IP.Board 3.2 Front-End Skin

IP.Board 3.2 introduces a new front end skin with many new CSS classes.

This document lists the classes available to you which should be used when designing applications for IP.Board to ensure your interface is consistent and visually appealing.

DISPLAYING TEXT

Page Headings

- All pages **must** have a h1 tag containing the page title.
- The page title must use the “ipsType_pagetitle” class

When the page is user-generated content, should be inside a div with class “ipsBox_withphoto” below the photo of the user of size medium (see “User Photos” section).

```
<h1 class='ipsType_pagetitle'>Page Title</h1>
```

Page Title

```
<a href='{parse url="showuser={$member['member_id']}"
seotitle="{$member['members_seo_name']}" template="showuser"
base="public"}' class='ipsUserPhotoLink'>
     <img src='{$member['pp_small_photo']}' class='ipsUserPhoto
ipsUserPhoto_medium left' />
</a>
<div class='ipsBox_withphoto'>
     <h1 class='ipsType_pagetitle'>Page Title</h1>
</div>
```



Page Title

Sub Headings

- Should be used when there is more than one section on a page
- Should use “ipsType_subtitle” class
- May use any “h” tag (e.g. “h2”, “h3”) so long as it makes sense in the HTML.

```
<h2 class='ipsType_subtitle'>Subtitle</h2>
```

Subtitle

Page Description

- Should be used for displaying a description of the page, section rules, or information about when the page was created and by whom.
- May use either a p or div.
- You may also add the class “blend_links” to make links the same color as the description and not blue. This should be used for user links.

```
<h1 class='ipsType_pagetitle'>Page Title</h1>
<p class='ipsType_pagedesc'>Description</p>
```



Small Fonts & Other Descriptions

- You should not use style tags to define small font sizes
- Small font sizes may be used by adding either of the following classes:
    - ipsType_small (12px)
    - ipsType_smaller (11px)
- If the content being defined is a description, it should use “ipsType_pagedesc” if it is a description for the entire page (see “Page Description” section) or “desc” for other descriptions.
- If using “desc”, “ipsType_x” classes are not necessary, it will be automatically smaller.
- “desc lighter” for a lighter description when the content is less important.
- When using “desc” or “desc ligher”, you may also add the class “blend_links” to make links the same color as the description and not blue. This should be used for user links.

```
<p class='ipsType_small'>Small Text</p>
```




```
<p class='desc'>Description</p>
```




```
<p class='desc lighter'>Lighter Description</p>
```



UTILITY CLASSES

Padding

- You should not use style tags to define padding
- Padding may be used by adding any of the following classes:
    - ipsPad (9px)
    - ipsPad_double (19px)
    - ipsPad_half (4px)

Layouts

- Can be used to create pages with sidebars
- Should start with a div with the class “ipsLayout”, plus either or both of the following additional classes:
    - ipsLayout_withleft - if a left sidebar should be used
    - ipsLayout_withright - if a right sidebar should be used
- The size of the sidebar can be adjusted by adding any of the following classes, replacing *<bar>* with *left* or *right*. You may specify different sizes for the different bars (the default is 200px):
    - ipsLayout_tiny*<bar>* (40px)
    - ipsLayout_small*<bar>* (140px)
    - ipsLayout_large*<bar>* (270px)
    - ipsLayout_big*<bar>* (320px)
    - ipsLayout_huge*<bar>* (370px)
- The content will be in divs inside this div with one of the following classes:
    - ipsLayout_left - for the left sidebar
    - ipsLayout_right - for the right sidebar
    - ipsLayout_content - for the main content

```
<div class='ipsLayout ipsLayout_withleft'>
     <div class='ipsLayout_left'>
          <div class='ipsSideBlock'>
               <h3>Side Block</h3>
               I am a side block
          </div>
     </div>
     <div class='ipsLayout_content'>
          <div class='ipsBox'>
               <div class='ipsBox_container ipsPad'>
                    I am the main content
               </div>
          </div>
     </div>
</div>
```




```
<div class='ipsLayout ipsLayout_withleft ipsLayout_withright
ipsLayout_smallleft ipsLayout_largeright'>
     <div class='ipsLayout_left'>
          <div class='ipsSideBlock'>
               <h3>Side Block</h3>
               I am a small side block
          </div>
     </div>
     <div class='ipsLayout_right'>
          <div class='ipsSideBlock'>
               <h3>Side Block</h3>
               I am a large side block
          </div>
     </div>
     <div class='ipsLayout_content'>
          <div class='ipsBox'>
               <div class='ipsBox_container ipsPad'>
                    I am the main content
               </div>
          </div>
     </div>
</div>
```



left, right and clearfix

- Used to float items
- Should not be used for creating sidebars (see “Layout” section for alternative)
- Floating content should be inside div with class “clearfix”

```
<div class='clearfix'>
     <span class='left'>Left</span>
     <span class='right'>Right</span>
</div>
Content Below
```



