



# **“Simply”**

## **Premium Template By iDangero.us**

### **Documentation**

Thank you for purchasing this item. If you have any questions that are beyond the scope of this help file, please feel free to contact us via support ticket form [here](#).

# Table of Contents

---

1. About “Simply”
2. Simply Files & Folders Structure
3. Installation
4. HTML Structure
5. Typography
6. Inner Linking (How it works?)
7. CSS Files and Structure
8. JavaScripts
9. Popup Window
10. Themes
11. Ajax Contact Form
12. PSD Files and Structure
13. Licensing Terms

# 1. About “Simply”

---

**Simply** is a premium Ajax based animated site template, designed by the iDangero.us team. This template is perfect for small business websites, projects, or it can be used as a fashion site.

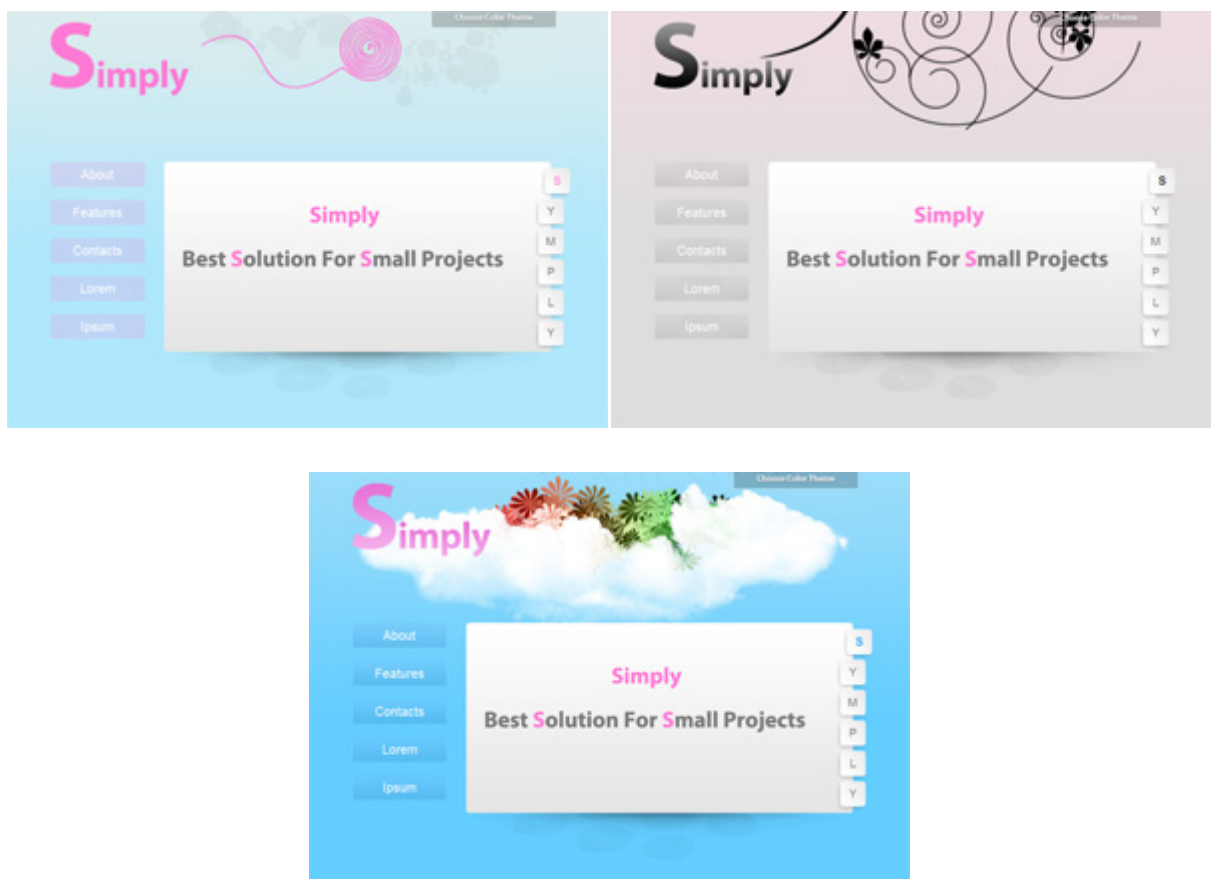
The main advantage of this template is that it uses only one page. All content pages are loading "on the fly", so your users do not need to wait the reloading of entire site. Template is fully animated without the use of Flash technology. Smooth and good looking animation and Ajax features of this template are realized with the JavaScript jQuery library.

Smooth and good looking animation and Ajax features of this template are realized by using the JavaScript jQuery library.

This template is easy to redesign, it has very simple and clear API.

Also **Simply** is powered with an Ajax Contact Form with validation and a Popup Engine.

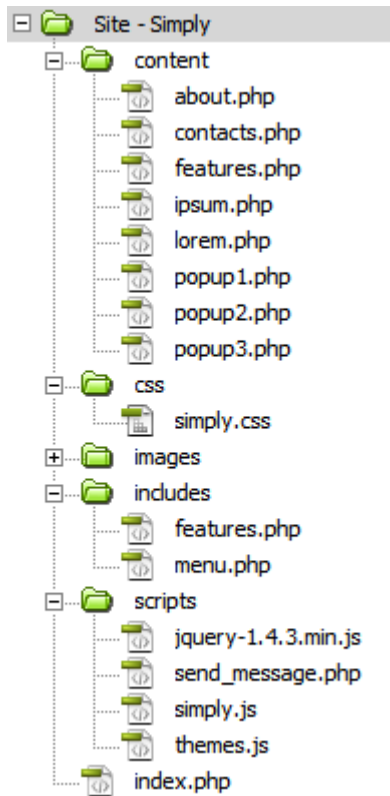
Here is a couple of screenshots:



## 2. Simply Files Structure

---

**Simply** files are located in the **simply/** folder from the downloaded archive:



**content:** in this folder there are .php files with a html content pages.

**css:** folder with a .css files

**images:** here you can find layouts images used in this Template

**includes:** 2 php files with menu and features section.

**scripts:** JavaScripts & PHP scripts used in this template

And the main **index.php** in the root folder

## 3. Installation

---

To install **Simply** template you have to copy all files & folders from the **simply/** folder to your web-site root folder on the server (It could be a real server or local testing server).

## 4. HTML Structure

Let's look at the main HTML/PHP structure of Simply Template. Open index.php file with a text editor (Highly recommended to use Adobe® Dreamweaver® or Notepad ++ editors).

