

MediaWiki:Common.js

```
/* Any JavaScript here will be loaded for all users on every page load. */

if ( /^MediaWiki(V.+)?$/i.test( wgPageName ) && wgAction == 'view' ) { addOnloadHook(function(){
document.body.className+=" mainpage"; }) }

/* Force preview for anons */ function forcePreview() { if( wgUserName != null || wgAction != 'edit' ) return;
saveButton = document.getElementById('wpSave'); if( !saveButton ) return; saveButton.disabled = true;
saveButton.value = 'Save page (use preview first)'; saveButton.style.fontWeight = 'normal';
document.getElementById('wpPreview').style.fontWeight = 'bold'; } addOnloadHook(forcePreview); /* End of
forcePreview */

/** includePage *****

* force the loading of another JavaScript file
* @todo Is this still needed? We have importScript and importScriptURI in wikibits.js
* Maintainer: Commons:User:Dschwen
*/

function includePage( name ){ document.write('<script type="text/javascript" src="' + wgScript + '?title=' + name
+ '&action=raw&ctype=text/javascript"></script>'); } /* End of includePage */

/* Including extra .js pages */

// switches for scripts // TODO: migrate to JSConfig // var load_extratabs = true; var load_edittools = true;

// extra drop down menu on editing for adding special characters includePage( 'MediaWiki:Edittools.js' );

// Editpage scripts if( wgAction == 'edit' || wgAction == 'submit' ) importScript('MediaWiki:Editpage.js')

/* End of extra pages */

/* Test if an element has a certain class *****

*
*
* Description: Uses regular expressions and caching for better performance.
* Maintainers: User:Mike Dillon, User:R. Koot, User:SG
*/

var hasClass = (function() { var reCache = {}; return function ( element, className ) { return
(reCache[className] ? reCache[className] : (reCache[className] = new RegExp("(?:\\s|^)" + className +
"(?:\\s|$)"))).test(element.className); }; })();

/** Collapsible tables *****

*
*
* Maintainers: User:R. Koot
*/

var autoCollapse = 2; var collapseCaption = 'hide'; var expandCaption = 'show';
```

MediaWiki:Common.js

```
function collapseTable( tableIndex ){ var Button = document.getElementById( 'collapseButton' + tableIndex );
var Table = document.getElementById( 'collapsibleTable' + tableIndex );

if ( !Table || !Button ) { return false; }

var Rows = Table.rows;

if ( Button.firstChild.data == collapseCaption ) { for ( var i = 1; i < Rows.length; i++ ) { Rows[i].style.display =
'none'; } Button.firstChild.data = expandCaption; } else { for ( var i = 1; i < Rows.length; i++ ) {
Rows[i].style.display = Rows[0].style.display; } Button.firstChild.data = collapseCaption; } }

function createCollapseButtons(){ var tableIndex = 0; var NavigationBoxes = new Object(); var Tables =
document.getElementsByTagName( 'table' );

for ( var i = 0; i < Tables.length; i++ ) { if ( hasClass( Tables[i], 'collapsible' ) ) { /* only add button and
increment count if there is a header row to work with */ var HeaderRow = Tables[i].getElementsByTagName( 'tr'
)[0]; if( !HeaderRow ) continue; var Header = HeaderRow.getElementsByTagName( 'th' )[0]; if( !Header )
continue;

NavigationBoxes[tableIndex] = Tables[i]; Tables[i].setAttribute( 'id', 'collapsibleTable' + tableIndex );

var Button = document.createElement( 'span' ); var ButtonLink = document.createElement( 'a' ); var ButtonText =
document.createTextNode( collapseCaption );

Button.style.styleFloat = 'right'; Button.style.cssFloat = 'right'; Button.style.fontWeight = 'normal';
Button.style.textAlign = 'right'; Button.style.width = '6em';

ButtonLink.style.color = Header.style.color; ButtonLink.setAttribute( 'id', 'collapseButton' + tableIndex );
ButtonLink.setAttribute( 'href', "javascript:collapseTable(" + tableIndex + ");" ); ButtonLink.appendChild(
ButtonText );

Button.appendChild( document.createTextNode( "[" ) ); Button.appendChild( ButtonLink ); Button.appendChild(
document.createTextNode( "]" ) );

Header.insertBefore( Button, Header.childNodes[0] ); tableIndex++; }

}

for ( var i = 0; i < tableIndex; i++ ) { if ( hasClass( NavigationBoxes[i], 'collapsed' ) || ( tableIndex >=
autoCollapse && hasClass( NavigationBoxes[i], 'autocollapse' ) ) ) { collapseTable( i ); } } }

addOnloadHook( createCollapseButtons );

/** Dynamic Navigation Bars (experimental) *****/

*
* Maintainers: UNMAINTAINED
*/
```

