

1. uDraw Designer UI	2
1.1 Plugin Creation Guide	2
1.1.1 Create Your Plugin	2
1.1.2 Designer Files	2
1.1.2.1 CSS	3
1.1.2.2 JS	3
1.1.2.3 PHP	4
1.1.2.3.1 Data Attribute Tags	4
1.1.2.3.2 Localization Text	45

uDraw Designer UI

All things related to the uDraw Designer and its UI.

If you want to learn to create your own Designer UI for our wordpress plugin, start with [Plugin Creation Guide](#).

If you have a problem, noticed a bug, or would like to make a suggestion, please [submit a ticket](#).

Plugin Creation Guide

You may decide to create your own uDraw Designer UI as the ones we provide do not fit your needs. You will need to create your own wordpress plugin. It will need to include some CSS, JS, and php files. We've put together some instructions on how to make your own UI. We will update and add more content as we improve the uDraw Designer.

[Next Page >](#)

Create Your Plugin

We suggest including a function to check for the main uDraw plugin prior to initializing your plugin.



Filters

Include the following filters in your plugin's initialization.

`udraw_designer_register_skin` – function assigned to this filter shall take one parameter. This function will push your UI into the array of UIs the main uDraw plugin may use.

`udraw_designer_ui_override` – function assigned to this filter shall take in 8 parameters. This function will register the files required for your UI to function.



Functions

The function used for registering your skin requires one parameter, and will return the same parameter with your UI name appended to the array.



The function used for including your UI files requires eight parameters, and will return either true or false depending on the current UI selected in uDraw > Settings > Designer UI. We suggest not changing the parameter names in the function or in your UI, but if you do, please check that they match. In the image below, you will see that some files are declared, or "required". These files are the UI specific JS, CSS, and PHP files. You will need to create and include these.



[< Previous Page](#)

[Next Page >](#)

Designer Files

You will be required to create your own [CSS](#), [JS](#), and [PHP](#) files for your UI. You may include any images you would like to use in your plugin folder.

We currently use Font Awesome for icons, and Select2 for select dropdowns. If you wish to use different libraries, you must include them in your plugin folder, register it in your main PHP file, and initialize it in your javascript file.

CSS

You are required to include your own CSS file. You may name it whatever you like, as long as it is registered via wordpress styles register and enqueue hooks.

Please include the following in your file:



The following are not required, but we recommend that you do include it:



We suggest adding the tag `#udraw-bootstrap` at the beginning of each rule to prevent wordpress themes from overwriting your rules.

JS

You are required to include your own javascript file. You may name it however you like, as long as it is registered via wordpress script register and enqueue hooks. This file will hold your custom code and events. An example would be binding a function to a button's click event.

We recommend to include the following in your file:



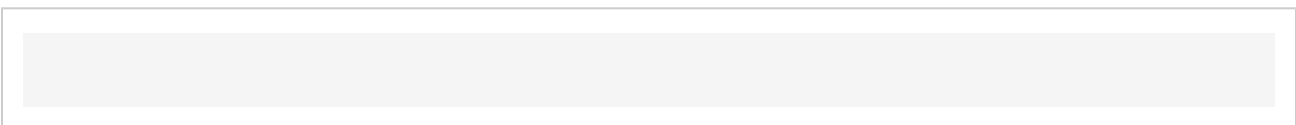
This block of code will trigger when mobile browser is detected. You may decide that the UI is not as visually appealing on a mobile screen as it is on a bigger screen.

Events, namespaces, functions... etc

For full documentation on canvas objects and events, please refer to [fabric.js](#). We use the namespace 'RacadDesigner' to hold our functions and events.

To access...	Use...
Main canvas	RacadDesigner.canvas
Select object	RacadDesigner.canvas.getActiveObject()
Select group of objects	RacadDesigner.canvas.getActiveGroup()
Current zoom	RacadDesigner.zoom

Example: To trigger a function when selecting an object



We have also bound some events to `#udraw-bootstrap`.

PHP

The HTML of your UI will be held here. Please include the following, as it is required for the Designer to function:

You may choose to use a carousel to display your pages. To do so, please add the following line in the document.ready portion.

```
var carousel_override = true;
```

By default, this setting is set to false, but you may override it.

Data Attribute Tags

We use data attribute tags to prevent errors caused by class or id duplication from other elements, such as theme objects. The tags will be listed in the following format:

attribute_name
<ul style="list-style-type: none">Description and details <div><html></html></div>

Usage example:

```
<a href="#" data-udraw="attribute_name"></a>
```

We have separated the tags into three general categories:

- [Required Tags](#)
- [Recommended Tags](#)
- [Optional Tags](#)

Please refer to [Localization Text](#) for displaying translatable text.

Required Tags

These tags must be included (but does not have to be visible) in order for the custom UI to be functional.

Canvas

- Main canvas; Where all the designing will occur
- Must use the <canvas> element

```
<canvas id="racad-designer-canvas" width="504" height="288" data-udraw="canvas"></canvas>
```

canvasWrapper

- Holds canvas
- Is required to trigger events

```
<div id="racad-designer" data-udraw="canvasWrapper">  
  <canvas id="racad-designer-canvas" width="504" height="288" data-udraw="canvas"></canvas>  
</div>
```

progressDialog

- Shows progress bar when loading / saving design

```
<div id="racad-designer-loading" data-udraw="progressDialog">  
  <div class="alert alert-info">  
    <strong><span data-il8n="[html]common_label.progress"></span></strong>  
    <div class="progress progress-striped active">  
      <div class="progress-bar" role="progressbar" aria-valuenow="105" aria-valuemin="0" aria-valuemax="100" style="width: 100%></div>  
    </div>  
  </div>  
</div>
```

canvasContainer

- Contains canvasWrapper, loading dialog

```
<div id="canvas-container" data-udraw="canvasContainer">
  <div id="racad-designer-loading" data-udraw="progressDialog">
    <!--progress dialog content-->
  </div>
  <div id="racad-designer" data-udraw="canvasWrapper">
    <canvas id="racad-designer-canvas" width="504" height="288"
data-udraw="canvas"></canvas>
  </div>
</div>
```