**Index.php <body> structure:**

White Layer is used for themes changing

Image which appears near with menu link

Header Section – contains logo and theme switcher

```
1 <body>
2 <div class="white-layer"></div>
3 <div class="menu-decor"></div>
4 <div id="header"> <a href="index.php" title="Simply"></a>
5     <div class="theme-switcher">
6         
7         
8         
9         
10    </div>
11 </div>
12 <div class="wrapper">
13     <div class="ajax-loader"></div>
14     <div id="menu">
15         <?php include("includes/menu.php") ?>
16     </div>
17     <div id="features" <?php if(!empty($_GET['content'])) echo 'style="display:none;"' ?>
18         <div class="features-r">
19             <div class="features-l">
20                 <div class="features-mid">
21                     <?php include("includes/features.php") ?>
22                 </div>
23             </div>
24         </div>
25     </div>
26     <div class="main-content">
27         <div class="content-wrap" <?php if(!empty($_GET['content'])) echo 'style="display:block;"' ?> >
28             <div class="content">
29                 <?php
30                     if (!empty($_GET['content'])) {
31                         include ("content/".$_GET['content'].".php");
32                     }
33                 <?>
34             </div>
35             <div class="footer"><p align="center">&copy; 2011 Simply Premium Template by iDangero.us. All
   Rights Reserved.</p></div>
36         </div>
37     </div>
38 </div>
39 <?php
40 if (!empty($_GET['popup'])) {
41     $closeStyle = 'style="display:block"';
42     $windowStyle = 'style="display:block;width:600px;"';
43     $pLayerStyle = 'style="display:block;opacity:0.6;filter:alpha(opacity=60)";';
44 }
45 <?>
```

Menu Section

Features Section

Main Content Section. All content pages (files) will be loaded here

Footer Section.

We need these PHP rules to make popup window visible when the page is loaded with the "popup" variable in the site URL

Popup window

```
46 <div class="window" <?php echo $windowStyle ?>>
47   <div class="close" <?php echo $closeStyle ?>></div>
48   <div class="pop-content">
49     <?php
50     if (!empty($_GET['popup'])) include("content/".$_GET['popup'].".php");
51     ?>
52   </div>
53 </div>
54 <div id="p-layer" <?php echo $pLayerStyle ?>></div>
55 </body>
```

This is the dark layer, which will fade in when the popup window is opened

## Header Section

Header Section contains site Logo and Theme Switcher panel.

### Logo

To change site **logo** you have to replace the logo.png (located in **images** folder) image to yours own or you could change the source of logo image to your image like this:

```
<a href="index.php" title="My Site">
  
</a>
```

### Theme Switcher

More about Theme Switcher in chapter 9

### Menu Section

Menu HTML content located in the **includes/menu.php** file:

```
<div class="toHome" onclick="goHome()"></div>
<ul class="menu">
<li><span class="link-bg"></span><a href="index.php?content=about">About</a></li>
<li><span class="link-bg"></span><a href="index.php?content=features">Features</a></li>
<li><span class="link-bg"></span><a href="index.php?content=contacts">Contacts</a></li>
<li><span class="link-bg"></span><a href="index.php?content=lorem">Lorem</a></li>
<li><span class="link-bg"></span><a href="index.php?content=ipsum">Ipsum</a></li>
</ul>
```

Back to home link. Use CSS rules to style it

Empty **span** element with a **"link-bg"** class is used here to divide background image for menu link and text link.

To create your own menu link, first of all you have to create list element inside the **ul** list with a “menu” class using the `<li>` tag. Then add an empty span element with a “link-bg” class. After that put the link title inside using `<a>` and `</a>` tags.

Then you have to create new PHP file inside the **content/** folder with a content of new page. Name it for example – **mypage.php**. So all you have to do is to specify link to this file in the created link with a HREF attribute. But you should use the following rule: **You have to specify dynamic path to this file, not direct.**

So for the file above you have to specify the following HREF attribute:

```
<li><span class="link-bg"></span><a href="index.php?content=mypage">My Page</a></li>
```

Don't need to write it like this: `href="content/mypage.php"` Use it like in example above. JavaScript automatically will include content from the **PHP** file which is located in the **content/** folder with filename specified in **content** variable (in this case is the **mypage**)

We need to use such formatting to make these pages visible for search engines and users with disabled JavaScript.

## Features Section

HTML content of the home page features section is located in the **includes/features.php** file:

```
<div class="feature current">
    <!-- First Feature HTML content -->
</div>
<div class="feature">
    <!-- Second Feature HTML content -->
</div>
<div class="feature">
    <!-- Third Feature HTML content -->
</div>
<div class="feature">
    <!-- Fourth Feature HTML content -->
</div>
<div class="feature">
    <!-- Fifth Feature HTML content -->
</div>
<div class="feature">
    <!-- Sixth Feature HTML content -->
</div>
<div class="switches">
    <span class="current">S</span>
    <span>Y</span>
    <span>M</span>
    <span>P</span>
    <span>L</span>
    <span>Y</span>
</div>
```

“current” class must be added to the first feature block

Any HTML content could be here

Index number of switch must be same as the index number of feature block which must be controlled with this switch

“current” class must be added to the first switch span

Any HTML content could be inside the switch span element

## Main Content Section

All content pages (files) will be loaded into the div block with a “content” class.

## Popup Window

All content pages (files) will be loaded into the div block with a “pop-content” class.

## Footer Section

Footer section is used for copyrights information of you company or project. It will be automatically added to any loaded page.

```
<!-- Footer Section -->
<div class="footer">
  <p> --- Footer Content --- </p>
</div>
```

Change --- Footer Content --- to something like this: © 2010 My Company. All Rights Reserved

# 5. Typography

---

## Headings

This is the content heading. You should use the following formatting: `<h2 class="content-heading"> ... Some Heading ... </h2>`

## Simply Features

This is the inner heading. You should use the following formatting: `<h2 class="inner-heading"> ... Some Heading ... </h2>`

## Features Section

On the homepage you can find an animated "Features" section which will show to your visitors all advantages, features or services.

This is the inner heading. You should use the following formatting: `<h2 class="inner-heading"> ... Some Heading ... </h2>`

## Ready to use Color Themes

Simply comes with 4 ready to use color themes, which you can change "on the fly". Click the following images to change color theme:



## Popup Window

Check how the popup looks. Open Footer which allows you to add copyright.



