

TabView

Version 1.0

Description:

This PHP component class is used to generate navigable tabs.

Usage:

1. Require the "tab_view.php" file in the PHP script you wish to write the tabs.
e.g. `require("path/to/tree_view.php")`
2. Instantiate the TabView class.
e.g. `$TabView = new TabView;`
3. Set the following TabView class properties (public variables):
 - a. `$TabView->BackColor = (String)`
Default: #EEEEFF
Sets the background color for tabs that are not selected.
 - b. `$TabView->BodyBackground = (String)`
Default: #FFFFFF
Sets the background color for the TabView area to match the background color of the page.
 - c. `$TabView->Class = (String)`
The name of the style sheet class name that defines the styles for the tabs.
 - d. `$TabView->ImagePath = (String)`
The relative path to the TabView images.
 - e. `$TabView->Orientation = (Integer)`
Default: 0
0 = left aligns the tabs
1 = right aligns the tabs
 - f. `$TabView->QueryString = (String)`
The query string to append to each of the tab URLs.
 - g. `$TabView->SelectedBackColor = (String)`
Default: #CCCCFF
Sets the background color for the tab that has been selected.
 - h. `$TabView->SelectedBold = (Boolean)`
Default: false
When set to true, bolds the text of the selected tab.
 - i. `$TabView->SelectedForeColor = (String)`
Sets the font color for the tab that has been selected.
 - j. `$TabView->StartTab = (String)`

Sets the tab that is selected when the page is first loaded by using the tab's key value.

- k. `$TabView->Image = (String)`
Sets the image for all unselected tabs. This setting can be overridden by each tab's *Image* property.
- l. `$TabView->SelectedImage = (String)`
Sets the image for the selected tab. This setting is overridden by each tab's *SelectedImage* property.
- m. `$TabView->DHTML = (String)`
Adds user defined DHTML to each tab's URL. That is, you can add properties to the anchor tag `<a>` of each tab such as the *onClick* event.
- n. `$TabView->ForceDHTML = (String)`
Omits the *href* property of the anchor tag and uses only the DHTML property value for each tab's URL.

4. Add tabs using the *Add()* method of the *TabView* class.
`$Tab =& $TabView->Add($strKey, $strText, $strURL, $strTarget, $strCaption);`

Where the given variables are defined as the following:

- strKey* Unique key identifying a given tab.
- strText* Text to display on the tab.
- strURL* Web address to fetch when the tab is clicked.
- strTarget* Target frame or window name when fetching the address.
- strCaption* Text Tool Tip to display when the mouse hovers over the tab's text.

5. The *Add()* method returns a pointer to the new *Tab* object that was created and added to *TabView*. You will notice in the last step that the *\$Tab* variable captured the pointer to the new *Tab* object returned by the *Add()* method. The *Tab* class has several additional properties that can be set after the tab has been created:

- a. `$Tab->ForeColor = (String)`
Sets the font color for the given tab.
- b. `$Tab->Bold = (Boolean)`
Controls whether or not the tab's text is bold.
- c. `$Tab->Image = (String)`
Sets the image for the tab when it is not selected. This setting overrides the *TabView* class *Image* property.
- d. `$Tab->SelectedImage = (String)`

Sets the image for the tab when it is selected. This setting overrides the `TabView` class `SelectedImage` property.

- e. `$Tab->DHTML = (String)`
Adds user defined DHTML to the given tab's URL. That is, you can add properties to the anchor tag `<a>` of this tab such as the `onClick` event.
- f. `$Tab->ForceDHTML = (String)`
Omits the `href` property of the anchor tag and uses only the DHTML property value for the given tab's URL.
- g. `$Tab->TabWidth = (Integer)`
Defines the width of the given tab measured in pixels.

6. To display `TabView` simply call the `Show()` method:
e.g. `$TabView->Show();`