



PREMIUM SITE TEMPLATES



“Device”

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 “Device”
2. Device Files & Folders Structure
3. Installation
4. HTML Structure
5. Window Layout Templates
6. Typography
7. Work With Windows
8. CSS Files and Structure
9. PSD Files and Structure
10. Customization
11. JavaScripts
12. Ajax Contact Form
13. Licensing Terms

1. About “Device”

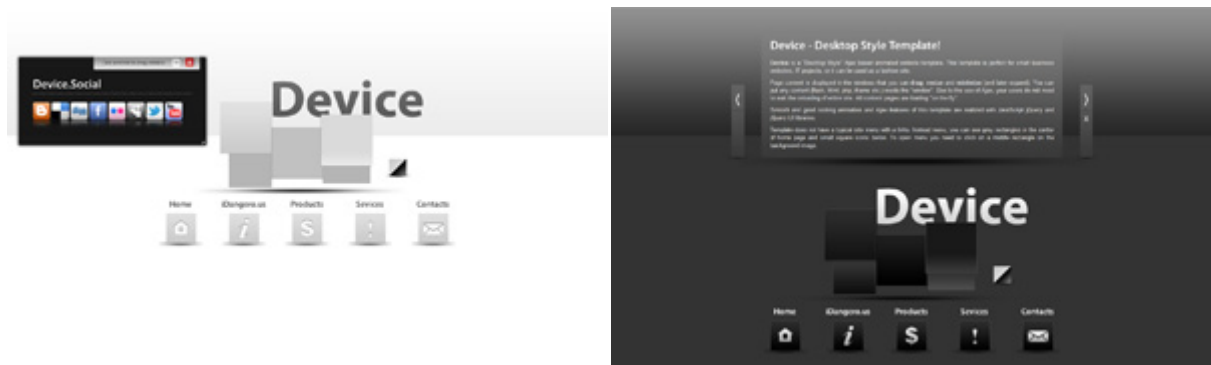
Device is a "Desktop Style" Ajax based animated website template, designed by the iDangero.us team. This template is perfect for small business websites, IT projects, or it can be used as a fashion site.

Page content is displayed in the windows that you can drag, resize and minimize (and later expand). You can put any content (flash, html, php, iframe etc.) inside the "window". Due to the use of Ajax, your users do not need to wait the reloading of entire site. All content pages are loading "on the fly".

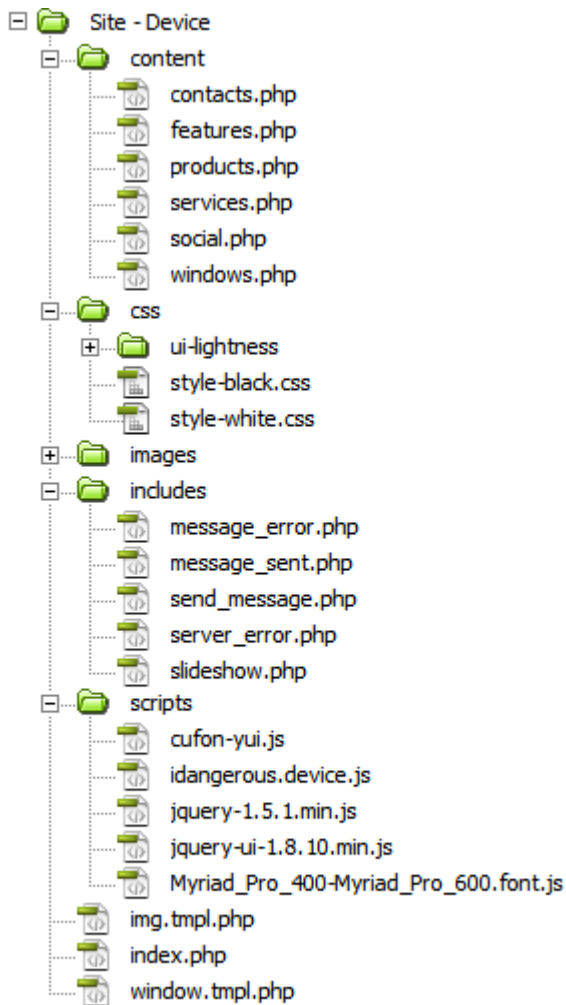
Smooth and good looking animation and Ajax features of this template are realized with JavaScript jQuery and jQuery UI libraries. This template is easy to redesign, it has very simple and clear API.

Also **Device** is powered with an Ajax Contact Form with validation.