## Sidebar

If you are using page WITHOUT sidebar you have to use following formatting in files from **content/** folder:

```
<!-- Page Title -->
<h2 class="content-heading">Page Title</h2>
<!-- Some Page Content -->

<p align="justify">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer a libero at diam consequat vulputate feugiat vel ipsum...</p>

<!-- End of Page Content -->

<!-- End of Page -->
```

But if you want to use sidebar you have to put page content into the DIV block with a “**with-sidebar**” class, like here:

```
<!-- Page Title -->

<h2 class="content-heading">Page Title</h2>
<div class="with-sidebar">
  <!-- Some Page Content -->

  <p align="justify">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer a libero at diam consequat vulputate feugiat vel ipsum...</p>

  <!-- End of Page Content -->
</div>
<!-- Sidebar -->
<div class="sidebar">
  <div class="module">
    <div class="module-inner">
      <h3 class="module-title">Module 1 Title</h3>
      <p align="justify">Duis pharetra semper sapien, in posuere justo euismod id. Curabitur convallis quam et metus cursus eu tempor ligula viverra...</p>
    </div>
  </div>
  <div class="module">
    <div class="module-inner">
      <h3 class="module-title">Module 2 Title </h3>
      <p align="justify">Duis pharetra semper sapien, in posuere justo euismod id. Curabitur convallis quam et metus cursus eu tempor ligula viverra...</p>
    </div>
  </div>
</div>
<!-- End of Sidebar -->

<!-- End of Page -->
```

This page will look like this:

This is the module title. You should use the following formatting: `<h3 class="module-title"> ... Some Heading ... </h3>`

## Page Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer a libero at diam consequat vulputate feugiat vel ipsum. Nunc quis dui ac sapien adipiscing posuere at ac sapien. Etiam vitae convallis urna. Duis pharetra semper sapien, in posuere justo euismod id. Curabitur convallis quam et metus cursus eu tempor ligula viverra. Nulla eros quam, elementum vel dictum vel, rhoncus id nunc. In posuere lobortis consequat. Sed ultrices erat ac lacus sollicitudin quis aliquet libero faucibus. Sed molestie molestie purus non cursus. In in tortor turpis. Vestibulum a velit nunc. Vestibulum eu diam tellus, et aliquam ante. Pellentesque et magna vel mi commodo mollis a id purus. Curabitur elementum vestibulum urna, vel tincidunt risus pretium eu. Etiam tempus, erat vel iaculis scelerisque, turpis metus vestibulum enim, at pellentesque libero eros id odio. Sed aliquet ipsum quis nisl pharetra vitae rutrum nisl porta. Nunc hendrerit scelerisque nisi, a pretium metus consectetur sed. Curabitur hendrerit eros eget metus elementum gravida.

### Module 1 Title

Duis pharetra semper sapien, in posuere justo euismod id. Curabitur convallis quam et metus cursus eu tempor ligula viverra...

### Module 2 Title

Duis pharetra semper sapien, in posuere justo euismod id. Curabitur convallis quam et metus cursus eu tempor ligula viverra...

© 2011 Simply Premium Template by iDangerous.us. All Rights Reserved.

## 6. Inner Linking (How it works?)

So let's see how the Inner Linking works. This you should to know first:

*All files with HTML content you want to load with Ajax must be located in the **content/** folder.*

*All files must have a **.php** extension.*

**Example.** We want to load HTML content from the **content/about.php** file directly from the simple link in the text. We should use the following formatting:

```
<a href="index.php?content=about" class="goTo">About My Site</a>
```

Filename of the **php** file from the **content/** folder specified as a value of the **content** variable.

When we'll specify an "goTo" class, the script will load file content with Ajax into the content section

As you can see it is very easy to use Inner Linking. All you need is to specify filename as a value of the content variable using the **href** attribute and to specify the "**goTo**" class to load page with Ajax.

## 7. CSS Files and Structure

---

There is one CSS file used in this Template. You can find css attachment in the HEAD section of index.php file.

```
<head>
-----
<link rel="stylesheet" type="text/css" href="css/simply.css" />
-----
</head>
```

By default it is located in the **css/** folder. You can easily change the path or filename in the **href** attribute of **link** tag.

**simply.css** file contains all general layouts for the template. It is separated into sections:

```
/* --- Common Rules --- */
```

*There are common rules in this section, such as font size, fonts, link colors, layout images etc.*

```
/* --- Heading --- */
```

*Rules for headings.*

```
/* --- Header--- */
```

*This section contains style rules for header*

```
/* --- Theme Switcher --- */
```

*Rules for Theme Switcher*

```
/* --- Menu --- */
```

*Rules for menu and navigation buttons*

```
/* --- Content--- */
```

*All styling rules for content pages*

```
/* --- Features--- */
```

*Features Section styling*

```
/* --- Sidebar--- */
```

*Sidebar styling*

```
/* --- Contacts--- */
```

*Forms styling*

```
/* --- Footer --- */
```

*All rules for the Footer*

```
/*----- Popup Window-----*/
```

*Styling rules for the popup window*

```
/*----- Ajax Loader-----*/
```

*Ajax Loader styling*

```
/*-----End of CSS File-----*/
```

If you would like to edit a specific section of the site, simply find the appropriate section in the CSS file, and then scroll down until you find the appropriate style that needs to be edited.

## 8. JavaScripts

Simply Template uses three JavaScript files

Attachment of JavaScript files can be found in the HEAD section of index.php file:

```
<head>
-----
<script type="text/javascript" src="scripts/jquery-1.4.3.min.js"></script>
<script type="text/javascript" src="scripts/simply.js"></script>
<script type="text/javascript" src="scripts/themes.js"></script>
-----
</head>
```

You can easily change the path or filename in the **src** attribute of **script** tag.

The first script file is the jQuery (JavaScript library that greatly reduces the amount of code that you must write).

The second script file (simply.js) is the custom script used for animation, effects and Ajax requests. You can learn more about jQuery methods used here (`fadeOut()`, `fadeIn()`, `fadeTo()`, `animate()`, `jQuery.get()`, `html()`, `delay()`, `hover()` etc.) by visiting this [jQuery Documentation](#) Site.

And the third script file (themes.js) is the JavaScript file which is used for Theme Switcher.

First of all let's look at the custom script more closely.

The first part of the script is executed when the document is loaded. Here are the effects and rules for handling menu links, its animation, Ajax loader, Content pages initialization, function used for popup window.

```
1 // JavaScript Document
2 var getContent;
3 var isIE6 = navigator.appVersion.indexOf("MSIE 6")>0;
4 jQuery(document).ready(function() {
5     /*-----Menu-----*/
6     if(getContent) {
7         for (var i=0;i<=$("#menu li:last").index();i++) {
8             var menuLink = $("#menu li:eq("+i+")")
9             menuLink.children(".link-bg").fadeTo(0,0.2)
10            menuLink.css({"left":(menuLink.width()+10)*i,"top":0})
11        }
12        $("#toHome").show()
13    }
14    else {
15        for (i=0;i<=$("#menu li:last").index();i++) {
16            var menuLink = $("#menu li:eq("+i+")")
17            menuLink.children(".link-bg").fadeTo(0,0.2)
18            menuLink.css({"top":(menuLink.height()+20)*i+40,"left":0})
19        }
20    }
21 }
```

isIE6 – variable will return true if the user browser is Internet Explorer 6.  
We will need it for popup window.

If the **getContent** variable is defined. It means that the site loaded with a **content** variable in URL. So we have to place menu links above the content section.