designerColourPicker

- Colour selector for canvas objects
- ⚠ type="button" does not seem to work on Safari (tested on Safari v.5.1.7)

```
<input type="text" value="#000000" data-opacity="1"
class="standard-js-colour-picker text-colour-picker"
style="background-color: rgb(255, 255, 255);"
data-udraw="designerColourPicker">
```

fontFamilySelector

- Selection of fonts - your uploaded fonts will be displayed here.
- Must use the <select> element

```
<select class="font-family-selection"
name="font-family-selection" data-udraw="fontFamilySelector">
  <option value="Arial"
style="font-family:'Arial';">Arial</option>
  <option value="Calibri"
style="font-family:'Calibri';">Calibri</option>
  <option value="Times New Roman" style="font-family:'Times
New Roman'">Times New Roman</option>
  <option value="Comic Sans MS" style="font-family:'Comic Sans
MS';">Comic Sans MS</option>
  <option value="French Script MT" style="font-family:'French
Script MT';">French Script MT</option>
</select>
```

fontSizeSelector

- Selection of font sizes - list of font sizes will be generated automatically
- Must use the <select> element

```
<select class="dropdownList font-size-select-option" data-udraw="fontSizeSelector"></select>
```

layersContainer

- Lists objects on canvas
- Is used for rendering objects on canvas in correct

```
<ul class="layer-box" id="layersContainer" data-udraw="layersContainer"></ul>
```

Recommended Tags

These tags are recommended, but not necessary. However, if using a button that is surrounded by a php if-statement, please include the php code, as it is required for the Designer to function as intended.

Note: UDRAW_DESIGNER_IMG_PATH refers to the directory of which images from the main uDraw plugin are held. This directory is where we hold the images we use in our UIs. You may use these images if you wish.

Each section will be separated into its own page, as there are many tags to cover.

- [General](#)
- [Text](#)
- [Images](#)
- [Shapes](#)
- [Toolboxes](#)
 - [Image Toolboxes](#)
 - [Text Toolbox](#)
 - [Misc Toolboxes](#)

General

Quick jump to sections:

- [General Buttons](#)
- [Zoom](#)
- [Pages](#)
- [Object Stroke](#)
- [Object Alignment](#)

General Buttons

undoButton

- Undoes the user's last action

```
<button data-udraw="undoButton">
  <i class="fa fa-undo fa-2x"></i>
  <span data-il8n="[html]button_label.undo"></span>
</button>
```

redoButton

- Redoes the undone action

```
<button data-udraw="redoButton">
  <i class="fa fa-repeat fa-2x"></i>
  <span data-il8n="[html]button_label.redo"></span>
</button>
```

removeButton

- Deletes the selected object(s)

```
<a href="#" data-udraw="removeButton">
  <div class="innerAnchorDiv">
    <i class="fa fa-trash"></i>
    <span data-il8n="[html]common_label.remove"></span>
  </div>
</a>
```

gradientButton

- Opens the gradient window if gradient can be applied to selected object
- Available only when the option is enabled in uDraw > Settings (please leave it wrapped in the php if-statement for proper functionality)

```
<?php if (!$_udraw_settings['designer_disable_text_gradient']) {
?>
  <div data-udraw="gradientButton"></div>
<?php } ?>
```

shadowButton

- Shows the [shadowModal](#)

```
<a href="#" id="add-shadow-btn" data-udraw="shadowButton">
  <i class="fa fa-tags"></i>&nbsp;<span
data-il8n="[html]menu_label.shadow"></span>
</a>
```

backgroundColour

- Opens the background colour picker

```
<button data-udraw="backgroundColour">
  <i class="fa fa-pencil-square fa-2x"></i>
  <span data-il8n="[html]menu_label.background"></span>
</button>
```

toggleLayers

- Toggles the layers toolbox
- Requires [layersModal](#)

```
<a href="#" data-udraw="toggleLayers"></a>
```

toolboxHide, toolboxClose

- Shrinks / closes the parent toolbox
- The parent toolbox must have the class "toolbox-modal" to use these buttons

```
<div data-udraw="layersModal" class="toolbox-modal">
  <a href="#" data-udraw="toolboxHide"><i class="fa
fa-chevron-up"></i><span
data-il8n="[html]common_label.hide"></span></a>
  <a href="#" data-udraw="toolboxClose"><i class="fa
fa-close"></i><span
data-il8n="[html]common_label.close"></span></a>
</div>
```

[Back to top](#)

Zoom

zoomLevel

- Slider to adjust the zoom level of the canvas

```
<div data-udraw="zoomLevel"></div>
```

zoomDisplay

- Displays the current zoom level

```
<span data-il8n="[html]text.zoom"  
data-udraw="zoomDisplay"></span>
```

If you prefer buttons to zoom in and out instead of sliders, create your buttons and add the following into your JS file (replacing "your_zoom_increase_tag" and "your_zoom_decrease_tag" with your values):

```
jQuery('[data-udraw="your_zoom_increase_tag"]').click(function () {  
  RacadDesigner.changeZoom(0.1);  
  jQuery('[data-udraw="zoomDisplay"]').text(Math.round(RacadDesigner  
    .zoom.currentZoom * 100) + "%");  
});  
jQuery('[data-udraw="your_zoom_decrease_tag"]').click(function () {  
  RacadDesigner.changeZoom(-0.1);  
  jQuery('[data-udraw="zoomDisplay"]').text(Math.round(RacadDesigner  
    .zoom.currentZoom * 100) + "%");  
});
```

The zoom change is set to 10% (0.1) but you may change it if you wish.