Here is a couple of screenshots:



2. Device Files Structure



Device files are located in the **device/** 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: php files with slideshow and scripts used for Ajax contact form.

scripts: JavaScripts used in this template

img.tmpl.php and **window.tmpl.php** in root folder files are window layout templates (for image and content)

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

3. Installation

To install **Device** template you have to copy all files & folders from the **device/** 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 Device Template. Open **index.php** file with a text editor (Highly recommended to use Adobe® Dreamweaver® or Notepad ++ editors).

Index.php <body> structure:

"white" class is used for color theme changing

Ajax loader container

"wrapper" div contains Slideshow, Menu and Control Panel sections

Slideshow container

slide-left and slide-right divs contain arrow images to slide Slideshow

Slideshow HTML content located in the includes/slideshow.php file

This is the "Control Panel" (CP) – grey rectangles with links in the center of homepage

Image with a "change-color" class is used to change color theme, when we click on it

```
18 <body class="white">
19 <div class="ajax-loader">
20      Loading...
22 </div>
23 <div id="wrapper">
24     <div id="slideshow">
25         <div class="slide-left">
26             
28         </div>
29         <div class="slide-right">
30             <br/>
32         <div class="close-ss" src="images/layouts-white/close-ss.png" width="13" height="13" alt=
33         "close" title="Close SlideShow" />
34     </div>
35     <div class="slideshow-wrap">
36         <div class="slideshow-wrap2">
37             <?php include("includes/slideshow.php") ?>
38         </div>
39     </div>
40     <div id="cp">
41         <a class="left-top new-window" title="Device Features" href="content/features.php" rel=
42         "800">
43             
44         </a>
45         <a class="left-bot openSS" href="#" title="Open Slideshow">
46             
47         </a>
48         <a class="middle openMenu" href="#" title="Open Menu">
49             
50         </a>
51         <a class="right-top new-window" href="content/contacts.php" title="Contacts">
52             
53         </a>
54         <a class="right-bot new-window" rel="450" href="content/social.php" title="Device Social">
55             
56         </a>
57         
59     </div>
```

```

55 <div id="menu">
56   <a href="index.php" class="menu-link">
57     <span class="cufoned">Home</span>
58     
59   </a>
60   <a href="http://www.idangero.us" class="menu-link new-window iframe cufon" rel="900">
61     <span class="cufoned">iDangero.us</span>
62     
63   </a>
64   <a href="content/products.php" class="menu-link new-window">
65     <span class="cufoned">Products</span>
66     
67   </a>
68   <a href="content/services.php" rel="700" class="menu-link new-window">
69     <span class="cufoned">Services</span>
70     
71   </a>
72   <a href="content/contacts.php" class="menu-link new-window">
73     <span class="cufoned">Contacts</span>
74     
75   </a>
76 </div>
77 </div>
78 <div id="footer"> &copy; 2010 "Device" Premium Template by iDangero.us. All Rights
Reserved </div>
79 <div id="windows"></div>
80 <div id="white-layer"></div>
81 </body>

```

Menu container.

Menu link with its title and thumbnail image

Footer Section.

All windows will be added into this div with a "windows" id attribute

This is the white layer, which will fade in on the theme color change

Slideshow Section

As mention above Slideshow content is located in the **includes/slideshow.php** file. Let's open this file and look at the HTML code of it:

```

<div class="slide withPadding withBG">
  <h3 class="cufoned slide-heading">Device - Desktop Style Template!</h3>
  ----- First slide HTML content -----
</div>
<div class="slide withPadding">
  <h3 class="cufoned slide-heading">Three types of window layouts:</h3>
  ----- Second slide HTML content -----
</div>
<div class="slide">
  ----- Third slide HTML content -----
</div>
<div class="slide">
  ----- Fourth slide with just image -----
  
</div>
<div class="slide withBG withPadding">
  ----- Fifth slide HTML content -----
</div>

```

First slide container with additional typographic classes

First slide heading

Third slide without additional classes and with custom HTML content inside

Fourth slide contains only image

Fifth slide with additional classes and with custom HTML content inside

As you can see it is very easy to create your own (or to change existing one) slides. Every slide must be enclosed inside of div with a “slide” class. Let’s look at the additional classes used for slides and their function:

- “withPadding”. Use this class to add 20px padding with every side of slide
- “withBG”. Use this class to add gradient background image for slide

Remember every slide is 700px in width and 300px in height. But if you are using “withBG” class slide will be 660px in width and 260px in height.

