

Note: This is for [Tiki24](#) and under, for [Tiki27](#) and beyond, look at [PluginTour](#).

Plugin Tour

Introduced in [Tiki15](#)

Use this [wiki plugin](#) to build your product tours, page component describers and other sequential information displays with Bootstrap popovers in a quick and easy way, using [Bootstrap Tour](#).

Parameters

Quick and easy way to build your product tours with Driver.js

Introduced in Tiki 15. Required parameters are in **bold**.

[Go to the source code](#)

Preferences required: `wikiplugin_tour`

Parameters	Accepted Values	Description	Default	Since
(body of plugin)		Content of the step		
title	text	Title of the step		15.0
element	text	Element to show the popup on; if empty, use the plugin location itself		15.0
next_button_text	text	Text to show on the Next button. (Set only in the first step.)	Next	27.0
number_of_steps	int	Number of steps in the tour. (Set only in the first step.)		27.0
overlay_color	text	Set an overlay color to be shown behind the popover and its element, highlighting the current step. (Set only in the first step.)	transparent	27.0
overlay_opacity	int	Set the opacity of the overlay. (Set only in the first step.)	0.5	27.0
prev_button_text	text	Text to show on the Prev button. (Set only in the first step.)	Previous	27.0

<code>progress_text</code>	text	Template for the progress text. You can use the following placeholders in the template: // - : The current step number. // - : Total number of steps. Example: Step of . (Set only in the first step.)	of	27.0
<code>show_restart_button</code>	text	Display a button to restart the tour. Enter the text to appear on the button. (Set only in the first step.)		15.0
<code>tour_id</code>	text	Set a tour ID to be able to only show the tour once. (Set only in the first step.)	default	15.0
<code>align</code>	end start center	The alignment of the popup. (Set only in the first step.)	end	15.0
<code>animate</code>	y n	Animate the popup. (Set only in the first step.)	y	27.0
<code>show_once</code>	(blank) y n	Show automatically only once. <code>tour_id</code> should also be set if there are multiple tours. (Set only in the first step.)	n	15.0
<code>show_progress</code>	y n	Show the progress text in popover. (Set only in the first step.)	n	27.0
<code>side</code>	right top bottom left	The side of the popup. (Set only in the first step.)	right	15.0
<code>start</code>	y n	Start the tour on page load? If "No", then a start button can be made with "Restart Button", below. (Set only in the first step.)	y	15.0
<code>allow_close</code>	y n	Allow closing the popover by clicking on the backdrop. (Set only in the first step.)	y	16.0

Examples

The first step:

```
{TOUR(title="Welcome to the tour" prev="-1" element="#PluginTour")}
```

Here's the start of a tour.

```
{TOUR}
```

A middle step:

```
{TOUR(title="A typical middle step")}
```

This step is neither first nor last on the page, isn't anchored to an element, and isn't an orphan. It's nothing special.

```
{TOUR}
```

A step that specifies a page element:

```
{TOUR(element="#module_1 .panel-heading" title="This is a module heading.")}
```

This is a module.

```
{TOUR}
```

A step that indicates that the following step is on another page:

```
{TOUR(path="PluginTour-page-2")}
```

This plugin has a title and body content, but these won't be displayed - this step is simply a redirect to the target page.

```
{TOUR}
```

The last step:

```
{TOUR(next="-1")}
```

This is the last step of the tour.

```
{TOUR}
```

Usage notes

- For each step of the tour, input the Tour plugin syntax as shown above.
- The first and last steps need a `prev="-1"` and `next="-1"` parameter, respectively. Without these, the tour won't display correctly.
- If the tour isn't configured to start on page load (if `start="n"` is set), a start button can be made with the `show_restart_button` parameter.
- Each step will pop over the page element where the plugin syntax is input in the wiki page, etc.
- The position of the popup relative to the element it pops up over can be set with the *placement* parameter.
- To create a step that is positioned other than where the plugin syntax is input, use the `element` parameter to specify an HTML id or other unique selector.
- Use the `path` parameter to indicate that the next step is on another page. Simply using a wiki page name is OK.
- A tour plugin step that uses the `path` parameter doesn't itself display (it is not visible); it simply acts as a redirect, so it shouldn't be used to contain information. Any guidance to the user that the tour is continuing on another page should be inserted in the *previous* plugin step (as of May 2017).