[Back to top](#)

Pages

togglePages

- Shows or hides the pages panel
- Requires `pagesContainer` and `pagesList`

```
<a href="#" data-udraw="togglePages"><span  
data-i18n="[html]common_label.pages"></span></a>
```

- Function shows and hides `pagesList` by default, but you may change it to toggle the `pagesModal` instead by adding the following code to your JS file

```
RacadDesigner.togglePages = function () {  
  if ($('[data-udraw="pagesModal"]').is(':visible')) {  
    $('[data-udraw="pagesModal"]').modal('hide');  
  } else {  
    $('[data-udraw="pagesModal"]').modal('show');  
    highlightModal(RacadDesigner.modal.pages);  
  }  
}
```

pagesContainer

- Contains pages related elements
- Contains `pagesList`

```
<div data-udraw="pagesContainer"></div>
```

pagesList

- Lists the pages in the current template
- You may set `RacadDesigner.settings.useCarousel = true`; for a nicer interface (see fullscreen UI)

```
<div data-udraw="pagesList"></div>
```

[Back to top](#)

Object Stroke

objectStrokeSpinner

- Sets the stroke on the selected object - default colour is black
- Must have the classes listed in the example below
- Must use <input type="text">

```
<input data-udraw="objectStrokeSpinner" type="text" value="0" data-opacity="1" class="stroke-spinner spinedit noSelect form-control" />
```

objectStrokeColour

- Sets the colour of the stroke on the selected object. If no stroke width was set, it will default to 1
- Must use <input> element
- Must have the class 'stroke-colour-picker'

```
<input data-udraw="objectStrokeColour" type="color" value="" data-opacity="1" class="stroke-colour-picker" />
```

[Back to top](#)

Object Alignment

objectsAlignLeft, objectsAlignCenter, objectsAlignRight, objectsAlignTop, objectsAlignMiddle, objectsAlignBottom

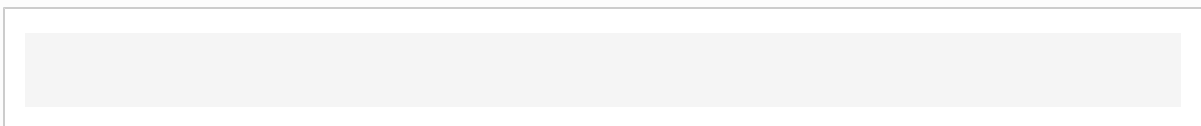
- Aligns selected object to described direction

```
<a href="#" data-udraw="objectsAlignLeft">
  <div class="innerAnchorDiv">
    
  </div>
</a>
```

[Back to top](#)

Images

We do not have a predefined attribute for converting SVG images into path-groups. You may bind the following event to a data attribute value of your own (please remember to check that the value is not already in use).



- imageColouringModal is required

replacelImage

- Replaces the selected image with another image of the user's choice
- Available only when the option is enabled in uDraw > Settings (please leave it wrapped in the php if-statement for proper functionality)

```
<?php if (!$udraw_settings['designer_disable_image_replace']) {  
?>  
<a href="#" data-udraw="replacelImage">  
  <div class="innerAnchorDiv">  
    <i class="fa fa-retweet"></i>  
    <span data-il8n="[html]common_label.replace"></span>  
  </div>  
</a>  
<?php } ?>
```

cropButton

- Opens image cropping window
- Requires [cropModal](#) and its contents

```
<a href="#" data-udraw="cropButton">  
  <div class="innerAnchorDiv">  
    <i class="fa fa-crop"></i>  
    <span data-il8n="[html]button_label.crop"></span>  
  </div>  
</a>
```

clipImage

- Shows image clipping tool
- Requires [imageClippingModal](#) and its contents

```
<a href="#" data-udraw="clipImage">  
  <div class="innerAnchorDiv">  
    <i class="fa fa-cut"></i>  
    <span data-il8n="[html]button_label.clip-image"></span>  
  </div>  
</a>
```

toggleImageFilters

- Shows image filters toolbox
- Requires `imageFilterModal` and its contents

```
<a href="#" data-udraw="toggleImageFilters"><span  
data-i18n="[html]menu_label.image-properties"></span></a>
```

uploadImage

- Required for uploading images

```
<input type="file" name="files[]" multiple  
data-udraw="uploadImage" />
```

imagePlaceholder

- Adds an image placeholder to the canvas

```
<a href="#" data-udraw="imagePlaceholder">  
  <span data-i18n="[html]menu_label.image-placeholder"></span>  
</a>
```

userUploadedImages

- Opens container holding previously uploaded images
- Requires `userUploadedModal` and its contents

```
<a href="#" data-udraw="userUploadedImages"></a>
```

clipartCollection

- Opens container containing the clipart collection
- Requires `clipartModal` and its contents

```
<a href="#" data-udraw="clipartCollection">  
  <span data-i18n="[html]common_label.clipart-collection"></span>  
</a>
```

privateClipartCollection

- Opens the private clipart collection container
- Will display only if the setting is enabled in uDraw > Settings
- Requires [privateClipartModal](#) and its contents

```
<?php if ($_udraw_settings['designer_enable_local_clipart']) {  
?>  
<a href="#" id="private-clipart-collection-nav-btn"  
data-udraw="privateClipartCollection">  
  <span  
data-il8n="[html]menu_label.private-clipart-collection"></span>  
</a>  
<?php } ?>
```

facebookPhotos

- Shows the photos uploaded to your Facebook account
- Available only when the option is enabled in uDraw > Settings (please leave it wrapped in the php if-statement for proper functionality)
- Requires [facebookModal](#) and its contents

```
<?php if ($_udraw_settings['designer_enable_facebook_photos'])  
{ ?>  
<a href="#" id="facebook-image-btn" data-udraw="facebookPhotos">  
  <span data-il8n="[html]menu_label.facebook-uploads"></span>  
</a>  
<?php } ?>
```

instagramPhotos

- Shows the photos uploaded to your Instagram account
- Available only when the option is enabled in uDraw > Settings (please leave it wrapped in the php if-statement for proper functionality)
- Requires [instagramModal](#) and its contents

```
<?php if ($_udraw_settings['designer_enable_instagram_photos'])  
{ ?>  
<a href="#" id="instagram-image-btn"  
data-udraw="instagramPhotos">  
  <span data-il8n="[html]menu_label.instagram-uploads"></span>  
</a>  
<?php } ?>
```

qrCode

- Shows the QR code toolbox
- Requires `qrModal` and its contents

```
<a href="#" data-udraw="qrCode">
  <span data-il8n="[html]common_label.QRcode"></span>
</a>
```