You can also use the inline slide heading using the following formatting:

```
<h3 class="slide-heading">Device - Desktop Style Template!</h3>
```



Additional “slide-heading” class for h3 element

Read more about “**cufoned**” class in the “Typography” chapter.

Control Panel (CP)

As mentioned above Control Panel content is located inside of div with “cp” id attribute. It contains 5 links and 1 image used to change color theme. Every link contains its image which hidden by default. Every link has a position class such as “left-top”, “left-bot” etc, they are used to set link position and size according to the background image with rectangles.

As you can see every link also has one of the following classes:

- “new-window” class is used to open window with content. More about its usage in chapter 7 (Work With Windows)
- “openSS” class is used to open Slideshow section
- “openMenu” class is used to open Menu section

Menu Section

Menu HTML content located in the **div** element with a “menu” id attribute:

```
<div id="menu">
  <a href="index.php" class="menu-link">
    <span class="cufoned">Home</span>
    
  </a>
  <a href="http://www.idangero.us" class="menu-link new-window iframe cufon" rel="900">
    <span class="cufoned">iDangero.us</span>
    
  </a>
  <a href="content/products.php" class="menu-link new-window">
    <span class="cufoned">Products</span>
    
  </a>
  <a href="content/services.php" rel="700" class="menu-link new-window">
    <span class="cufoned">Sevices</span>
    
  </a>
  <a href="content/contacts.php" class="menu-link new-window">
    <span class="cufoned">Contacts</span>
    
  </a>
</div>
```

Menu appearance:



As you can see every menu link contains its title and thumbnail image (square icon with a shadow)

To create your own menu link, first of all you have to create link element inside the **div** with a “menu” id attribute using the **<a>** and **** tags. You’ll need to specify file (with HTML content) location inside of **href** attribute of this link. Add to created link “menu-link” class.

Then, add a span element with a “cufoned” class (not required) and link’s title inside of it. After that put the thumbnail image using the **** tag.

As you can see some of the links also have one of the following classes: “new-window”, “iframe”. More about their usage in chapter 7 (Work With Windows)

Read more about “**cufoned**” class in the “Typography” chapter

So your new link will look like this one:

```
<a href="myfile.php" class="menu-link">
  <span class="cufoned">Link's Title</span>
  
</a>
```

Footer Section

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

```
<div id="footer">
  ----- Footer Content -----
</div>
```

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

5. Window Layout Templates

Content and iFrame layouts

These two main layout templates are located in the **window.tmpl.php** file located in the root folder of website. Here are screenshots if window with HTML content and iFrame content:



Let's look at the window.tmpl.php file content:

All these variables will be posted by JavaScript

```

1 <?php
2 $id = $_GET['id'];
3 $url = $_GET['url'];
4 if (!empty($_GET['width'])) $width = $_GET['width'];
5 else $width = 60;
6 $contentType = $_GET['contentType'];
7 if ($contentType=="iframe") $iframe=1;
8 else $iframe = 0;
9 ?>
10 <div class="window" style="z-index:4000;width:<?php echo $width ?>px" id="window_<?php echo $id ?>">
11   <div class="w-tbl">
12     <div class="w-tbr">
13       <div class="w-tbm">
14         <div class="toolbar">
15           <img class="minimize" onclick="minimize(<?php echo $id ?>)" src="images/black/minimize.png" width
16           = "20" height="20" alt="-" title="Minimize Window" />
17           <img class="close" onclick="closeWindow(<?php echo $id ?>)" src="images/black/close.png" width=
18           "20" height="20" alt="X" title="Close Window" />
19         </div>
20       </div>
21     </div>
22   <div class="w-ml">
23     <div class="w-mr">
24       <div class="w-mm">
25         <div class="w-content">
26           <?php
27             if ($iframe == 0 ) include($url) ;
28             if ($iframe == 1) { ?>
29               <iframe frameborder="0" src="<?php echo $url ?>" width="100%" height="400"></iframe>
30             <?php ?>
31           </div>
32         </div>
33       </div>
34     <div class="w-bot">
35       <div class="w-bl">
36         <div class="w-br">
37           <div class="w-bm"></div>
38         </div>
39       </div>
40     </div>
41 </div>
  
```

Window container

We need an opportunity open window with a desired width

We need to use unique ID attribute for every window to make it unique.

This is a grey toolbar with minimize (expand) and close buttons

"w-ml" is a middle window container. It contains some style divs and a window content

All content will be placed inside this div

If we are opening window with HTML content, then the required file with content will be included here.