Tooltips

- Should be used to display important tooltips.
- Can be added to any tag.
- Add the “data-tooltip” attribute with the message ass the content.

```
<span data-tooltip="This is a tooltip">Hello, World!</span>
```



Textboxes with default content

- Should be used where a textbox has default content.
- See example for use.

```
<input id='my_textbox' class='input_text' />

<script type='text/javascript'>
$('my_textbox').defaultize( "The placeholder string" );
</script>
```



USER-INTERFACE ELEMENTS

Buttons

- For buttons that submit a form, the class “input_submit” should be used.
- Pages which have actions that can be performed (for example “Start New Topic”, etc.) should have a list of buttons with the class “topic_buttons”. This should come after the page title and further navigation (for example, subforums), but before the content the buttons are pertaining to (for example, the buttons should show above the topic list). This list will be automatically right-aligned, meaning buttons will show in the reverse order to how they are listed.
- The less important actions in the “topic_buttons” list should use the class “non_button”. Ideally, only one button should not have this class.
- Buttons in the “topic_buttons” list which are disabled may use the “disabled” class.
- Most pages should have a primary action - if the page displays a form, it should be a button with the “input_submit” class. For other pages, it should be one of the buttons in topic_buttons list.
- Other buttons (i.e. not to submit a form or used in the “topic_buttons” list, may use either of the following classes:
    - ipsButton - for dark grey buttons
    - ipsButton_secondary - for less imposing buttons

```
<ul class='topic_buttons'>
     <li><a href='#'>Button</a></li>
     <li class='disabled'><a href='#'>Disabled Button</a><li>
     <li class='non_button'><a href='#'>Less important Button</
a></li>
</ul>
```




```
<form>
     <input type='submit' class='input_submit' value='Submit' />
</form>
```




```
<a href='#' class='ipsButton'>Button</a>
<a href='#' class='ipsButton_secondary'>Secondary Button</a>
```



Messages

- Used to display confirmation and error messages
- Should use class “message” and optionally one of the following additional classes:
    - error
    - unspecific

```
<p class='message'>Message</p>
<br />
<p class='message error'>Error</p>
<br />
<p class='message unspecific'>Unspecific Message</p>
```



User Photos

- Links that contain a user photo must use the class “ipsUserPhoto_link”
- The image tag itself should have one of the following classes:
    - ipsUserPhoto_icon (16x16px)
    - ipsUserPhoto_tiny (20x20px)
    - ipsUserPhoto_mini (30x30px)
    - ipsUserPhoto_medium (50x50px)
    - ipsUserPhoto_large (90x90px)

```
<a href='{parse url="showuser={$member['member_id']}"
seotitle="{$member['members_seo_name']}" template="showuser"
base="public"}' class='ipsUserPhotoLink'>
     <img src='{$member['pp_small_photo']}' class='ipsUserPhoto
ipsUserPhoto_medium left' />
```

```
</a>
```



Badges

- Should be used to add tags to content, for example, to highlight queued content.
- Should use the class “ipsBadge”, plus one of the following classes:
    - ipsBadge_green
    - ipsBadge_purple
    - ipsBadge_grey
    - ipsBadge_lightgrey
    - ipsBadge_orange
    - ipsBadge_red

```
<p class='ipsBadge ipsBadge_green'>FEATURED</p>
<p class='ipsBadge ipsBadge_purple'>FAVORITE</p>
<p class='ipsBadge ipsBadge_grey'>OFFLINE</p>
<p class='ipsBadge ipsBadge_lightgrey'>PREFIX</p>
<p class='ipsBadge ipsBadge_orange'>QUEUED</p>
<p class='ipsBadge ipsBadge_red'>DELETED</p>
```