Otherwise, site is opened on the home page, and the menu links will be placed from the left of Features section.

```

21  $( "ul.menu li" ).hover(
22      function() {
23          if (!$ (this).hasClass('active-link')) {
24              $(this).children(".link-bg").fadeTo(300,1)
25              $(".menu-decor").css({"left":$(this).offset().left-$(".menu-decor").width(),
26              "top":$(this).offset().top-($(".menu-decor").height()-$(this).height())/2}).fadeIn(600)
27          },
28          function() {
29              if (!$ (this).hasClass('active-link')) {
30                  $(this).children(".link-bg").fadeTo(300,0.2)
31              }
32              $(".menu-decor").stop(true,true).hide()
33          }
34      )
35  }
36  $("#menu a").click(function(e) {
37      e.preventDefault();
38      var menuLink = $(this).parent("li");
39      if(!menuLink.hasClass("active-link")) {
40          $(".menu .active-link .link-bg").fadeTo(300,0.2)
41          $(".menu .active-link").removeClass('active-link')
42          menuLink.addClass('active-link');
43          menuLink.children(".link-bg").fadeTo(300,1);
44          var url = $(this).attr('href').split('content=');
45          goTo(url[1]);
46      }
47  })
48  /*-----Theme Switcher-----*/
49  $(".theme-switcher").hover(
50      function() {
51          $(this).animate({"top":0},300)
52      },
53      function() {
54          $(this).animate({"top":-60},300)
55      }
56  )
57  /*-----Features-----*/
58  $(".switches span").click(function() {
59      if (!$ (this).hasClass("current")) {
60          var currentIndex = $(".switches .current").index()
61          $(".switches .current").animate({"margin-left":0},300,function() {
62              $(".switches span:eq("+currentIndex+")").removeClass("current")
63          });
64          $(this).animate({"margin-left":10},300,function() {
65              $(this).addClass("current")
66          })
67          $(".feature:eq("+currentIndex+")").fadeOut(300,function() {
68              $(".feature:eq("+currentIndex+")").removeClass("current")
69          })
70          $(".feature:eq("+$(this).index()+")").delay(300).fadeIn(300,function() {
71              $(".feature:eq("+$(this).index()+")").addClass("current")
72          })
73      }
74  })
75  /*-----Ajax Loader-----*/
76  $(".ajax-loader").ajaxStart(function() {
77      $(this).css({"position":"absolute","top":$(window).scrollTop()+250}).show()
78  })
79  $(".ajax-loader").ajaxComplete(function() {
80      $(this).hide();
81  })
82  /*-----Popup Window-----*/
83  if ($('window').css("display")=="block") {
84      $('window').css({"left":($(window).width()-$('window').width())/2})

```

Fade effect for the menu links. It means that when you take mouse cursor on this link its background (span with a **link-bg** class) will fade to opacity 1 for 0.3 seconds. And "menu-decor" block will fade in for 0.6 seconds.

And when you take mouse cursor away from link its background (span with a **link-bg** class) will fade to opacity 0.2 for 0.3 seconds. And "menu-decor" block will disappear.

Handling for menu links. When you'll click on it, first of all, function will check if this link is active or not. If not, then the "active-link" will fade to opacity 0.2 for 300ms. And the "active-link" class will be removed from it. After that the "active-link" class will be added to clicked link, and it will fade in for 300ms.

And after the **goTo()** function will be executed with the attribute equal to the value of the "content" variable from HREF attribute of link.

Animation of the Theme Switcher icons. First function will be executed when you take mouse cursor on it. And the second one is used when you take mouse cursor away from icon.

Features section animation. When we'll click on the "switch", and if it has not "current" class, we have to get the index number of active slide (it will be the same as an index number of active switch).

After that we have to set active switch to its "normal" position, and after that to remove "current" class from it.

Then we have to animate clicked switch to margin-left=10px for 300ms, and to add "current" class to it.

Then we have to fade out active "feature" block for 300ms, and after that to remove "current" class from it

We need to fade in new "feature" block (with index number such as a clicked switch index number) for 300ms after the 300ms delay, and to add the "current" class to it

This code means that when the Ajax request is starting, the Ajax Loader will be shown and its top position will be equal to 250px

When the Ajax request is completed, the Ajax Loader will be hidden

Positioning of the **popup** window.

```

85  $("#p-layer").click(function() {
86      closePopup();
87  });
88  $("#a.popup").click(
89  function(e) {
90      e.preventDefault();
91      var width=$(this).attr("rel");
92      var height=$(this).attr("rev");
93      if($(this).hasClass("direct")) var uri = $(this).attr("href");
94      else {
95          var uri = "content/" + $(this).attr("href").split("popup=")[1] + ".php"
96      }
97      popUp(uri,width,height)
98  })
99  /*-----Ajax Content-----*/
100  $("#a.goTo").click(
101  function(e) {
102      e.preventDefault();
103      var url = $(this).attr("href").split("content=")[1]
104      goTo(url)
105  })
106  })

```

When we'll click the dark layer, the popup window will be closed

Handling for links with a "popup" class. Width and height variables will be equal to the REL and REV attributes. If link has the "direct" class then the URL will be equal to the value of HREF attribute, otherwise it will be equal to the value of the "content" variable from HREF attribute of link. And the popUp() function will be executed.

Handling for links with a "popup" class. If it is clicked, then the goTo() function will be executed with a URL equal to the value of the "content" variable from HREF attribute of link

The next part of code contains functions which could be called from any place of the document. These functions are used to load files with content, for submitting and validation of the contact form, to call a popup window etc:

```

107 function goTo(url) {
108     if ($('#features').css("display")!="none") {
109         jQuery.get("content/"+url+".php",function(newContent){
110             $('#features').fadeOut(600,function(){
111                 for (i=0;i<=$("#menu li:last").index();i++) {
112                     var menuLink = $("#menu li:eq("+i+")")
113                     menuLink.delay(200*(i-1)).animate({"left":(menuLink.width()+10)*i,"top":
0},300)
114                 }
115                 $(".content").html(newContent)
116                 pageScript();
117                 $(".content-wrap").delay(600).fadeIn(600,function(){
118                     $(".toHome").fadeIn(600)
119                 })
120             })
121         })
122     }
123     else {
124         jQuery.get("content/"+url+".php",function(newContent){
125             $(".content-wrap").fadeOut(600,function(){
126                 $(".content").html(newContent)
127                 pageScript()
128                 $(".content-wrap").delay(300).fadeIn(600);
129             })
130         })
131     }
132 }
133 }
134 }

```

goTo() function is used to load php files from content/ folder and put their HTML content to site content section.

This case is when we are loading new page from home page where the features section is visible.

Ajax GET request to the specified php file from the content/ folder

Features section will fade out for 600ms and after that every menu link will change its position with animation

HTML content will be inserted into the div with a "content" class, and the pageScript() function will be executed.

"content-wrap" block will fade in for 600ms after the 600ms delay.

"toHome" link will fade in for 600ms

This case used when we are loading new page from another page

Ajax GET request to the specified php file from the content/ folder

"content-wrap" block will fade out for 600ms to hide "active" page. After that new HTML content will be inserted into the div with a "content" class, and the pageScript() function will be executed. "content-wrap" block will fade in for 600ms after the 300ms delay.