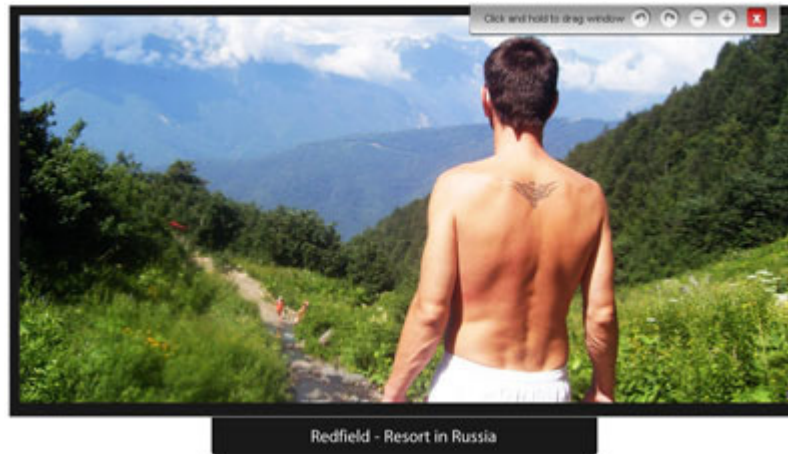
If we are opening window with iFrame content, then the iFrame will be added here with a "src" attribute equal to \$url variable

"w-bot" is a bottom window part. We need it for styling

End of window container

Image layout

This type of layout is used to open window with an image (or photo) inside. It has awesome image controller which allows you to rotate and scale this window.



Its layout template is located in the **img.tmpl.php** file in the root folder of website:

```

1 <?php
2 $id = $_GET['id'];
3 $url = $_GET['url'];
4 if (!empty($_GET['title'])) $title = $_GET['title'];
5 $imageSize = getimagesize($url)
6 ?>
7
8 <div class="window withImage" style="z-index:4000;width:<?php echo ($imageSize[0]+20) ?>px" id=
"window_<?php echo $id ?>">
9     <div class="w-content">
10         <div class="toolbar">
11             )" alt="rotate ccw" />
12             )" alt="rotate cw" />
13             
14             
15             <img class="close" onclick="closeWindow(<?php echo $id ?>)" src="images/black/close.png"
width="20" height="20" alt="X" title="Close Window" />
16         </div>
17         " width="<?php
echo $imageSize[0] ?>" />
18         <?php if(!empty($title)) { ?>
19             <div class="image-title cufoned"><?php echo $title ?></div>
20             <?php } ?>
21         </div>
22 </div>

```

This is a grey toolbar with rotate, scale and close buttons

Requested image will be loaded here

Container with an Image title

6. Typography

Additional Myriad Pro font usage

Device is powered with an awesome additional Myriad Pro font. Its usage is available due to the Cufon JavaScript library.

To use Myriad Pro font for template elements all you need is to add “cufoned” class to them.