Wizards

- Should be used when there are multiple steps to a process (for example, registration)
- Should be displayed above the h1 tag
- Should be a ul tag (each li is a step) inside a container div with the classes “ipsSteps” and “clearfix” (so that it extends to the end of the page).
- The active step should use the class “ipsSteps_active”.
- All completed steps should use the class “ipsSteps_done”.
- The li tag should contain three elements:
    - A strong tag with the class “ipsSteps_title” containing the step number.
    - A span with the class “ipsSteps_desc” containing a description of the step.
    - A span with the class “ipsSteps_arrow” containing a

```
<div class='ipsSteps clearfix'>
     <ul>
          <li class='ipsSteps_done'>
               <strong class='ipsSteps_title'>Step 1</strong>
               <span class='ipsSteps_desc'>Description</span>
```

```
            <span class='ipsSteps_arrow'> </span>
        </li>
        <li class='ipsSteps_active'>
            <strong class='ipsSteps_title'>Step 2</strong>
            <span class='ipsSteps_desc'>Description</span>
            <span class='ipsSteps_arrow'> </span>
        </li>
        <li>
            <strong class='ipsSteps_title'>Step 3</strong>
            <span class='ipsSteps_desc'>Description</span>
            <span class='ipsSteps_arrow'> </span>
        </li>
    </ul>
</div>
```

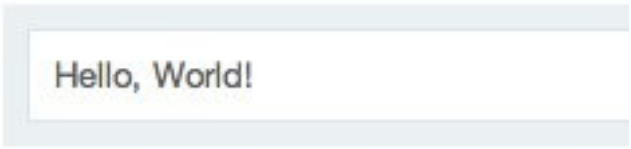


DISPLAYING CONTENT

Content Boxes

- All content should be inside a div with “ipsBox”, and a div underneath with “ipsBox_container”.
- The div with class “ipsBox_container” should in most circumstances also use the class “ipsPad” (see “Padding” section)

```
<div class='ipsBox'>
     <div class='ipsBox_container ipsPad'>
          Hello, World!
     </div>
</div>
```

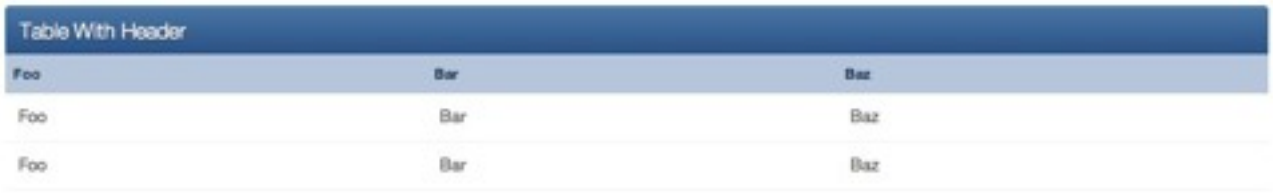


Tables

- Should use class “ipb_table”
- Should have a div above with class “maintitle” - this may be used as a title for the table or as a filter bar. If a filter bar is used, the div should also use the class “ipsFilterbar”.
- Table content should be in appropriate content box (see “Content Boxes” section).
- If a filter bar is used, it should be comprised of a list of common views, with a custom option for additional filters. The list should be a ul tag with the class “ipsList_inline” and “ipsType_smaller” - the active filter should use the class “active”.
- The “custom” option for the filter bar needs to define an id which will be used in a line of javascript to activate the dropdown menu (see the second example below). The content for the dropdown menu also needs to define an id to be used in the javascript and should use the classes “ipbmenu_content” and “ipsPad”.
- If present, the header should use class “header”.
- If the table is used to list links to content, the links should use the class “topic_title”.
- If the table contains content that can be read and/or contributed to (e.g. posted in, commented on), the far-left column should be used for displaying status icons, and unread rows should be highlighted (using class “unread”). Columns with status icons should use the class “col_f_icon”, and valid status icons are:
    - t_closed - Item is closed/locked and can no longer be contributed to.
    - t_unread_dot - Item has been contributed to by the user and contains unread items.