- At this time, the "prev" button on the first step of the tour's second page is ghosted and doesn't work to return to the first page.

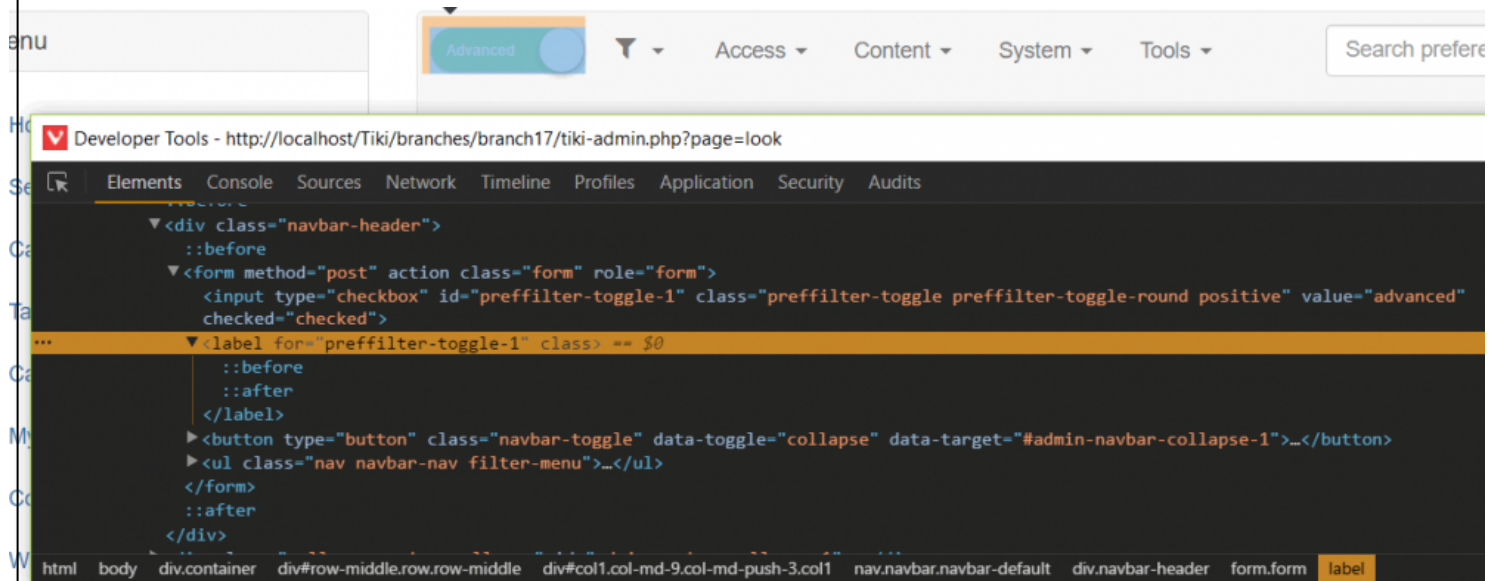
Adding a tour to a page that doesn't have wiki text content

Normally, as described above, the tour plugins are written into the wiki text of the page(s) on which the tour displays. However, it is possible to have a tour on an admin page or other such page that doesn't have wiki text.

This method uses a [module](#) for the tour plugin text that is assigned to, for example, the pagebottom module zone. Since all the steps of the tour are in the admin page content (and not in the module zone), the *element* parameter is used for each step that needs to be located in the page. But not all page elements in an admin page have unique DOM selectors, or ones that are easy to identify, so the ability to have a tour with complete tour-step placement may be limited for some pages.

