your website to determine which language WalkMe text should appear in.

The following steps take you through implementing Auto-Toggle.

For any additional questions, or if you do not control the underlying code of your site but would like the language of your WalkMe content to change with your site language, please reach out to your Customer Success Manager or WalkMe contact to discuss possible options.

□ Digital Adoption Institute

- View the [Technical Configuration](#) course in the DAI
- Don't have a DAI account yet? [Sign up here](#)

How It Works

1. Find Your Language Variable

On external websites, you should implement the JavaScript function **walkme_get_language** on your website, and this function will return the Language Custom Identifier(s) you specified in the [admin.walkme.com](#) during the initial language setup.

In order to implement the *walkme_get_language* function, you must first have a language variable at the window level that detects the end-user's language. Ask your IT team for the following:

- Language Variable Name
- Language Variable Outputs

Note

- The Custom Identifier can only contain alphanumeric characters (letters and numbers) and hyphens
- Please refer to the following article for more information: [Multi-Language](#)

2. Create Your Multi-Language Code

1. Copy the following code to a Text Editor:

- **Note:** The highlighted text should be replaced with your own values

```
<script type="text/javascript">
window.walkme_get_language = function (){
  if (window.langCode)
  {
    if (window.langCode == 'EN') {
      return '';
    }
    else if (window.langCode == 'FR') {
      return 'FR';
    }
    else {
      return ''
    }
  }
}
</script>
```

2. Change the language variable name (in the example above its **window.langCode**), to the language variable name your IT team created

- We will use **window.langvar** as our IT team's variable name throughout this example
- The result looks like this:

```
<script type="text/javascript">
window.walkme_get_language = function (){
  if (window.langvar)
```

3. Define your default language:

Where the example above says this:

```
if (window.langCode == 'EN') {
  return '';
}
```

change it to this:

```
if (window.langvar == 'EN') {
```

```
return '';
```

- Change the variable output for whatever language you have set as default (in this example **EN** is our default language)
- The return function should be left blank for the default language or be the Language Custom Identifier found in the insights.walkme.com **ACCOUNT** tab

4. Add alternate languages:

- Repeat the process above (for *EN* in this case) with each alternative language you want to translate to
- For our example, we'll add 'FR' as our alternate language - this would appear as follows:

```
else if (window.langvar == 'FR') {  
    return 'FR';  
}  
else {  
    return ''  
}
```

Your final result should look like this:

```
<script type="text/javascript">  
window.walkme_get_language = function () {  
    if (window.langvar)  
    {  
        if (window.langvar == 'EN') {  
            return '';  
        }  
        else if (window.langvar == 'FR') {  
            return 'FR';  
        }  
        else {  
            return ''  
        }  
    }  
}  
</script>
```

3. Add The Multi-Language Code to Your Website

1. Copy your completed code
2. Open the head tag for your site
3. Paste the Multi-Language code on every page

In order to function correctly, Multi-Language code must be placed in the head tag of every page on

which WalkMe will appear.

Technical Notes

- If a manual toggle has been added to the WalkMe Menu, the manual selection will override auto-toggle and it will not work
- **For regular, non-single page applications**
 - The language function should be declared prior to the WalkMe snippet on each page
 - Do not set `walkme_get_language()` on the login page
- **For Single Page Applications**
 - Use the `WalkMeAPI.changeLanguage()` on the main window