    - t_unread - Item contains unread items, but has not been contributed to by the user.
    - t_moved - Item has been moved to a different location and clicking it will take you to that new location.
- If the table is showing content which is queued or deleted, but visible (because the user logged in is a moderator), the row should have the class “moderated”. This will likely be accompanied by a badge (see “Badges” section).
- If a table has no content, you can use a div with the “no_messages” class instead of showing an empty table.

```
<div class='maintitle'>Table With Header</div>
<table class='ipb_table'>
     <tr class='header'>
          <th>Foo</th>
          <th>Bar</th>
          <th>Baz</th>
     </tr>
     <tr>
          <td>Foo</td>
          <td>Bar</td>
          <td>Baz</td>
     </tr>
     <tr>
          <td>Foo</td>
          <td>Bar</td>
          <td>Baz</td>
     </tr>
</table>
```




```
<div class='maintitle ipsFilterbar'>
     <ul class='ipsList_inline ipsType_smaller'>
          <li><a href='#'>Most Read</a></li>
          <li class='active'><a href='#'>Latest</a></li>
          <li><a href='#'>Last Updated</a></li>
          <li><a href='#custom_menucontent' id='custom'>Custom</
a></li>
     </ul>
</div>
<div id='custom_menucontent' class='ipbmenu_content ipsPad'>
     <form>
```

```
          <strong>Option</strong><br />
          <input />
          <br /><br />
          <strong>Option</strong><br />
          <input />
          <br /><br />
          <input type="submit" value="Submit"
class="input_submit" />
     </form>
</div>
<script type='text/javascript'>
     new ipb.Menu( $('custom'), $('custom_menucontent'),
{ stopClose: true } );
</script>
<table class='ipb_table'>
     <tr>
          <td>Foo</td>
          <td>Bar</td>
          <td>Baz</td>
     </tr>
     <tr>
          <td>Foo</td>
          <td>Bar</td>
          <td>Baz</td>
     </tr>
</table>
```






```
<div class='maintitle'>Table With Status Icons</div>
<table class='ipb_table'>
     <tr>
          <td class='col_f_icon'>{parse replacement="t_closed"}</
td>
          <td><a href='#' class='topic_title'>Locked</a></td>
          <td>Foo</td>
     </tr>
     <tr class='unread'>
```

```
          <td class='col_f_icon'>{parse replacement="t_unread_dot"}
</td>
          <td><a href='#' class='topic_title'>Unread and
contributed in</a></td>
          <td>Foo</td>
     </tr>
     <tr class='unread'>
          <td class='col_f_icon'>{parse replacement="t_unread"}</
td>
          <td><a href='#' class='topic_title'>Unread</a></td>
          <td>Foo</td>
     </tr>
     <tr>
          <td class='col_f_icon'>{parse replacement="t_moved"}</td>
          <td><a href='#' class='topic_title'>Moved</a></td>
          <td>Foo</td>
     </tr>
     <tr>
          <td class='col_f_icon'> </td>
          <td><a href='#' class='topic_title'>Normal</a></td>
          <td>Foo</td>
     </tr>
     <tr class='moderated'>
          <td class='col_f_icon'> </td>
          <td><p class='ipsBadge ipsBadge_orange'>QUEUED</p> <a
href='#' class='topic_title'>Queued</a></td>
          <td>Foo</td>
     </tr>
</table>
```




```
<div class='maintitle'>Table With No Content</div>
<div class='no_messages'>There are no messages to show</div>
```



Forms

- The form tag should contain a ul tag and one of the following additional classes:
    - ipsForm_vertical - labels will be shown above the input element
    - ipsForm_horizontal - labels will be shown next to the input element
- Each li will contain a field and should have the class “ipsField”.
- Inside the li tag, the label tag will be specified first with the class “ipsField_title”. A p or div tag with the class “ipsField_content” will then display below that, containing the input element.
- For select boxes, the class “ipsField_select” should be added tho the li tag.
- For checkboxes, the class “ipsField_checkbox” should be added to the li tag, and instead of the layout described above, the checkbox will go first (with no container) and then the p or div tag with “ipsField_content” which will contain the label, without the “ipsField_title” class.
- If the field is required, the label should contain a span with the class “ipsForm_required”, containing an asterisk.
- If a description is required in addition to the title, it should be in the “ipsField_content” p/div, in a span with the class “desc”, appearing after the input element.
- The submit button should use the class “input_submit” (see “Buttons” section) and be inside a fieldset with the class “submit”. This should appear inside the “ipsBox”, but outside the “ipsBox_container” (see “Content Boxes” section).
- If an error message needs to be displayed (for example, after submitting the form to highlight errors), it should appear inside the “ipsField_content” container, in a span with the class “error”.

```
<div class='ipsBox'>
     <form>
          <div class='ipsBox_container ipsPad'>
               <ul class='ipsForm ipsForm_vertical'>
                    <li class='ipsField'>
                         <label class='ipsField_title'>Title <span
class='ipsForm_required'>*</span></label>
                         <div class='ipsField_content'>
                              <input /><br />
                              <span class='desc'>Description</span>
                         </div>
                    </li>
                    <li class='ipsField'>
                         <label class='ipsField_title'>Title</
label>
                         <div class='ipsField_content'>
```

```
                              <input />
                         </div>
                    </li>
                    <li class='ipsField ipsField_select'>
                         <label class='ipsField_title'>Title</
label>
                         <div class='ipsField_content'>
                              <select></select><br />
                              <span class='error'>This is an
error</span>
                         </div>
                    </li>
                    <li class='ipsField ipsField_checkbox'>
                         <input type='checkbox' />
                         <div class='ipsField_content'>
                              Title
                         </div>
                    </li>
               </ul>
          </div>
          <fieldset class='submit'>
               <input type='submit' class='input_submit'
value='Submit' />
          </fieldset>
     </form>
</div>
```