Example: Look & Feel Admin page tour

The Look & Feel admin page is checked with a DOM inspector to identify the selectors where tour steps are to be placed.

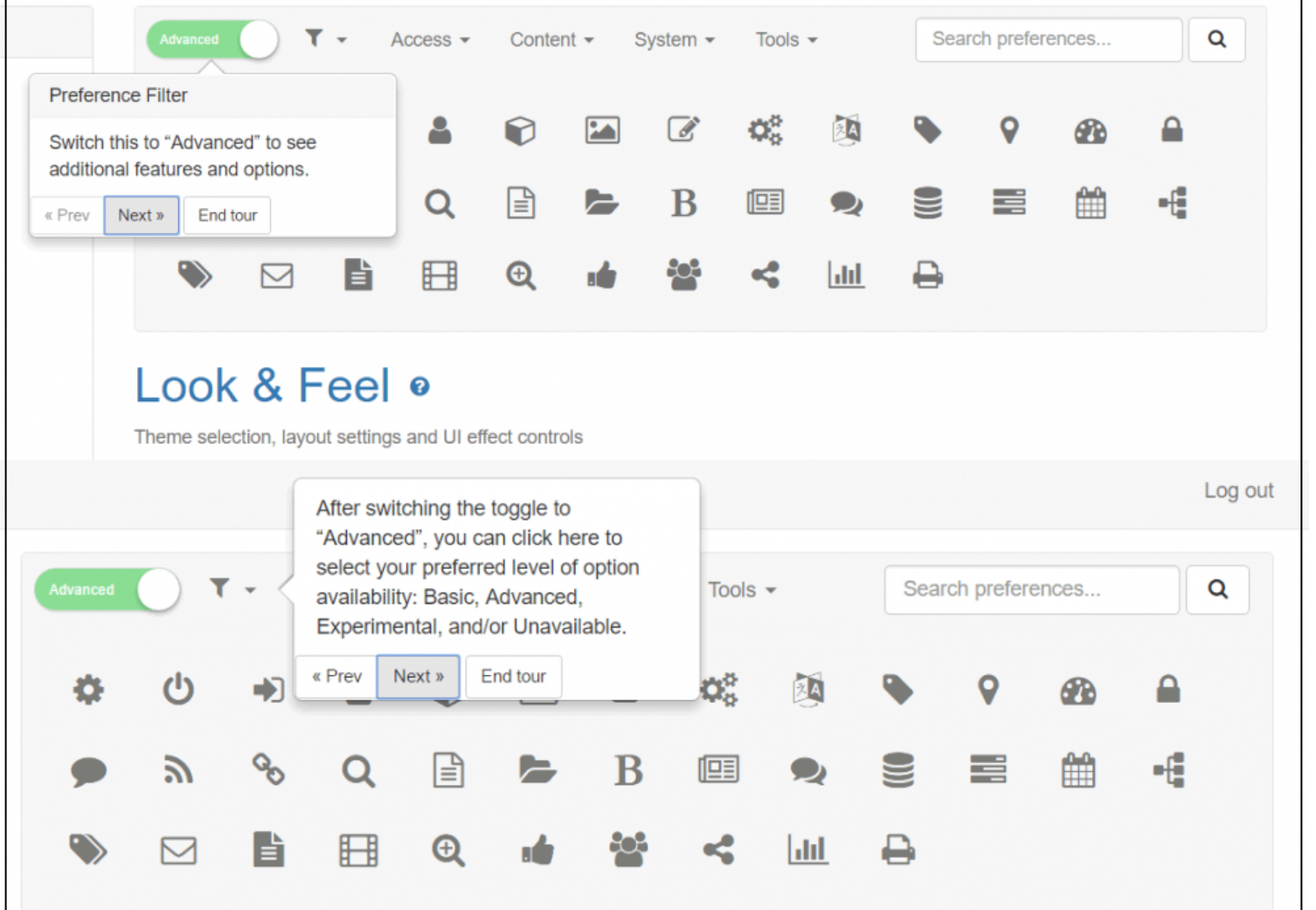


The tour's steps are specified using `PluginTour` in a new wiki page, and a new [menupage module](#) is created to place that wiki page's content in the pagebottom module zone. (Any module zone could be used but some may already be crowded with content. Also, while the `menupage` module was made to create menus, it can be used to display any wiki page content, so is good for this purpose.) For example, a page named `Admin_look_tour_test` was made that has this content:

```
{TOUR(title="Preference Filter" element="#preffilter-toggle-1 + label" placement="bottom"
prev="-1")}<Switch this to "Advanced" to see additional features and options.</TOUR>
{TOUR(element=".filter-menu" placement="right")}<After switching the toggle to "Advanced",
you can click here to select your preferred level of option availability: Basic, Advanced,
```

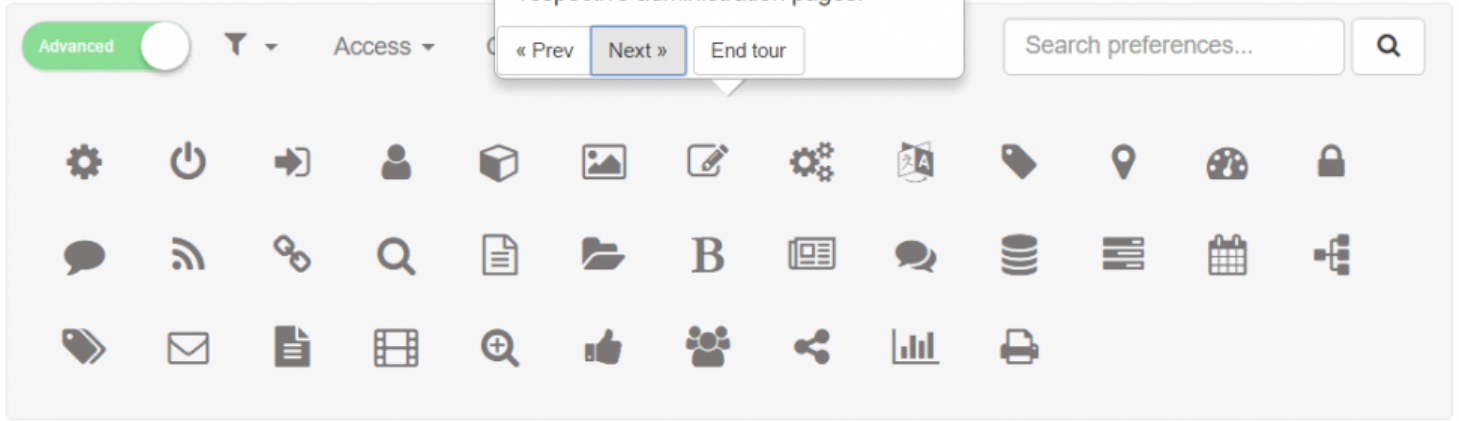
Experimental, and/or Unavailable. {TOUR} {TOUR(element=".adminanchors" placement="top")} These are the icons of features of this web site. Click them to navigate to their respective administration pages. {TOUR} {TOUR(element="form#look" placement="top" title="Save your changes")} Change the settings and click the "Apply" button to save any desired changes. Or to cancel changes, just navigate away from the page without applying the changes. {TOUR} {TOUR(element="form#look" placement="bottom" title="End of tour" next="-1")} {TOUR}

This produces these tour steps:



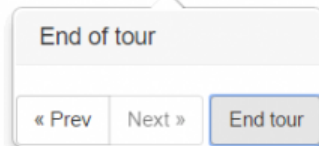
These are the icons of features of this web site. Click them to navigate to their respective administration pages.

Log out



Theme Control

Apply



Keep in mind that the tour code specified in the module will be available to all pages at the site. But the tour steps only activate when the *element* set in each step is found in the page. For this tour to show up only on `tiki-admin.php?page=look`, be sure the *elements* specified are only found on that page. Otherwise you may have random tour steps popping up on other pages.