Introduction

Pop-up dialog windows are quite common, compared to the "old" way (when the browser presents you the dialog box) the jQuery dialog boxes look nice, they can have their own styles and additional functionalities like resize, drag&drop, etc.

jQuery offers the dialog in its API to help you complete such task easily.

Tiki uses the external jQuery libraries to "write less and do more". jQuery is really useful and can spare you a lot of time once you learn how to use it.

---

adding a basic dialog

For practice let's see how to add a new link and have a simple dialog box popping up upon clicking the link

Based on the sample at Nemikor Blog

```javascript
//link added to the tpl
<a href="" id="opener">{tr}Basic dialog{/tr}</a>

//in the jq section at the end of the tpl first initialize the dialog
var $dialog = $('<div></div>
    .html('This is a basic Tiki dialog!')
    .dialog({
            autoOpen: false,
            title: 'Tiki Basic Dialog'
    });

//than the script to be executed upon clicking
$('#opener').click(function() {
    $dialog.dialog('open');
    // prevent the default action, e.g., following a link
    return false;
    });
```

This is the end result
serviceDialog

Usually dialog boxes are not for showing some plain text, but to present some functionality to perform a task. The serviceDialog function is a modified jQuery dialog function. Using this you can call Tiki services through controllers, such as:
- translation
- category
- file
etc

Controllers are listed at `/tiki-ajax_services.php`

serviceDialog helps to avoid code duplication, enables you to keep functionality logic in one place and to be able to call it from anywhere using parameters

Sample case - translation maintenance

Tiki 8.x: Above the wiki pages you have the language selector/translation maintenance select.

When you make your selection, either the page language changes, or you create a new translation, etc. This task is performed using `/templates/translated-lang.tpl`.

Now how about changing the selection list to a nice pop-up and add a link to the pop-up to open the maintenance dialog?

First is to create a new .tpl called translaged-lang2.tpl and add it to the library. This is only to be able to compare old and the new way side by side. When development is done, it will removed.
Using css menu feature and adding some lines to layout.css the "new" layout is created. (= the "old" option way is replaced by links). Now it is sufficient to move your mouse over the language name and your options appear nicely.

The sample case concerns the below parts of the code:

**Calling the maintenance dialog**

**OLD**

```html
<option value="_manage_">{tr}Manage translations{/tr}</option>
```

**NEW**

```html
<a href="#" onclick="attach_detach_translation('wiki page', '{$page|escape:"quotes"}')">{tr}Manage translations{/tr}</a>
```

In the old way the pop-up dialog windows was triggered when you made your selection in the dropdown.

In the new way it is triggered when you perform a click on the link called "Manage translation".

The code section in the .tpl related to what should happen when you perform your action:

```javascript
//OLD
...
else if( option.value == '_manage_' ) {
  $(document).serviceDialog({
    title: $(option).text(),
    data: {
      controller: 'translation',
      action: 'manage',
      type: object_type,
      source: object_to_translate
    }
  });

  return;

  ...
```
//NEW - you call the below defined function from the <a> element created above
function attach_detach_translation( object_type, object_to_translate ) {
    $(document).serviceDialog({
        title: '{tr}Manage translations{/tr}',
        data: {
            controller: 'translation',
            action: 'manage',
            type: object_type,
            source: object_to_translate
        }
    });
    return;
}

That is it!