This function will be executed with every page load.

Handling for links with a "popup" class.

```
135 function pageScript() {
136     $("a.popup").click(
137         function(e) {
138             e.preventDefault();
139             var width=$(this).attr("rel");
140             var height=$(this).attr("rev");
141             if($(this).hasClass("direct")) var uri = $(this).attr("href");
142             else {
143                 var uri = "content/"+$(this).attr("href").split("popup=")[1]+".php"
144             }
145             popUp(uri,width,height)
146         })
147     $("a.goTo").click(
148         function(e) {
149             e.preventDefault();
150             var url = $(this).attr("href").split("content=")[1]
151             goTo(url)
152         })
153 }
```

Handling for links with a "goTo" class.

This function provides an opportunity to hide "active" page, place menu links from the left of Features section, to show features section with fade animation, and to hide goHome button.

```
154 function goHome() {
155     $(".active-link").children('.link-bg').fadeOut(300,0.2);
156     $(".active-link").removeClass('active-link')
157     $(".content-wrap").fadeOut(600,function(){
158         for (i=0;i<= $(".menu li:last").index();i++) {
159             var menuLink = $(".menu li:eq("+i+")")
160             menuLink.delay(200*(i-1)).animate({"top":(menuLink.height()+20)*i+40,"left":0},
300)
161         }
162         $("#features").delay(900).fadeIn(600);
163     })
164     $(".toHome").fadeOut(300)
165 }
```

popUp() function provides an opportunity to call a Popup Window. First of all function calculates window size to set it in the center of screen. After that it will be shown with slide animation for 0.6 seconds. And after, the dark layer will be faded in for 0.3 seconds to opacity 0.6

```
166 /*-----Popup Windows-----
167 function popUp(uri,width,height) {
168     var width=width||600;
169     var height=height||"auto";
170     var windowTop = $(window).scrollTop() + 100;
171     jQuery.get(uri,function(data) {
172         $(".pop-content").html(data);
173         $(".window").css({"width":width,"height":height})
174         $(".window").css({"left":($ (window).width()-$(".window").width())/2,"top":
windowTop}).delay(300).fadeIn(300)
175         $(".close").delay(600).fadeIn(200)
176         if(!isIE6) $(".#p-layer").fadeOut(300,0.6)
177         else {
178             $(".#p-layer").css({"position":"absolute","height":$(window).height(),"top":$
(window).scrollTop()}).fadeOut(300,0.6)
179             $(window).scroll(function(){
180                 $(".#p-layer").css({"top":$(window).scrollTop()})
181             })
182         }
183     })
184 }
185 }
186 function closePopup() {
187     $(".close,#p-layer").fadeOut(300)
188     $(".window").fadeOut(300);
189 }
```

Default width value (600px)

Default height value ('auto')

Top position for popup window

Ajax GET request to the specified file. After that new content (from file) will be inserted into the div with a "pop-content" class

If browser is not IE6, then the dark-layer will just fade to opacity 0.6 for 300ms

This case is used for IE6 browser, because it not supports "fixed" position CSS property. And when the window will scroll, the "dark-layer" will follow it.

closePopup() function provides an opportunity to close a Popup Window. Window will be closed using fade animation for 0.3 seconds



```

190 /*-----Validation And Send Message Functions-----*/
191 function validateEmail() {
192     var email = $("input#email").attr("value");
193     var re = /^[^\w+([.-]?\w+)*@(((([a-z0-9]{2,})|([a-z0-9]{-}[a-z0-9]{+}))[\.]?[a-z0-9]{1,3})|([a-z0-9]{1,3}+)[a-z0-9]{1,3})+$/i;
194     if(re.test(email)) $("input#email").removeClass("invalid");
195     else $("input#email").addClass("invalid");
196 }
197 function validateMessage() {
198     message = $("textarea#message").val();
199     m_length = ($.trim(message)).length;
200     if (m_length < 10) $("textarea#message").addClass("invalid");
201     else $("textarea#message").removeClass("invalid");
202 }
203 }
204 function validateText(fieldId,length) {
205     field = $("input#" + fieldId);
206     text = field.val();
207     m_length = ($.trim(text)).length;
208     if (m_length < length) field.addClass("invalid");
209     else field.removeClass("invalid");
210 }
211 function sendMessage() {
212     validateEmail();
213     validateMessage();
214     validateText('name',6);
215     validateText('subject',6);
216     if (!$("textarea#message,input#email,input#subject,input#name").hasClass("invalid")) {
217         name = $("input#name").attr("value");
218         email = $("input#email").attr("value");
219         subject = $("input#subject").attr("value");
220         message= $("textarea#message").val().replace(/<\/?[>]+>/gi, '');
221         jQuery.post("scripts/send_message.php",{name : name, email : email, subject :
subject, message : message},
222             function (status) {
223                 $("#contact-form").fadeOut(300,0,function() {
224                     $(this).css({"visibility":"hidden"})
225                     $(".message_status").html(status).fadeIn(300)
226                 })
227             }
228         );
229     }
230 }

```

**validateEmail()** function is used to validate email field in the contact form. Validation is based on the regular expressions. If the email address is wrong then the "invalid" class will be added to the email field (input field with an "email" id attribute). Otherwise it will be removed.

**validateMessage()** function is used for validating message area in the contact form. If the message contains less than 10 characters, then the "invalid" class will be added to the contact form **textarea** with a "message" id attribute. Otherwise the "invalid" class will be removed.