Shapes

shapesGroup

- Container to hold the list of shape buttons - can be a list

```
<?php if (!$udraw_settings['designer_disable_shapes']) { ?>
<li data-udraw="shapesGroup">
  <!--Shapes content-->
</li>
<?php } ?>
```

addCircle

- Adds a circle object to the canvas

```
<a href="#" data-udraw="addCircle">
  
  &nbsp;<span data-il8n="[html]menu_label.circle-shape"></span>
</a>
```

addRectangle

- Adds a rectangle object to the canvas

```
<a href="#" data-udraw="addRectangle">
  
  &nbsp;<span data-il8n="[html]menu_label.rect-shape"></span>
</a>
```

addTriangle

- Adds a triangle object to the canvas

```
<a href="#" data-udraw="addTriangle">
  
  &nbsp;<span data-il8n="[html]menu_label.triangle-shape"></span>
</a>
```

addLine

- Adds a line object to the canvas

```
<a href="#" data-udraw="addLine">
  
  &nbsp;<span data-il8n="[html]menu_label.line-shape"></span>
</a>
```

addCurvedLine

- Adds a curved line object to the canvas

```
<a href="#" data-udraw="addCurvedLine">
  
  &nbsp;<span
data-il8n="[html]menu_label.curved-line-shape"></span>
</a>
```

addPolygon

- Opens the polygon toolbox, where the user will be able to decide the number of sides the polygon will have prior to adding it to the canvas
- Requires [polygonModal](#)

```
<a href="#" data-udraw="addPolygon">
  
  &nbsp;<span
data-il8n="[html]menu_label.polyshape-shape"></span>
</a>
```

addStar

- Adds a star object to the canvas

```
<a href="#" data-udraw="addStar">
  
  &nbsp;<span data-il8n="[html]menu_label.star-shape"></span>
</a>
```

Text

- Requires [textModal](#) and its contents

addText

- Adds a text object to the canvas

```
<a href="#" data-udraw="addText">
  <span data-il8n="[html]common_label.text"></span>
</a>
```

addCurvedText

- Adds a curved text object to the canvas

```
<a href="#" data-udraw="addCurvedText">
  <span data-il8n="[html]menu_label.curvetext"></span>
</a>
```

addTextbox

- Adds a textbox object to the canvas

```
<a href="#" data-udraw="addTextbox">
  <span data-il8n="[html]menu_label.textbox"></span>
</a>
```

Toolboxes

(Also known as modals)

We contain a number of tools in containers called modals, which will show or hide according to the object (or group of objects) selected. We have broken them down into categories for easy navigation.

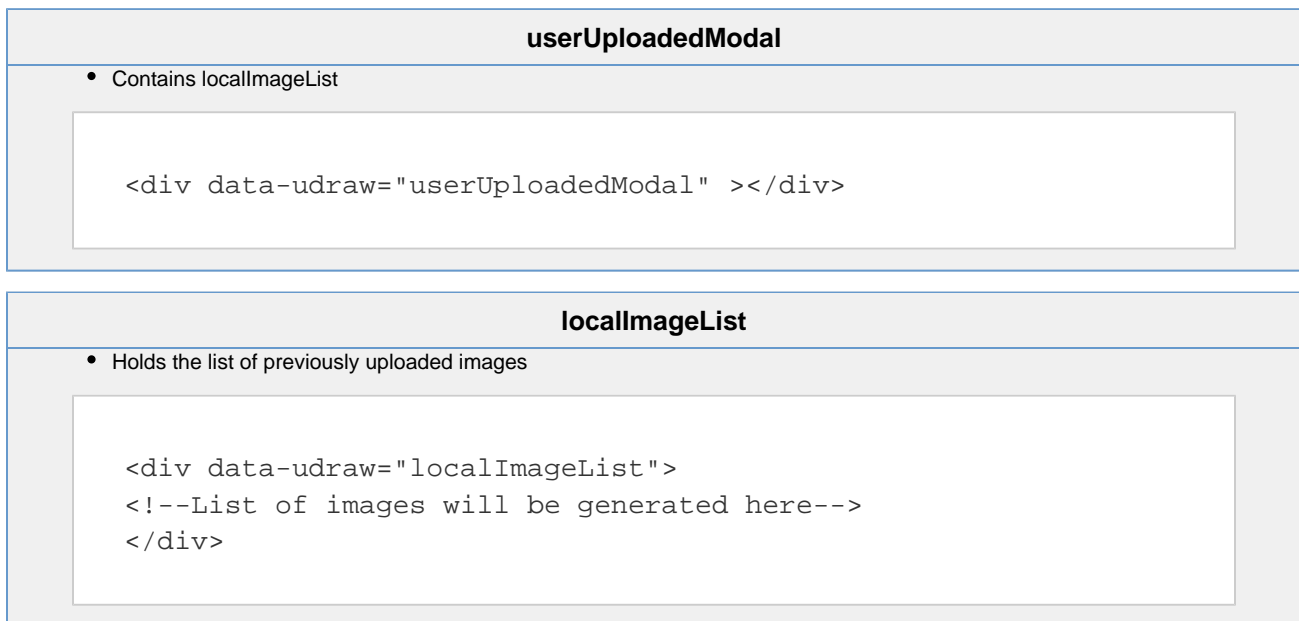
- [Image Toolboxes](#)
- [Text Toolbox](#)
- [Misc Toolboxes](#)

Image Toolboxes

Quick jump to the different toolboxes:

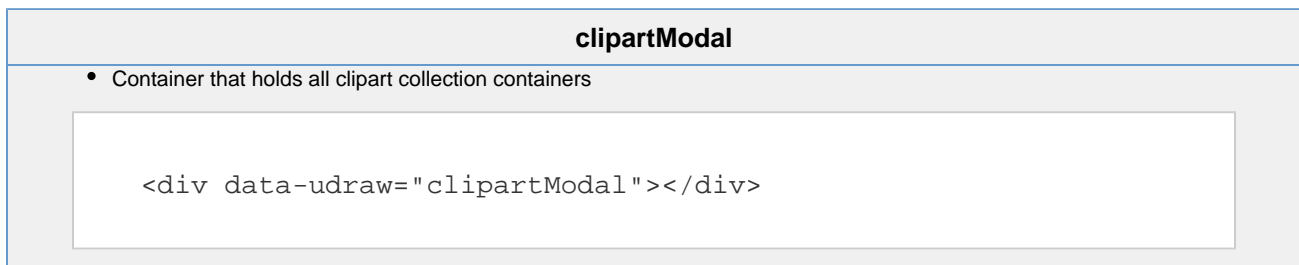
- [User Uploaded Images](#)
- [Clipart Collection](#)
- [Private Clipart Collection](#)
- [Facebook Modal](#)
- [Instagram Modal](#)
- [Image Filters](#)
- [Image Clipping](#)
- [Image Cropping](#)
- [Image Colouring](#)