MediaWiki:Common.js

```
// set up the words in your language var NavigationBarHide = '[' + collapseCaption + ']'; var NavigationBarShow
= '[' + expandCaption + ']';

// shows and hides content and picture (if available) of navigation bars // Parameters: // indexNavigationBar: the
index of navigation bar to be toggled function toggleNavigationBar( indexNavigationBar ){ var NavToggle =
document.getElementById('NavToggle' + indexNavigationBar); var NavFrame =
document.getElementById('NavFrame' + indexNavigationBar);

if( !NavFrame || !NavToggle ) { return false; }

// if shown now if( NavToggle.firstChild.data == NavigationBarHide ) { for( var NavChild =
NavFrame.firstChild; NavChild != null; NavChild = NavChild.nextSibling ) { if ( hasClass( NavChild, 'NavPic' )
) { NavChild.style.display = 'none'; } if ( hasClass( NavChild, 'NavContent' ) ) { NavChild.style.display = 'none'; }
} NavToggle.firstChild.data = NavigationBarShow;

// if hidden now } else if( NavToggle.firstChild.data == NavigationBarShow ) { for( var NavChild =
NavFrame.firstChild; NavChild != null; NavChild = NavChild.nextSibling ) { if( hasClass( NavChild, 'NavPic' ) )
{ NavChild.style.display = 'block'; } if( hasClass( NavChild, 'NavContent' ) ) { NavChild.style.display = 'block'; }
} NavToggle.firstChild.data = NavigationBarHide; } }

// adds show/hide-button to navigation bars function createNavigationBarToggleButton(){ var
indexNavigationBar = 0; // iterate over all < div >-elements var divs = document.getElementsByTagName('div');
for( var i = 0; NavFrame = divs[i]; i++ ) { // if found a navigation bar if( hasClass( NavFrame, 'NavFrame' ) ) {
indexNavigationBar++; var NavToggle = document.createElement("a"); NavToggle.className = 'NavToggle';
NavToggle.setAttribute('id', 'NavToggle' + indexNavigationBar); NavToggle.setAttribute('href',
'javascript:toggleNavigationBar(' + indexNavigationBar + ');');

var NavToggleText = document.createTextNode(NavigationBarHide); for( var NavChild = NavFrame.firstChild;
NavChild != null; NavChild = NavChild.nextSibling ) { if ( hasClass( NavChild, 'NavPic' ) || hasClass( NavChild,
'NavContent' ) ) { if( NavChild.style.display == 'none' ) { NavToggleText =
document.createTextNode(NavigationBarShow); break; } } }

NavToggle.appendChild(NavToggleText); // Find the NavHead and attach the toggle link (Must be this
complicated because Moz's firstChild handling is borked) for( var j=0; j < NavFrame.childNodes.length; j++) { if(
hasClass( NavFrame.childNodes[j], 'NavHead' ) ) { NavFrame.childNodes[j].appendChild(NavToggle); } }
NavFrame.setAttribute('id', 'NavFrame' + indexNavigationBar); } } } addOnloadHook(
createNavigationBarToggleButton );
```