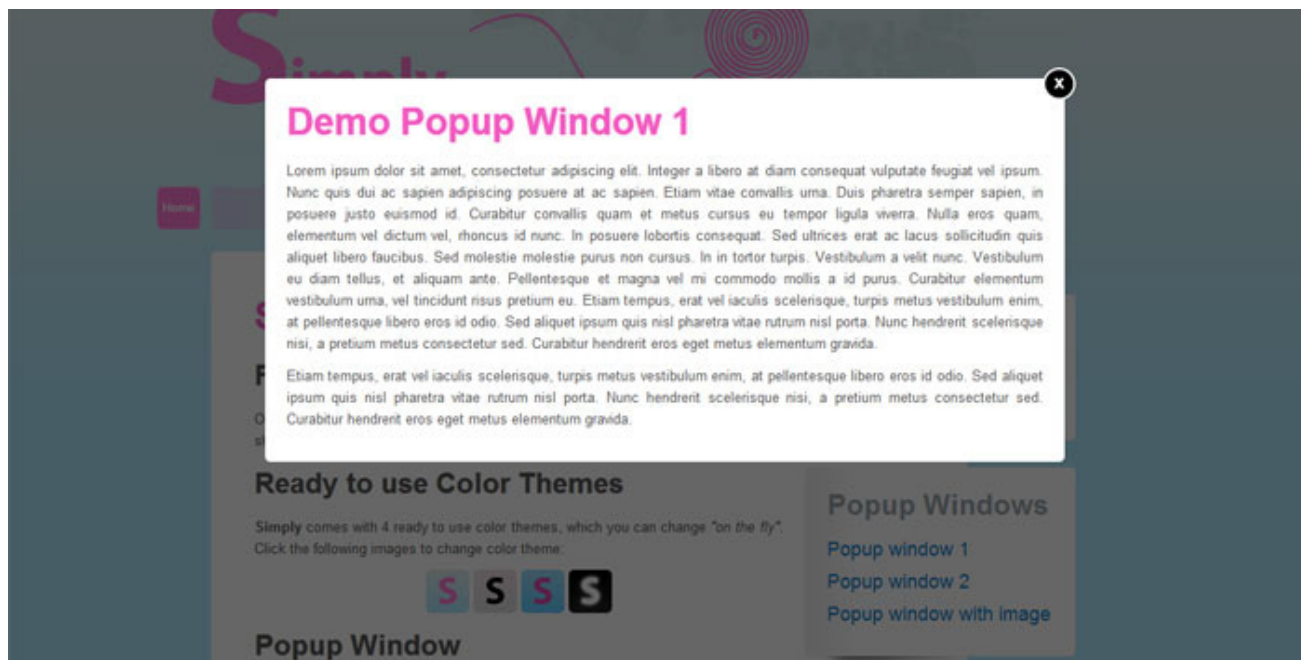
**validateText()** function is used to validating text fields in the contact form. It is used here to validate **Subject** and **Name**. If the field contains less than **length** characters, then the "invalid" class will be added to this field. Otherwise the "invalid" class will be removed.

**sendMessage()** function provides an opportunity to send e-mail message by the Ajax request (without reloading of the site). First of all validate functions will be executed (to validate **Email**, **Name**, **Subject** and **Message**). And then if some of the field has the "invalid" class, message will not be sent.

If all fields are correct, then the function will send an Ajax request to the php script used for sending e-mail messages with the following variables: **name**, **email**, **subject** and **message**. Script will return the message with the mailing status, contact form will fade out and mailing status will fade in instead of form.

## 9. Popup Window

**Simply** template has its own engine to call popup windows. Here is screenshot:



Now let's see how the popup window works and how to call it with a desired content:

Popup Window Layout is located in bottom of the index.php file:

```
46 <div class="window" <?php echo $windowStyle ?>>
47   <div class="close" <?php echo $closeStyle ?>></div>
49   <div class="pop-content">
50     <?php
51     if (!empty($_GET['popup'])) include("content/".$_GET['popup'].".php");
52     ?>
53   </div>
54 <div id="p-layer" <?php echo $pLayerStyle ?>></div>
```

Image link used to close popup window when you'll click it

All content will be placed into this block

Dark layer. It's used to obscure the whole site behind the popup window.

Popup engine allows you to popup HTML content from the file.

**Example 1.** We need to call a popup window with HTML content from the *content/about.php* file. We can use the following formatting for the links:

Use **"popup"** class to call a popup window with HTML content from file

Specify **filename** as a value of the **content** variable in the **href** attribute

Use **rel** attribute to specify window width in px (not required – 600px is default)

And **rev** attribute for window height in px (not required – 'auto' is default)

```
<a class="popup" href="index.php?content=about" rel="800" rev="200">About</a>
```

JavaScript automatically will include content from the **PHP** file which is located in the **content/** folder with filename specified in **content** variable (in this case is the **about**)

**Example 2.** We need to call a popup window with HTML content **NOT** from the **content/** folder file. In these cases additional class "direct" is required! For example we need content from the **myfolder/about.php** file. We should use the following formatting for the links:

Additional class - **"direct"**

Specify **direct** file location in the **href** attribute

Use **rel** attribute to specify window width in px (not required – 600px is default)

And **rev** attribute for window height in px (not required – 'auto' is default)

```
<a class="popup direct" href="myfolder/about.php" rel="800" rev="200">About</a>
```

## 10. Themes

---

Simply comes with 4 ready to use color themes, which you can change "on the fly". Now let's see how to create your own themes or how to change default ones. All information about themes and their configuration contained in the **scripts/themes.js** JavaScript file:

Container for all themes

New container for "grey" theme

New image in grey theme – bgImage (background image)

Location of **background** image

Background color

Location of **header** image

Location of **menu-link** image

Menu links color

Location of **menu-decor** image

Location of **home-link** menu image

Location of **logo** image

Color for heading with "content-heading" class

Color for heading with "module-title" class

Color for links

Color for switches (Features section)

Same code for **blue** theme

Same code for **dark** theme

```
1 // JavaScript Document
2 var theme = new Object();
3 theme.grey = new Object();
4 theme.grey.bgImage = new Image();
5 theme.grey.bgImage.src = 'images/grey/grey-bg.png';
6 theme.grey.bgColor = "#ddd";
7 theme.grey.header = new Image();
8 theme.grey.header.src = 'images/grey/header.jpg';
9 theme.grey.menuImg = new Image();
10 theme.grey.menuImg.src = 'images/grey/menu-link.png';
11 theme.grey.menuLinkColor = "#fff";
12 theme.grey.menuDecor = new Image();
13 theme.grey.menuDecor.src = 'images/grey/menu-decor.png';
14 theme.grey.homeLinkImg = new Image();
15 theme.grey.homeLinkImg.src = 'images/grey/home.png';
16 theme.grey.logo = new Image();
17 theme.grey.logo.src = 'images/grey/logo.png';
18 theme.grey.contentHeading = "#555";
19 theme.grey.moduleTitle = "#777";
20 theme.grey.linkColor = "#721555";
21 theme.grey.switchColor = "#000";
22
23 theme.blue = new Object();
24 theme.blue.bgImage = new Image();
25 theme.blue.bgImage.src = 'images/blue/blue-bg.png';
26 theme.blue.bgColor = "#64ceff";
27 theme.blue.header = new Image();
28 theme.blue.header.src = 'images/blue/header.jpg';
29 theme.blue.menuImg = new Image();
30 theme.blue.menuImg.src = 'images/blue/menu-link.png';
31 theme.blue.menuLinkColor = "#fff";
32 theme.blue.menuDecor = new Image();
33 theme.blue.menuDecor.src = 'images/blue/menu-decor.png';
34 theme.blue.homeLinkImg = new Image();
35 theme.blue.homeLinkImg.src = 'images/blue/home.png';
36 theme.blue.logo = new Image();
37 theme.blue.logo.src = 'images/blue/logo.png';
38 theme.blue.contentHeading = "#F25BC2";
39 theme.blue.moduleTitle = "#FFA1E1";
40 theme.blue.linkColor = "#2d9bed";
41 theme.blue.switchColor = "#0093ff";
42
43 theme.dark = new Object();
44 theme.dark.bgImage = new Image();
45 theme.dark.bgImage.src = 'images/dark/dark-bg.png';
46 theme.dark.bgColor = "#222";
47 theme.dark.header = new Image();
48 theme.dark.header.src = 'images/dark/header.jpg';
49 theme.dark.menuImg = new Image();
50 theme.dark.menuImg.src = 'images/dark/menu-link.png';
51 theme.dark.menuLinkColor = "#555";
52 theme.dark.menuDecor = new Image();
53 theme.dark.menuDecor.src = 'images/dark/menu-decor.png';
54 theme.dark.homeLinkImg = new Image();
55 theme.dark.homeLinkImg.src = 'images/dark/home.png';
56 theme.dark.logo = new Image();
57 theme.dark.logo.src = 'images/dark/logo.png';
58 theme.dark.contentHeading = "#333";
59 theme.dark.moduleTitle = "#555";
60 theme.dark.linkColor = "#111";
61 theme.dark.switchColor = "#000";
```

```

63 jQuery(document).ready(function() {
64     var menuDecor = new Image()
65     menuDecor.src = getBgSrc(".menu-decor")