User Uploaded Images



[Back to top](#)

Clipart Collection



uDrawClipartButton, openClipartButton

- Opens the named container and closes the other

```
<a href="#" data-udraw="uDrawClipartButton"><span  
data-il8n="[html]button_label.udraw-clipart"></span></a>  
<a href="#" data-udraw="openClipartButton"><span  
data-il8n="[html]button_label.open-clipart"></span></a>
```

uDrawClipartFolderContainer

- Contains list of categories of uDraw clipart
- List of categories will be generated automatically

```
<div data-udraw="uDrawClipartFolderContainer"></div>
```

uDrawClipartList

- Contains images of selected category
- List of images will be generated automatically

```
<div data-udraw="uDrawClipartList"></div>
```

clipartFolderList

- Navigation to the current uDraw category

```
<ol class="breadcrumbs" data-udraw="clipartFolderList"></ol>
```

openClipartContainer

- Container for openClipartList
- Required to show / hide openClipartList

```
<div data-udraw="openClipartContainer" style="display:  
none;"></div>
```

openClipartList

- Contains list of images retrieved from open-clipart.org

```
<div data-udraw="openClipartList"></div>
```

openClipartPageContainer

- Contains tools to browse through open-clipart's collection by page

```
<div data-udraw="openClipartPageContainer"></div>
```

openClipartPrevious, openClipartNext

- Buttons to navigate through the collection of images from open-clipart.org

```
<a href="#" class="btn btn-default btn-sm"
data-udraw="openClipartPrevious"><span
data-il8n="[html]common_label.previous"></span></a>
<a href="#" class="btn btn-default btn-sm"
data-udraw="openClipartNext"><span
data-il8n="[html]common_label.next"></span></a>
```

openClipartPageSelect

- Drop down for selection of pages
- Must use the `<select>` element

```
<select data-udraw="openClipartPageSelect"></select>
```

openClipartGoButton

- Goes to the selected page, selected from openClipartPageSelect

```
<a href="#" class="btn btn-default btn-sm"
data-udraw="openClipartGoButton"><span
data-il8n="[html]common_label.go"></span></a>
```

searchOpenClipartContainer

- Container to hold the search textbox for open-clipart

```
<div data-udraw="searchOpenClipartContainer" style="display: none;"></div>
```

searchOpenClipartInput

- Text input to enter the search term for open-clipart collection

```
<input type="text" data-i18n="[placeholder]text.search-by-word" data-udraw="searchOpenClipartInput" />
```

searchOpenClipartButton

- Button start the searching process with the entered term

```
<a href="#" class="btn btn-default btn-sm" data-udraw="searchOpenClipartButton"><span data-i18n="[html]button_label.search"></span></a>
```

[Back to top](#)

Private Clipart Modal

privateClipartModal

- Container that holds all private clipart collection containers

```
<div data-udraw="privateClipartModal"></div>
```

privateClipartFolderContainer

- Holds a list of private clipart categories
- Will be generated automatically

```
<div data-udraw="privateClipartFolderContainer"></div>
```

privateClipartList

- Holds a list of images in the current private clipart category

```
<div data-udraw="privateClipartList"></div>
```

privateClipartFolderList

- Shows the navigation to the current private clipart category

```
<ol class="breadcrumb" data-udraw="privateClipartFolderList"></ol>
```

[Back to top](#)


Facebook Modal

facebookModal

- Container that holds Facebook photos browsing tools

```
<div data-udraw="facebookModal"></div>
```

Facebook login / logout button

-  No data attribute tags

```
<div id="facebook-login">
  <div id="fb-root"></div>
  <div class="fb-login-button" data-scope="user_photos"
    data-max-rows="1" data-size="medium" data-show-faces="false"
    data-auto-logout-link="true" onlogin="javascript:
    RacadDesigner.Facebook.OnLoginLogout()"></div>
</div>
```

facebookYourPhotos, facebookTaggedPhotos

- Displays your Facebook photos, or the ones that you are tagged in, respectively

```
<a href="#" data-udraw="facebookYourPhotos">Your Photos</a>
<a href="#" data-udraw="facebookTaggedPhotos">Photos of You</a>
```

facebookImages

- Container that will hold all the images
- List of images will be generated automatically

```
<div data-udraw="facebookImages">
  <!--You may put a message here to prompt the user to log in-->
</div>
```

facebookPaging

- Pagination for Facebook photos

```
<div data-udraw="facebookPaging"></div>
```

[Back to top](#)

Instagram Modal

instagramModal

- Container that holds Instagram photo browsing tools

```
<div data-udraw="instagramModal"></div>
```

instagramLogin, instagramLogout

- Buttons to log in and out of Instagram

```
<a href="#" data-udraw="instagramLogin">Login / Authenticate</a>
<a href="#" data-udraw="instagramLogout" style="display:
none;">Logout</a>
```

instagramContent

- Container that will hold the images
- Will generate list automatically

```
<div data-udraw="instagramContent"></div>
```

instagramSearchContainer

- Holds the Instagram search tools
- Will hide if user is not logged in

```
<div data-udraw="instagramSearchContainer" style="display: none;"></div>
```

instagramSearchInput

- Instagram search input
- Must use <input> element

```
<input type="text" data-udraw="instagramSearchInput" />
```

instagramSearchButton

- Button to trigger the search with the input term

```
<a href="#" data-udraw="instagramSearchButton" class="btn btn-default" data-i18n="[html]button_label.search"></a>
```

[Back to top](#)

QR Code Modal

qrModal

- Container for QR code creation tools

```
<div data-udraw="qrModal"></div>
```

qrInput

- Input for domain
- Must be <input> element

```
<input type="text" cYvalue="http://somedomain" data-udraw="qrInput" />
```

qrColourPicker

- Colour picker for the QR code

```
<input type="hidden" value="#000000" data-udraw="qrColourPicker" />
```

qrRefreshButton

- Regenerates the QR image

```
<a href="#" class="btn btn-success btn-sm" data-udraw="qrRefreshButton">
  <i class="fa fa-refresh"></i>
  &nbsp;<span data-i18n="[html]common_label.refresh"></span>
</a>
```

qrPreviewContainer

- Contains the QR image

```
<div data-udraw="qrPreviewContainer"></div>
```

qrAddButton

- Adds the QR image to canvas

```
<a href="#" class="btn btn-success" data-udraw="qrAddButton"><span
data-i18n="[html]common_label.add"></span></a>
```