The same above code but with “ipsForm_horizontal” instead of “ipsForm_vertical”:

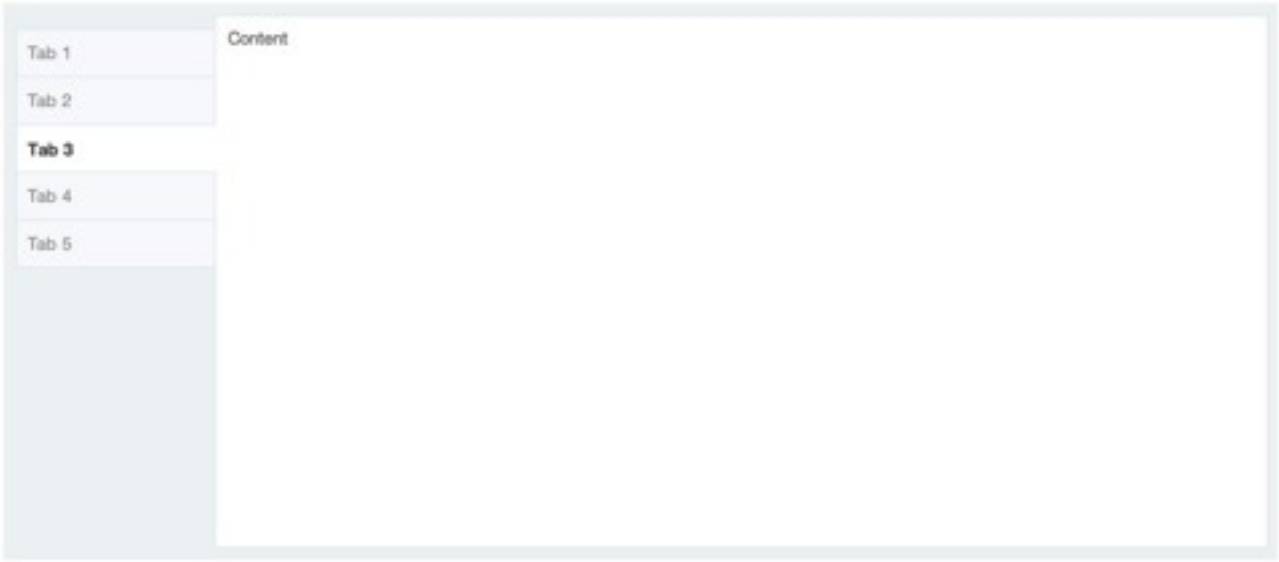


Vertical Tabs

Before using these styles, review the "Layouts" section.

- Should be used to display content in several different tabs.
- A container div should use the classes "ipsVerticalTabbed".
- The container div should also contain the layout classes. Most of the time, the left sidebar should use the *small* size.
- The list of tabs should be inside the container div, inside another div with the classes "ipsVerticalTabbed_tabs" and "ipsLayout_left". The list should use a ul tag and the active tab should use the "active" class.
- The content should be inside the container div inside another div with the classes "ipsVerticalTabbed_content" and "ipsLayout_content".

```
<div class='ipsBox'>
     <div class='ipsLayout ipsLayout_withleft ipsLayout_smallleft
ipsVerticalTabbed'>
          <div class='ipsLayout_left ipsVerticalTabbed_tabs'>
               <ul>
                    <li><a href='#'>Tab 1</a></li>
                    <li><a href='#'>Tab 2</a></li>
                    <li class='active'><a href='#'>Tab 3</a></li>
                    <li><a href='#'>Tab 4</a></li>
                    <li><a href='#'>Tab 5</a></li>
               </ul>
          </div>
          <div class='ipsLayout_content ipsVerticalTabbed_content
ipsBox_container'>
               <div class='ipsPad'>
                    Content
               </div>
          </div>
     </div>
     <br class='clear' />
</div>
```



Sideblocks

- Used for displaying hooks on the board index, or in sidebars when used with ipsLayout (see “Layouts” section).
- Uses a div with class “ipsSideBlock”.
- Uses a h3 tag for the block title.

```
<div class='ipsSideBlock'>
     <h3>Title</h3>
     Content
</div>
```