Window Headings

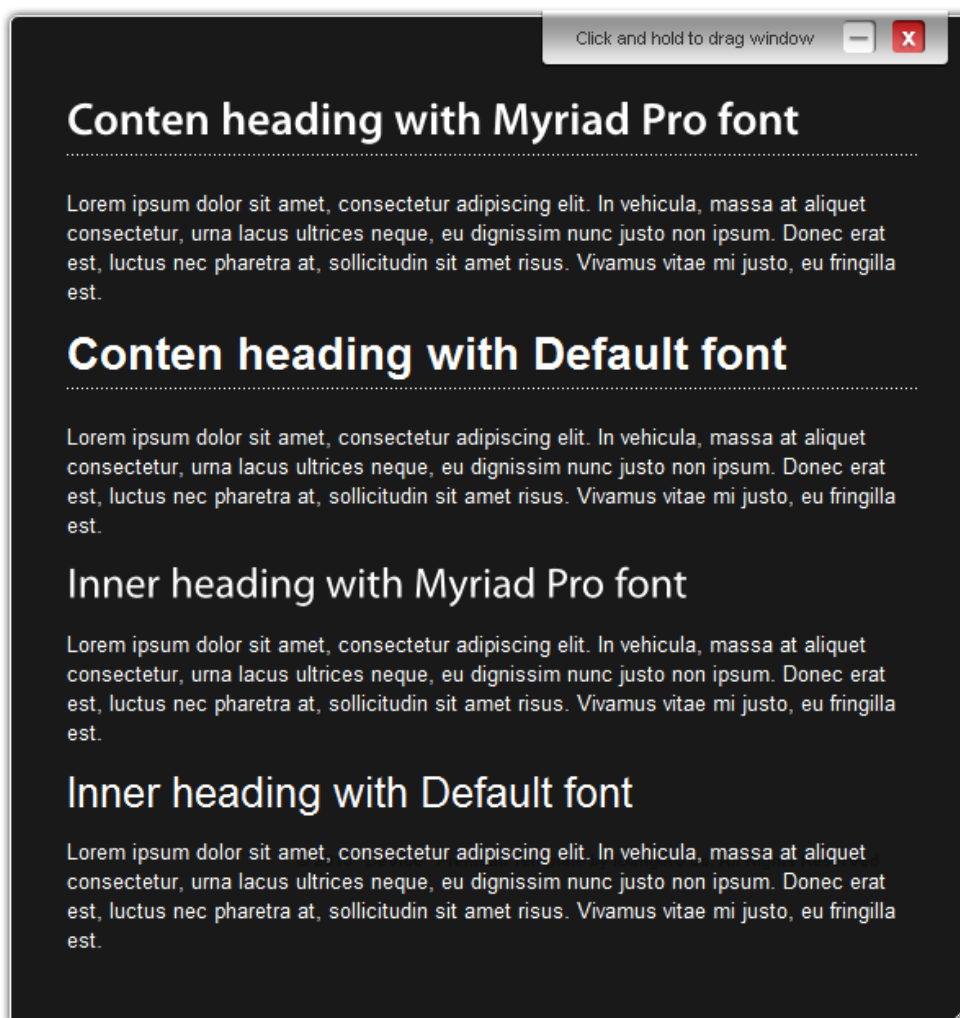
Let’s look at the four different window headings available in Device. You should use the following formatting for them:

```
<h2 class="content-heading cufoned">Conten heading with Myriad Pro font</h2>
```

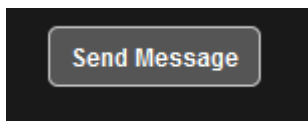
```
<h2 class="content-heading">Conten heading with Default font</h2>
```

```
<h3 class="inner-heading cufoned">Inner heading with Myriad Pro font</h3>
```

```
<h3 class="inner-heading">Inner heading with Default font</h3>
```



Button



This is the button. You should use the following formatting:
` ... Button text ... `
`<input type="submit" class="button" value="Button Text" />`

7. Work With Windows

So let's see how to open new windows with desired content and required parameters.

Window with a HTML content

To open such a window all you need is to specify additional "new-window" class for the link and to specify file (with HTML content) location.

Example 1. We want to open new window with HTML content from the **content/about.php** file directly from the simple link in the text. We should use the following formatting:

```
<a href="content/about.php" class="new-window">About</a>
```

Location of the file with HTML content

When we'll specify a "new-window" class, the script will load file content with Ajax into the new window

Example 2. We want to open new window with HTML content from the **content/features.php** file directly from the simple link in the text. And we need to specify size of new window. We should use the following formatting:

```
<a href="content/features.php" class="new-window" rel="800" rev="600">Features</a>
```

Location of the file with HTML content

Width of new window. Must be specified in REL attribute.

Height of new window. Must be specified in REV attribute.

After that new window will be opened with required width and height

Window with iFrame

To open such a window all you need is to specify additional “new-window” and “iframe” classes for the link and to specify iFrame URL.

Example. We want to open new window with iFramed Google.com inside directly from the simple link in the text. We should use the following formatting:

```
<a href="http://www.google.com" class="new-window iframe">Google</a>
```

iFrame URL

Additional “**iframe**” class will tell to JavaScript to open new window with iFrame inside

It also possible to specify window width and height like in **Example 2** above using **REL** and **REV** attributes

Window with Image layout

To open window with Image (or Photo) inside all you need is to specify additional “new-window” and “withImg” classes for the link and to specify image URL.

Example. We want to open new window with image inside directly from the simple link in the text. We should use the following formatting:

```
<a href="images/my_image.jpg" title="It is me" class="new-window withImg">My Photo</a>
```

image URL

You can specify additional image title. It will be show in the bottom of window.

Additional “**withImg**” class will tell to JavaScript to open new window with image inside

Opening Window

Device API allows you to open new window in three ways:

1. New content could be opened in new window
2. New content could be opened in new window and window with clicked link will close
3. New content could be opened in window with clicked link

To use the first way all you need is to use link in examples above.

To use the second way you need to specify an additional “**closeWindow**” class for “new-window” link like this:

```
<a href="content/about.php" class="new-window closeWindow">About</a>
```

After you click such a link, new window will be opened and window with this link will be closed.