[Back to top](#)

Image Filters

imageFilterModal

- Contains all the image filters

```
<div data-udraw="imageFilterModal"></div>
```

grayscale, sepiaPurple, sepiaYellow, sharpen, emboss, blur, invert

- Applies described filter to selected image
- Requires the class "image-filter-btn"

```
<a href="#" data-udraw="grayscale" class="btn image-filter-btn" data-il8n="[html]button_label.grayscale"></a>
```

tint, brightness, noise, pixelate

- Applies described filter to selected image
- Opens respective toolbox (**imageTintContainer**, **imageBrightnessContainer**, **imageNoiseContainer**, **imagePixelateContainer** - these containers require the class "image-filter-container") which contains their respective slider / selector for adjustments (**tintColourPicker**, **imageBrightnessLevel**, **imageNoiseLevel**, **imagePixelateLevel** - tintColourPicker is an input element, whereas the other 3 are div elements that require the class "slider-class")

```
<div data-udraw="imageTintContainer" class="image-filter-container">
  <input type="hidden" data-opacity="1" data-udraw="tintColourPicker" />
</div>
```

```
<a href="#" data-udraw="brightness" class="btn image-filter-btn" data-il8n="[html]button_label.brightness"></a>
<div data-udraw="imageBrightnessContainer" class="image-filter-container">
  <div class="slider-class" style="width: 90%" data-udraw="imageBrightnessLevel"></div>
</div>
```

opacityLevel

- Slider that determines the opacity level of the selected image
- Div element that requires the class "slider-class"

```
<div style="padding-top: 10px; padding-bottom: 5px; display: inline-block; width: 50%;">
  <div class="slider-class" id="image-opacity-slider" style="width: 100%" data-udraw="opacityLevel"></div>
</div>
```

[Back to top](#)

Image Clipping

imageClippingModal

- Holds image clipping tools

```
<div data-udraw="imageClippingModal"></div>
```

imageClippingSelection

- Selects the mask shape - must be <select> element
- Currently only circle, rectangle, and triangle are supported as shapes

```
<select data-udraw="imageClippingSelection">
  <option value="Circle"
    data-il8n="[html]menu_label.circle-shape"
    selected="selected"></option>
  <option value="Rectangle"
    data-il8n="[html]menu_label.rect-shape"></option>
  <option value="Triangle"
    data-il8n="[html]menu_label.triangle-shape"></option>
</select>
```

applyImageClippingMask, removeImageClippingMask

- Applies / removes the clipping mask, respectively

```
<a href="#" data-udraw="applyImageClippingMask"><span
  data-il8n="[html]button_label.clip-image"></span></a>
<a href="#" data-udraw="removeImageClippingMask"><span
  data-il8n="[html]button_label.clip-image-remove"></span></a>
```

imageClippingOffsetUp, imageClippingOffsetDown, imageClippingOffsetLeft, imageClippingOffsetRight

- Moves the image inside the clipping mask - requires the class "clip-image-offset-btn"

```
<a href="#" class="btn clip-image-offset-btn"
  data-udraw="imageClippingOffsetUp">
  <i class="fa fa-chevron-up"></i>
</a>
```

[Back to top](#)

Image Cropping

cropModal

- Holds image cropping tools

```
<div data-udraw="cropModal"></div>
```

crop_preview

- Contains the cropping canvas, which will be automatically generated

```
<div data-udraw="crop_preview"></div>
```

image_crop

- Will contain the image after cropping for use
- Must be the `` element

```

```

crop_apply, crop_cancel

- Applies the cropped area, or closes the cropping tool window, respectively

```
<a href="#" data-udraw="crop_apply"><span data-i18n="[html]common_label.apply"></span></a>
```

[Back to top](#)

Image Colouring

imageColouringModal

- Contains imageColourContainer

```
<div data-udraw="imageColouringModal"></div>
```

imageColourContainer

- Shows the available colour options for the selected path-group

```
<div data-udraw="imageColourContainer"></div>
```

[Back to top](#)

Text Toolbox

textModal

- Container which holds the text tools

```
<div data-udraw="textModal"></div>
```

textArea

- Used for editing the text of a text object
- Must use <textarea> element

```
<textarea class="form-control" data-udraw="textArea"></textarea>
```

fontFamilyContainer

- Used for holding [fontFamilySelector](#)

```
<div data-udraw="fontFamilyContainer"></div>
```

fontSizeContainer

- Used for holding [fontSizeSelector](#)

```
<div data-udraw="fontSizeContainer"></div>
```

fontHeightSelector

- Changes the line height for text objects
- Must use <select> element
- List of numbers is generated automatically

```
<select class="dropdownList"
data-udraw="fontHeightSelector"></select>
```

fontHeightContainer

- Used for holding fontHeightSelector

```
<div data-udraw="fontHeightContainer"></div>
```

boldButton, italicButton, underlineButton, overlineButton, strikeThroughButton

- Applies described style to selected text object(s)

```
<a href="#" class="btn" data-udraw="boldButton"
data-i18n="[html]text_label.bold"></a>
```

fontStyleContainer

- Used for holding font styling buttons, such as bold

```
<div data-udraw="fontStyleContainer"></div>
```

textAlignLeft, textAlignCenter, textAlignRight, textAlignJustify

- Applies the described text alignment to the selected text object(s)

```
<a data-udraw="textAlignLeft"><div class="innerAnchorDiv"><i
class="fa fa-align-left fa-2x"></i></div></a>
```

fontAlignContainer

- Contains the text alignment buttons

```
<div data-udraw="fontAlignContainer"></div>
```

curvedTextContainer

- Holds tools to adjust your curved text object

```
<div data-udraw="curvedTextContainer"></div>
```

curvedTextSpacing

- Adjusts the spacing in between each letter in a curved text object
- Must have the class "slider-class"

```
<div class="slider-class" data-udraw="curvedTextSpacing"></div>
```

curvedTextRadius

- Adjusts the curving of your curved text object
- Must have the class "slider-class"

```
<div class="slider-class" data-udraw="curvedTextRadius"></div>
```

reverseCurve

- Flips the direction of the curve in a curved text object

```
<a href="#" data-udraw="reverseCurve"><span  
data-il8n="[html]button_label.flip-curve"></span></a>
```