```

We have to preload this **menu-décor** image when site is loaded

```

66
67     theme.def = new Object();
68     theme.def.bgImage = new Image();
69     theme.def.bgImage.src = getBgSrc("body");
70     theme.def.bgColor = $("body").css("background-color");
71     theme.def.header = new Image();
72     theme.def.header.src = getBgSrc("#header");
73     theme.def.menuImg = new Image();
74     theme.def.menuImg.src = getBgSrc(".link-bg");
75     theme.def.menuLinkColor = $("#menu a").css("color");
76     theme.def.menuDecor = new Image();
77     theme.def.menuDecor.src = getBgSrc('.menu-decor');
78     theme.def.homeLinkImg = new Image();
79     theme.def.homeLinkImg.src = getBgSrc(".toHome");
80     theme.def.logo = new Image();
81     theme.def.logo.src = $("#logo").attr("src");
82     theme.def.contentHeading = "#F25BC2";
83     theme.def.moduleTitle = "#9CA7AF";
84     theme.def.linkColor = "#1a7ec8";
85     theme.def.switchColor = "#FF7BD5";

```

Same code for **default** theme

```

86 })
87 function getBgSrc(selector) {
88     var bgSrc = $(selector).css('background-image').split('');
89     if(bgSrc[1]) return bgSrc[1];
90     else {
91         var bgSrc = $(selector).css('background-image').substr(4)
92         bgSrc = bgSrc.substr(0, bgSrc.length-1)
93         return bgSrc;
94     }
95 }

```

**getBgSrc()** function is used to return source (location) of background image for any element specified in CSS file. As an attribute it accepts css selector.

**setTheme()** function is used to change the Theme. As an attribute it accepts name of some theme defined above.

```

96 function setTheme(themeColor) {
97     if(isIE6) {

```

This case is for IE6 browser. It is used to fix "**white-layer**" position. The same as for the **dark-layer** in the **popUp()** function

```

98         $(".white-layer").css({"position":"absolute", "top":$(window).scrollTop(), "left":0,
99         "width":"100%", "height":$(window).height() })
100     $(".white-layer").delay(200).fadeIn(600, function() {
101         $(".body").css({'background-color':theme[themeColor]['bgColor'], 'background-image':
102         'url('+theme[themeColor].bgImage.src+')'})
103         $(".header").css({'background-image':'url('+theme[themeColor].header.src+')'})
104         $(".link-bg").css({'background-image':'url('+theme[themeColor].menuImg.src+')'})
105         $(".toHome").css({'background-image':'url('+theme[themeColor].homeLinkImg.src+')'})
106         $(".logo").attr({"src":theme[themeColor].logo.src})
107         $(".menu a").css({"color":theme[themeColor].menuLinkColor})
108         $(".menu-decor").css({'background-image':'url('+theme[themeColor].menuDecor.src+')',
109         'width':theme[themeColor].menuDecor.width, 'height':theme[themeColor].menuDecor.height})
110         var CSSStyleSheet = document.styleSheets[0];

```

First of all the white layer will fade in for 600ms after the 200ms delay. And after, all BG images will be applied to the according elements

```

109         if(CSSStyleSheet.insertRule){
110             var lastRule = CSSStyleSheet.cssRules.length;
111             CSSStyleSheet.insertRule("h2.content-heading {color: "+theme[themeColor].
112             contentHeading+"}", lastRule);
113             CSSStyleSheet.insertRule(".module-title {color: "+theme[themeColor].moduleTitle+
114             "}", lastRule);
115             CSSStyleSheet.insertRule("a {color: "+theme[themeColor].linkColor+"}", lastRule)
116             ;
117             CSSStyleSheet.insertRule(".switches span.current {color: "+theme[themeColor].
118             switchColor+"}", lastRule);
119         }

```

This case is for all browsers excepting Internet Explorer



This case is for Internet Explorer browser

```
116     else {
117         var lastRule = CSSStyleSheet.rules.length;
118         CSSStyleSheet.addRule("h2.content-heading", 'color: '+theme[themeColor].
contentHeading);
119         CSSStyleSheet.addRule(".module-title", 'color: '+theme[themeColor].moduleTitle);
120         CSSStyleSheet.addRule("a", 'color: '+theme[themeColor].linkColor);
121         CSSStyleSheet.addRule(".switches span.current", 'color: '+theme[themeColor].
switchColor);
122     }
123     }).delay(200).fadeOut(600);
124 }
```

And after all rules are applied, the white layer will fade out for 600ms after the 200ms delay

To change color for links (headings, body background etc), all those “color” rules will be added right into the first linked CSS file (but not real)

As you can see it's not hard to add your own color theme or to change existing one. All you need is to use theme formatting like in code above.

## 11. Ajax Contact Form

HTML code of the contact page is in the **content/contascts.php** file:

Contact information here is formatted with the **hCard** microformat. So fill it up according to your company details

```
1 <h2 class="content-heading">Contacts</h2>
2 <div>
3     <div class="vcard"> <a class="fn org url" href="http://www.idangero.us/">Simply Ltd.</a>
4         <div class="adr">
5             <div class="street-address">114 Second Lane Street</div>
6             <span class="locality">Rostov-na-Donu</span>, <span class="postal-code">344013</span>
7             <div class="country-name">Russian Federation</div>
8         </div>
9         <div class="tel"> <span class="type">Work</span> +7-863-246-3608 </div>
10        <div class="tel"> <span class="type">Fax</span> +7-863-246-3608 </div>
11        <div>Email: <span class="email">info@idangero.us</span> </div>
12    </div>
13 </div>
14 <div class="message_status"> </div>
15 <form id="contact-form" action="javascript:sendMessage()" method="post">
16     <p><strong>You can contact us using the following contact form:</strong></p>
17     <p>
18         <label>Your Name: *<br />
19         <input type="text" size="50" name="name" id="name" onblur="validateText('name',6)" />
20     </label>
21 </p>
22 <p>
23     <label>Subject: *<br />
24     <input type="text" size="50" name="subject" onblur="validateText('subject',6)" id="subject" />
25 </label>
26 </p>
27 <p>
28     <label>E-mail: *<br />
29     <input type="text" size="50" name="email" onblur="validateEmail()" id="email" />
30 </label>
31 </p>
```

```

32 <p>
33 <label>Message: *<br />
34 <textarea id="message" onblur="validateMessage()" cols="30" rows="10"></textarea>
35 </label>
36 </p>
37 <p><em>Fields marked with an asterisk(*) are required</em></p>
38 <p>
39 <input type="submit" value="Send Message" />
40 </p>
41 </form>