To use the third way you need to specify an additional “**openHere**” class for “new-window” link like this:

```
<a href="content/about.php" class="new-window openHere">About</a>
```

For this case you can specify width and height with REL (for width) and REV (for height) attributes:

```
<a href="content/about.php" rel="800" rev="600" class="new-window openHere">About</a>
```

But now to make it works correctly both attributes must be specified!

Third way will not work if you'll try to open window with image or iFrame layout. Only for Content layout!

8. CSS Files and Structure

There are three CSS files used in this Template. You can find css attachments in the HEAD section of index.php file.

```
<head>
-----
<link rel="stylesheet" type="text/css" href="css/style-white.css" />
<link rel="stylesheet" id="blackStyle" type="text/css" href="css/style-black.css" />
<link rel="stylesheet" type="text/css" href="css/ui-lightness/jquery-ui-1.8.4.custom.css" />
-----
</head>
```

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

style-white.css file contains default white color theme styles and all general layouts for the template. It is separated into appropriate sections:

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

Common rules, such as main backgrounds, fonts, their size, headings, links

/* --- Slideshow --- */

Slideshow styling

/* --- Change Color Theme --- */

Style rules for “change color theme” button and “white-layer”

/* --- Control Panel --- */

Control panel styling. Position of cp links

/* --- Windows Styling --- */

Rules for window styling, its background, toolbar, spacing, size, position, text colors, links

/* --- Window with Image Styling --- */

Some additional rules for window with Image layout

/* --- Menu --- */

Style rules for menu links, their position, size, color

/* --- Forms --- */

Common rules for form elements such as inputs, buttons, textarea

/* --- Footer --- */

Rule for footer position, size and color


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

Rule to stylize Ajax loader. Its position, background, border, color

style-black.css file contains some extra styles for black color theme

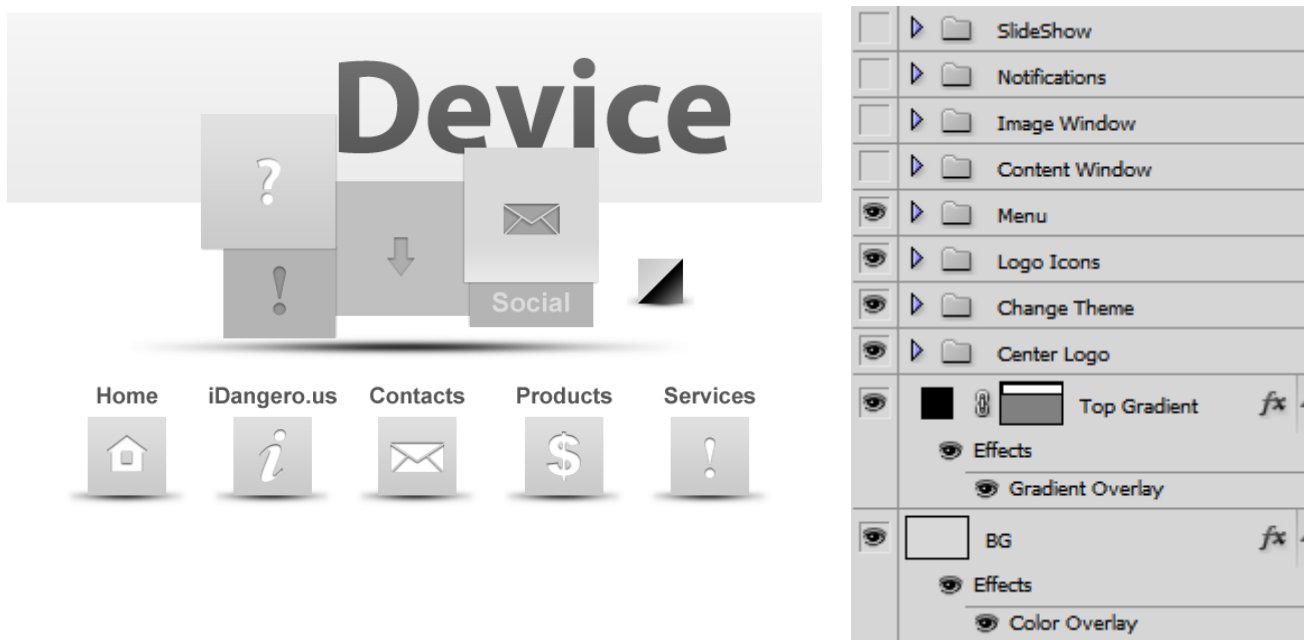
jquery-ui-1.8.4.custom.css file contain styles used for UI elements of jQuery UI library.