Misc Toolboxes

Quick jump to the different toolboxes:

- [Layers Toolbox](#)
- [Pages Toolbox](#)
- [Polygon Toolbox](#)
- [Gradient Toolbox](#)
- [Shadow Toolbox](#)
- [Labels Toolbox](#)
- [Linked Templates](#)

- Multilayer Image
- Image Fill Toolbox
- Freedraw Toolbox

Layers Toolbox

The positioning of the objects on canvas can be reorganized with layers. Our layering system also allows for some finer object manipulation.

layersModal

- Automatically shows when objects are on canvas
- Contains the required layersContainer

<div data-udraw="layersModal"></div>

layersRefresh

- Rebuilds the list of layers

[Back to top](#)

Pages Toolbox

- Contains pagesContainer and pagesList

<div data-udraw="pagesModal"></div>

[Back to top](#)

Polygon Toolbox

polygonModal

- Container for polygon creation tool

<div data-udraw="polygonModal"></div>

polygonSideSelector

- Input for the number of sides the polygon will have
- Must use <input> element

<input type="number" min="3" value="3"
data-udraw="polygonSideSelector" />

polygonCreate

- Adds a polygon with the selected amount of sides to the canvas

```
<a href="#" class="btn btn-success" tabindex="3"
data-udraw="polygonCreate"><span
data-i18n="[html]common_label.create"></span></a>
```

polygonCancel

- Cancels polygon creation

```
<a href="#" class="btn btn-danger" data-dismiss="modal"
data-udraw="polygonCancel"><span
data-i18n="[html]common_label.cancel"></span></a>
```

[Back to top](#)

Gradient

gradientModal

- Contains the gradientContainer

```
<?php if (!$udraw_settings['designer_disable_text_gradient']) {
?>
<div data-udraw="gradientModal"></div><?php } ?>
```

gradientContainer

- Gradient colour picker will be generated in this container

```
<div data-udraw="gradientContainer"></div>
```

[Back to top](#)

Shadow Toolbox

shadowModal

- Contains the shadow adjustment tools

```
<div data-udraw="shadowModal"></div>
```

shadowOffsetX, shadowOffsetY, shadowBlur

- Adjusts described shadow property on selected object; Will automatically add shadow on selected object if it does not have shadow
- Must have class "slider-class"
- Must be <div> element

```
<div class="slider-class" data-udraw="shadowOffsetX"></div>
```

shadowRemove

- Removes shadow on selected object

```
<a href="#" class="btn" data-udraw="shadowRemove"><span data-il8n="[html]button_label.remove-shadow"></span></a>
```

[Back to top](#)

Labels Toolbox

layerLabelsModal

- Contains layerLabelsContent
- Will appear if labels are detected and are assigned to objects on page

```
<div data-udraw="layerLabelsModal"></div>
```

layerLabelsContent

- Contains text inputs for text objects, and image upload buttons for images; These are generated automatically

```
<div data-udraw="layerLabelsContent"></div>
```

[Back to top](#)

Linked Templates

linkedTemplatesModal

- Contains linkedTemplatesContainer
- Will automatically show if linked templates are detected

```
<div data-udraw="linkedTemplatesModal"></div>
```

linkedTemplatesContainer

- Contains the list of linked templates (by template tag); These will be generated automatically.

```
<div data-udraw="linkedTemplatesContainer"></div>
```

[Back to top](#)

Multilayer Image

multilayerImageModal

- Contains multilayerImageContainer
- Will automatically show if multilayer image is detected

```
<div data-udraw="multilayerImageModal"></div>
```

multilayerImageContainer

- Contains list of linked images on page
- Must use the element

```
<ul data-udraw="multilayerImageContainer"></ul>
```

[Back to top](#)

Image Fill Toolbox

objectColouringModal

- Contains tools needed for using image fill

```
<div data-udraw="objectColouringModal"></div>
```

objectColouringUpload

- Uploads images to the patterns folder for image fill usage
- Must use <input type="file">

```
<input type="file" name="files[]" multiple  
data-udraw="objectColouringUpload" />
```

triggerObjectColouringUpload

- Triggers objectColouringUpload, if you wish to hide the original upload button

```
<a href="#" data-udraw="triggerObjectColouringUpload"></a>
```

objectColouringFillContainer, objectColouringStrokeContainer

- Contains list of images you may use for object fill or stroke

```
<div data-udraw="objectColouringFillContainer"></div>
```

[Back to top](#)

Freedraw Toolbox

freedrawModal

- Contains freedrawing tools

```
<div data-udraw="freedrawModal"></div>
```

brushSelection

- Brush style selection
- Must use the <select> element
- Only supporting the Pencil and Circle brushes at the moment

```
<select data-udraw="brushSelection">
  <option value="Pencil" selected="selected"
data-il8n="[html]select_text.pencil"></option>
  <option value="Circle">Circle</option>
</select>
```

brushColourPicker

- Colour picker for brush
- Must use <input> element

```
<input type="hidden" value="#000000" data-opacity="1"
data-udraw="brushColourPicker" />
```

brushSize

- Brush size selector
- Must use the `<input type="number">` element

```
<input type="number" value="1" min="1" max="25"
data-udraw="brushSize" />
```

brushShadowSize

- Shadow size selector
- Must use the `<input type="number">` element

```
<input type="number" value="0" min="0" max="50"
data-udraw="brushShadowSize" />
```

brushShadowDepth

- Shadow depth selector
- Must use the `<input type="number">` element

```
<input type="number" value="1" min="1" max="25"
data-udraw="brushShadowDepth" />
```

[Back to top](#)

Optional Tags

The following tags will not make a big impact on the Designer. However, some are experimental and may not work as intended.