```

If the all fields are correct sendMessage() script will send and Ajax request to the php mailing script located in the **includes/send\_message.php** file. This script will send a message and will return mailing status message. Let's look at this script more closely:

```

1 <?php
2 if (isset($_POST['name'])) $name = $_POST['name'];
3 if (isset($_POST['email'])) $email = $_POST['email'];
4 if (isset($_POST['subject'])) $subject = $_POST['subject'];
5 if (isset($_POST['message'])) $message = $_POST['message'];
6
7 $mail_to = "demo@idangero.us"; //change this email to yours
8
9 $message = stripslashes($message);
10 $headers = "From: ".$name." <".$email.">\r\n";
11 $headers .= "Reply-To: ".$email."\r\n";
12 $headers .= "X-Mailer: PHPMailer".".\r\n";
13 $headers .= "Content-Type: text/plain; charset=\\"utf-8\\"";
14 if (mail($mail_to, $subject, $message, $headers)) {
15     ?>
16
17 <div class="mail-sent">
18     <h4><img style="margin-right:10px" align="left" src=
19 "images/layouts/mail-sent.png" alt="Ok" />Thanks! Your message has been
20 successfully sent!</h4>
21 </div>
22 <?php
23 }
24 else {
25     ?>
26 <div class="mail-error">
27     <h4><img style="margin-right:10px" align="left" src=
28 "images/layouts/mail-error.png" alt="Error" />Error Occured. Try again later.
29 </h4>
30 </div>
31 <?php
32 }
33 ?>

```

Checking for existence of these variables.

Message will be sent to this e-mail address. So you have to change it to yours

Standard headers for the e-mail message.

If the message was successfully sent, the script will return this HTML message.

Otherwise script will return this HTML code with an error message.

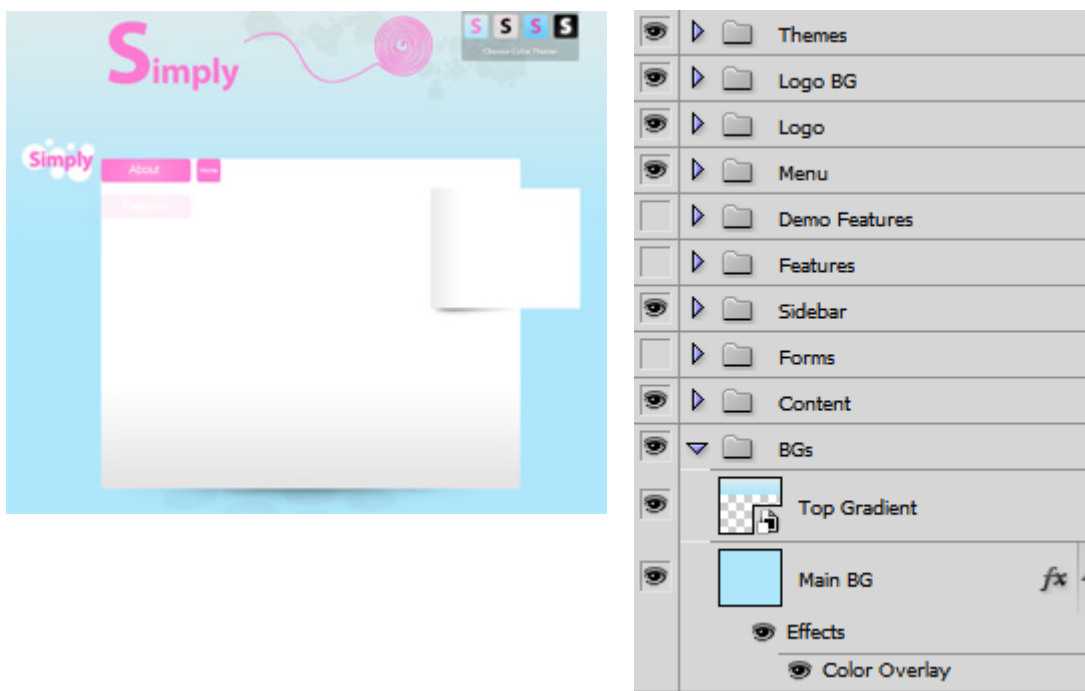
You can easily edit this status messages using the any HTML formatting.

## 12. PSD Files & Structure

---

There are the **five** layered **.psd** files comes with the **Simply** template. You can locate them into the **psd/** folder from the downloaded archive. Using these files you can easily redesign template. To edit these files you'll need an Adobe™ Photoshop™ (Recommended is CS4 or CS5 version)

First **.psd** file is the **simply.psd**. It contains all general layouts for template styling:



Also there are three **psd** files, which contain design of three other themes:

- **simply-grey.psd**
- **simply-dark.psd**
- **simply-blue.psd**

And the final **psd** file is the **ajax-loader.psd**, which will help you to redesign Ajax loader.

All layers in these file are well organized and easy to understand and redesign.



# 13. Licensing Terms

---

## Regular License (RL)

RL gives you as a customer non-exclusive & non-transferable right to use the product you've bought, in this case is the **Simply** (further "**Item**")

1. We do not limit the number of **Item**'s copies you are going to use. Using one **Item** you can create, for example, even 100 web-sites.
2. You can use the **Item** by itself or it's also possible to apply it in other project you work at.
3. You can use the **Item** for your own purpose as well as for your clients.
4. You can use the **Item** in commercial projects.
5. You can reproduce the **Item**:
  - on a web-site or as a web-site
  - as a part of software you create not for sale
  - as a printed variant
  - in digital(electronic) format (as a presentation or an e-book)
  - in video products
6. All photos used on iDangero.us for demonstration photo gallery functioning are our private property (if only there is nothing said about contrary). It's prohibited to use these photos wherever.
7. You are not allowed to sell, resell, license or give the **Item** free (any way) without our written consent. Please, do not offer to do it to any person.
8. You also do not have the right to use the **Item** in a project for selling (for example, for other templates, scripts, graphics and so on).
9. It's prohibited to rework / redesign / reproduce the **Item** (i.e.to rename it or change graphics & so on) and after this to sell it as your own.
10. In spite of reselling limitation you could claim money for the **Item** from your client.

11.If the **Item** (the whole **Item** or its parts) is created with materials used by GNU General Public License (GPL) (or some other license) it means you should follow all the terms of the license using the **Item**.