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.

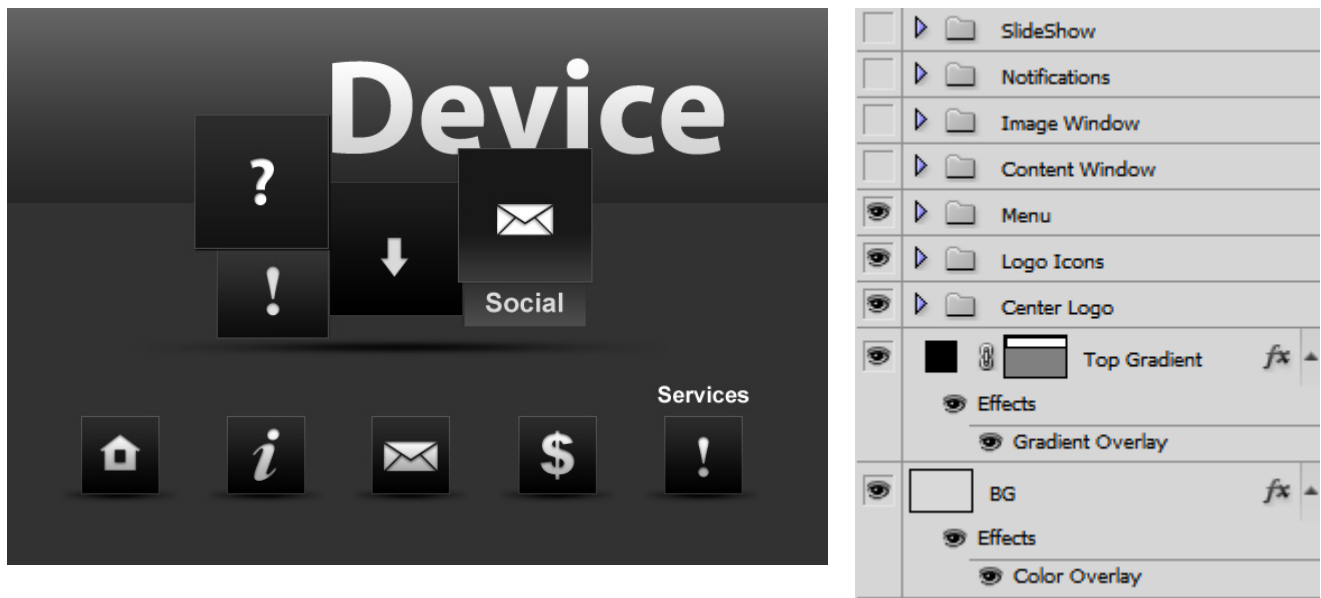
9. PSD Files & Structure

There are the **four** layered **.psd** files comes with the **Device** template. You can locate them into the **psd/** folder form 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)

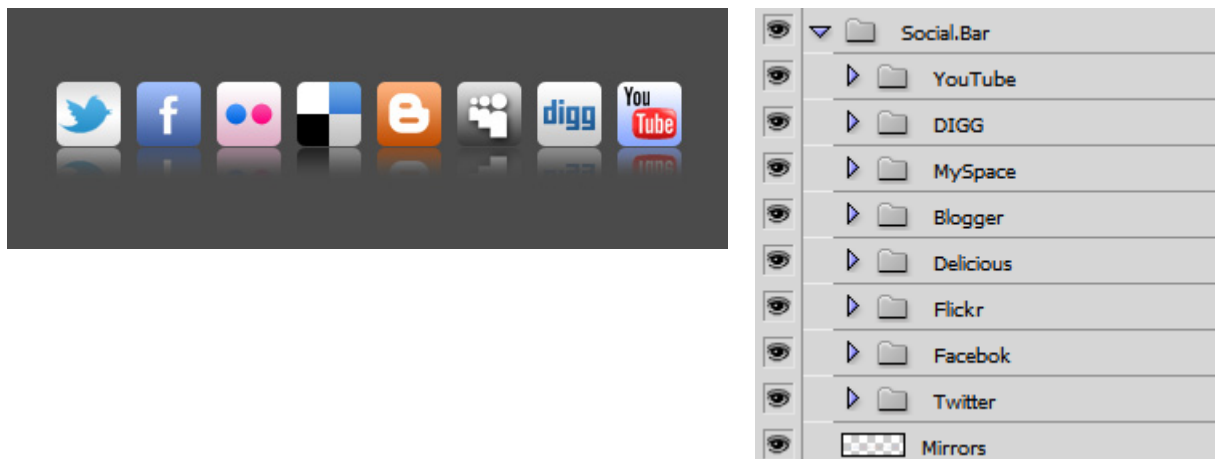
First .psd file is the **device-white.psd**. It contains all general layouts for template styling:



