

# ATutor Photo Gallery & Fluid Lightbox

ATutor is a PHP based Learning Content Management System being developed and maintained at the Adaptive Technology Resource Centre. The Photo Gallery module for ATutor was originally developed at the Garth Homer Society, in partnership with CanAssist and The University of Victoria. The module was developed as a tool that would be used with students who have developmental disabilities, providing them with an interactive way to share photos as a learning activity.



[My Start Page](#) > [Travels in Greece](#) > [Photo Album](#)

[Show](#)



## Photo Album

### Thumbnail view



[Go to Top](#) ▲

The Fluid lightBox was adapted for the module to make it possible to move and reorder images in the photo gallery. The implementation was relatively straight forward, and can be summarized in a few steps:

## Lightbox Implementation in ATutor

1. Include the Fluid libraries as a subdirectory within the modules code. For now the libraries can exist as part of the module, but eventually we'll probably end up including them as part of the ATutor trunk source code, and make them available as part of the ATutor API.
2. Modify the photo gallery template that displays thumbnails. The original table layout was replaced with a single block cut from the Lightbox.html example included with Fluid. In the following case tokens are set for

TABINDEX: an integer incremented each time the block is rendered by the template engine (Smarty)

LINK: the URL to the original image

IMAGE\_ALT: the Alt text for the current image

IMAGE\_SRC: the URL to the thumbnail image

IMAGE\_TITLE: the caption defined for the image

IMAGE\_ID: The unique ID number for the image, retrieve from the gallery database

```

<div class="float orderable-default" id="gallery:::gallery-thumbs:::lightbox-cell:{TABINDEX}:"
    role="wairole:gridcell"
    aaa:selected="true"
    aaa:readonly="false"
    aaa:disabled="false"
    aaa:grab="supported"
    aaa:dropeffect="move">
    <div>
        <div class="image-inner-container">
            <a href="{LINK}" >
                
            </a>
        </div>
        <div class="caption image-title">
            <a href="{LINK}" >{IMAGE_TITLE}</a>
        </div>

        <input name="{IMAGE_ID}" id="gallery:::gallery-thumbs:::lightbox-cell:{TABINDEX}:reorder-index"
value="{TABINDEX}" type="hidden"/>
    </div>
</div>

```

3. Add the script to initialize the lightbox into the index photo gallery page, into which the smarty templates are rendered, after the block in Step 2 that calls the template to display the thumbnails.

```

<script type="text/javascript">
    fluid.initLightbox ("gallery:::gallery-thumbs:::", "message-bundle:");
</script>

```

4. Add calls to the Fluid libraries into the photo gallery index page. In our case we've added this block into the body of the page that gets rendered when the gallery templates are render.

```

<?php $FLUID_URL = 'mods/photo_album/fluid/component-templates'; ?>
    <script type="text/javascript" src="<?php echo $FLUID_URL; ?>/js/jquery/jquery-1.2.1.js"></script>
    <script type="text/javascript" src="<?php echo $FLUID_URL; ?>/js/jquery.tabindex/jquery.tabindex.js"><
/s>script>
    <script type="text/javascript" src="<?php echo $FLUID_URL; ?>/js/jquery.ui-1.0/ui.mouse.js"></script>
    <script type="text/javascript" src="<?php echo $FLUID_URL; ?>/js/jquery.ui-1.0/ui.draggable.js"></script>
    <script type="text/javascript" src="<?php echo $FLUID_URL; ?>/js/jquery.ui-1.0/ui.droppable.js"></script>
    <script type="text/javascript" src="<?php echo $FLUID_URL; ?>/js/fluid/Fluid.js"></script>
    <script type="text/javascript" src="<?php echo $FLUID_URL; ?>/js/fluid/Reorderer.js"></script>
    <script type="text/javascript" src="<?php echo $FLUID_URL; ?>/js/fluid/Lightbox.js"></script>

```

5. Add an "order" column to the photo gallery database table that keeps track of which images belong to which courses, and to which individual users. Initially the order field is empty, to keep compatibility with older version of the gallery. When order is not defined, the gallery displays images descending by date submitted. Order is defined the first time the user clicks "Save Image Order," a button added to the thumbnail template to implement Lightbox reordering.

6. Add a Javascript to keep track of the image order in the DOM, and replace the POST order with the DOM order when the form is submitted. This might be something to add to the Fluid libraries.

```

<script language="javascript" type="text/javascript">
  /**
   * Capture the DOM subtree ordering.
   *
   * @param      the name of the form, that contains the thumbnails
   */
  function reordering_pa(form_name){
    //Quit if form_name isn't specified
    if (form_name==""){
      return;
    }

    //get form name
    var myForm = document.forms[form_name];

    //keep track of DOM order
    var count = 1;

    //loop through the new DOM tree and updates its associated values
    for (var i=0; i < myForm.elements.length; i++){
      // reassign values by DOM elements
      // TODO, maps the ID instead?
      var inputs = myForm.elements[i];
      if (inputs.id.indexOf('gallery::gallery-thumbs::lightbox-cell:') > -1){
        inputs.value = count;
        count++;
      }
    }
  }
</script>

```

7. The Form markup is modified to include the appropriate id:

```

<form name="{SAVE_FORM_NAME}" method="post" action="{SAVE_ACTION}" id="reorder-form">

```

SAVE\_FORM\_NAME: An arbitrary name replaced when smarty renders the page

SAVE\_ACTION: The URL to the script that inserts the new image order into the database, In our case this value is \$\_SESSION['fluid:PHP\_SELF'], the form pointing back to itself.

8. The Save Image Order button references the javascript with:

```

<input type="submit" name="save" value="{SAVE_STRING}" onclick="reordering_pa('save_form');" />

```

SAVE\_STRING is the text to display on the button, retrieved from the ATutor language database.

In our case changes to image order are made by pressing the Save Image Order button in order to minimize network traffic and hardware load whenever an image is moved, accommodating installations on older hardware and slower networks. To revert back to the Fluid default of submitting the new order each time an image is moved, the lines at the top of the gallery's index.php file can be uncommented, so each move (submit event) send the data to the server, rather than requiring a button press.

## Demo ATutor Photo Gallery

A demo of the lightbox implementation of the ATutor Photo Gallery can be found on the [ATutor Demo Site](#). Login with login name "demo" and password "demo", then click on Photo Album.

## Source Code

The entire source code for the ATutor Photo Gallery can be downloaded from the Modules page on ATutor.ca

[ATutor Modules](#)