Quick jump to sections:

- [Designer Version](#)
- [Grid](#)
- [Ruler](#)
- [Image Fill](#)
- [Freedraw](#)
- [Object Rotation](#)
- [Object Scaling](#)
- [Rectangle Corner](#)
- [Advanced Text](#)

Designer Version

designerVersion

- Displays the current version of the Designer.

```
<span data-udraw="designerVersion"></span>
```

[Back to top](#)

Grid

snapToGrid

- Toggles snapCheckbox

```
<a href="#" data-udraw="snapToGrid"></a>
```

snapCheckbox

- Ensures that the objects will move along the grid
- Will show grid overlay
- Must use the `<input type="checkbox">` element

```
<input type="checkbox" data-udraw="snapCheckbox">
```

toggleGridLines

- Toggles gridCheckbox

```
<a href="#" data-udraw="toggleGridLines"></a>
```

gridCheckbox

- Toggles grid overlay
- Must use the `<input type="checkbox">` element

```
<input type="checkbox" data-udraw="gridCheckbox">
```

[Back to top](#)

Ruler

toggleRuler

- Toggles rulers

```
<a href="#" data-udraw="toggleRuler"></a>
```

topRuler, sideRuler

- Measures the size of the canvas; includes mouse tracking line
- Must use the <canvas> element

```
<canvas data-udraw="topRuler"></canvas>
```

[Back to top](#)

Image Fill



This feature is currently not supported in PDF production.

There is no data attribute tag set to show the toolbox of this feature at them moment. Please set one which will use the following function:

```
RacadDesigner.ShowAdvancedColouringOptions();
```

- Requires `objectColouringModal`

[Back to top](#)

Freedraw

freedrawButton

- Shows the freedrawing toolbox
- Requires `freedrawModal`

```
<a href="#" data-udraw="freedrawButton"></a>
```

[Back to top](#)

Object Rotation

objectRotationButton

- Button to open the rotation container

```
<a href="#" data-udraw="objectRotationButton"></a>
```

objectRotationContainer

- Container to hold rotation tools

```
<div data-udraw="objectRotationContainer"></div>
```

objectRotationLabel

- Displays the angle of the selected object

```
<div data-udraw="objectRotationLabel"></div>
```

objectRotationSelector

- Slider to change the angle of the selected object
- Must have class "slider-class"

```
<div class="slider-class"  
data-udraw="objectRotationSelector"></div>
```

objectRotationClose

- Button to close the container

```
<a href="#" data-udraw="objectRotationClose"></a>
```

[Back to top](#)

Object Scaling

objectScaleButton

- Button to open the scaling container

```
<a href="#" data-udraw="objectScaleButton"></a>
```

objectScaleContainer

- Container to hold scaling tools

```
<div data-udraw="objectScaleContainer"></div>
```

objectScaleLabel

- Displays the scale of the selected object

```
<div data-udraw="objectScaleLabel"></div>
```

objectScaleSelector

- Slider to change the scale of the selected object
- Must have class "slider-class"

```
<div class="slider-class"  
data-udraw="objectScaleSelector"></div>
```

objectScaleClose

- Button to close the container

```
<a href="#" data-udraw="objectScaleClose"></a>
```

[Back to top](#)

Rectangle Corner

rectangleCornerContainer

- Container to hold rectangle corner rounding tools

```
<div data-udraw="rectangleCornerContainer"></div>
```

rectangleCornerSelector

- Selects the "roundness" of the corners
- Must use `<input type="number">`

```
<input type="number" min="0" max="50" step="1" data-udraw="rectangleCornerSelector" />
```

rectangleCornerClose

- Closes the container

```
<a href="#" data-udraw="rectangleCornerClose"></a>
```

[Back to top](#)

Advanced Text



This feature is still experimental. Use at your own risk.

editTextAdvancedText

- Converts the selected text object into an advanced text object
- Requires advancedTextContainer, to be contained in [textModal](#)

```
<a href="#" data-udraw="advancedTextContainer"></a>
```

advancedTextContainer

- Contains the advanced text tools

```
<div data-udraw="advancedTextContainer"></div>
```

letterSpaceIncrease, letterSpaceDecrease

- Increases / decreases the space in between each letter

```
<a href="#" data-udraw="letterSpaceIncrease"></a>
```

letterSpaceReset

- Resets the letter spacing

```
<a href="#" data-udraw="letterSpaceReset"></a>
```

textEffectsReset

- Removes any special effects applied to the text

```
<a href="#" data-udraw="textEffectsReset"></a>
```

editTextAdvancedText

- Converts the selected text object into an advanced text object
- Requires advancedTextContainer, to be contained in [textModal](#)

```
<a href="#" data-udraw="advancedTextContainer"></a>
```

textEffectsSelection

- Selection of text effects
- Must use <select>

```
<select data-udraw="textEffectsSelection">
  <option value="inflated">
data-i18n="[html]select_text.inflate"></option>
    <option value="deflated">
data-i18n="[html]select_text.deflate"></option>
    <option value="bridgeCurveUp">
data-i18n="[html]select_text.bridgeCurveUp"></option>
    <option value="bridgeCurveDown">
data-i18n="[html]ui-controls.bridgeCurveDown"></option>
    <option value="chevronUp">
data-i18n="[html]select_text.chevronUp"></option>
    <option value="chevronDown">
data-i18n="[html]select_text.chevronDown"></option>
    <option value="fadeLeft">
data-i18n="[html]select_text.fadeLeft"></option>
    <option value="fadeRight">
data-i18n="[html]select_text.fadeRight"></option>
    <option value="fadeUp">
data-i18n="[html]select_text.fadeUp"></option>
    <option value="fadeDown">
data-i18n="[html]select_text.fadeDown"></option>
    <option value="triangleUp">
data-i18n="[html]select_text.triangleUp"></option>
    <option value="triangleDown">
data-i18n="[html]select_text.triangleDown"></option>
    <option value="wave">
data-i18n="[html]select_text.wave"></option>
</select>
```

textEffectsLevel

- Increases / decreases the effect on text
- Must use <input type="number">

```
<input type="number" min="0" max="10" step="1" value="0"
data-udraw="textEffectsLevel" />
```

[Back to top](#)

Localization Text

We generally use a localization text file to display text instead of directly typing into the HTML. We have included the contents of our English file as a reference.

Usage:

```
<span data-i18n="[html]object_name.property_name"></span>
```

Example:

```
<span data-i18n="[html]common_label.close"></span>
```

is equivalent to

```
<span>Close</span>
```

Localization JSON object