Second .psd file is the **device-black.psd**. It contains layouts for black color theme:



Third file is the **social-bar.psd**. It contains social icons styling:



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

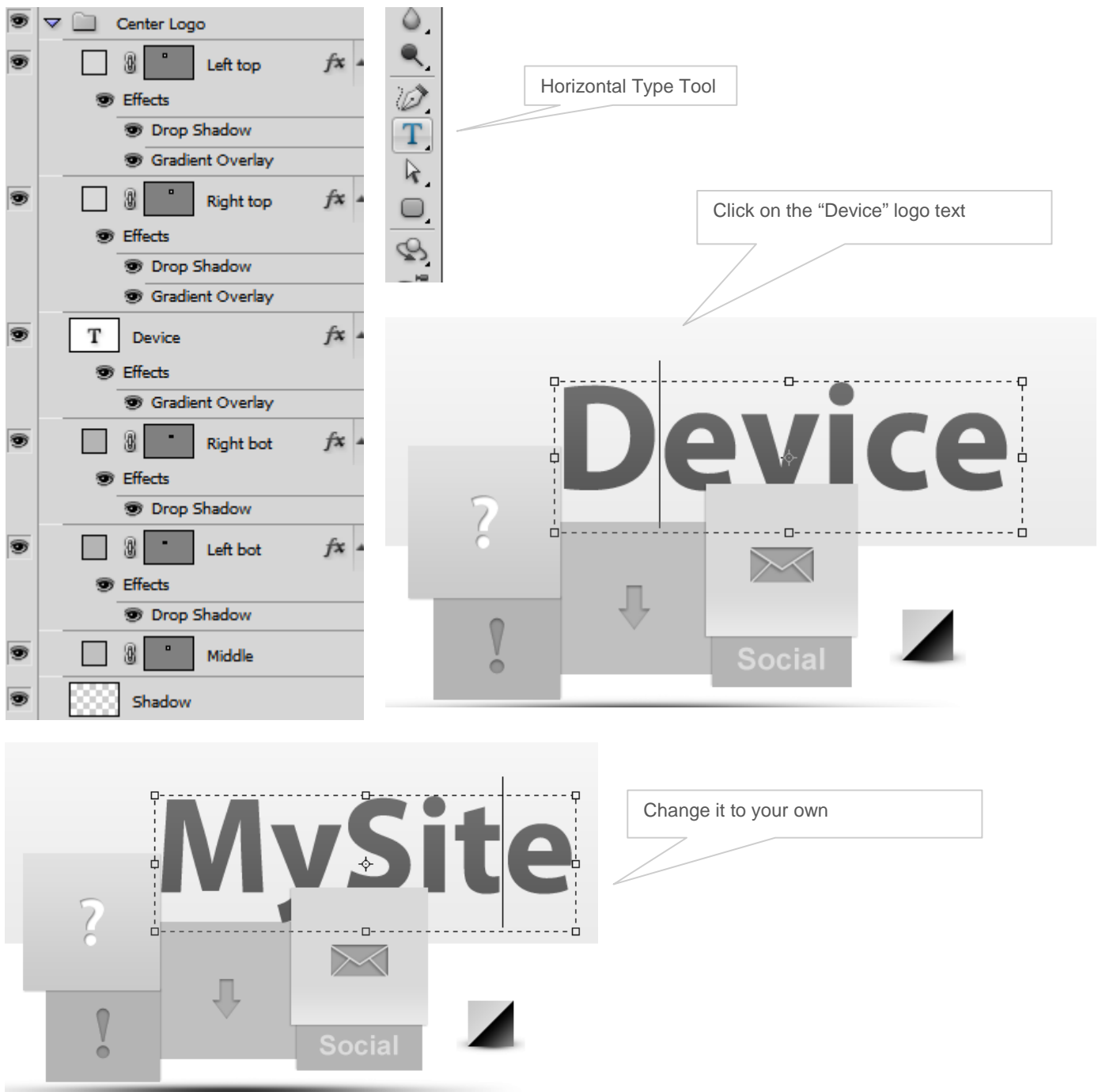
All layers in these file are well organized and easy to understand and redesign.

10. Customization

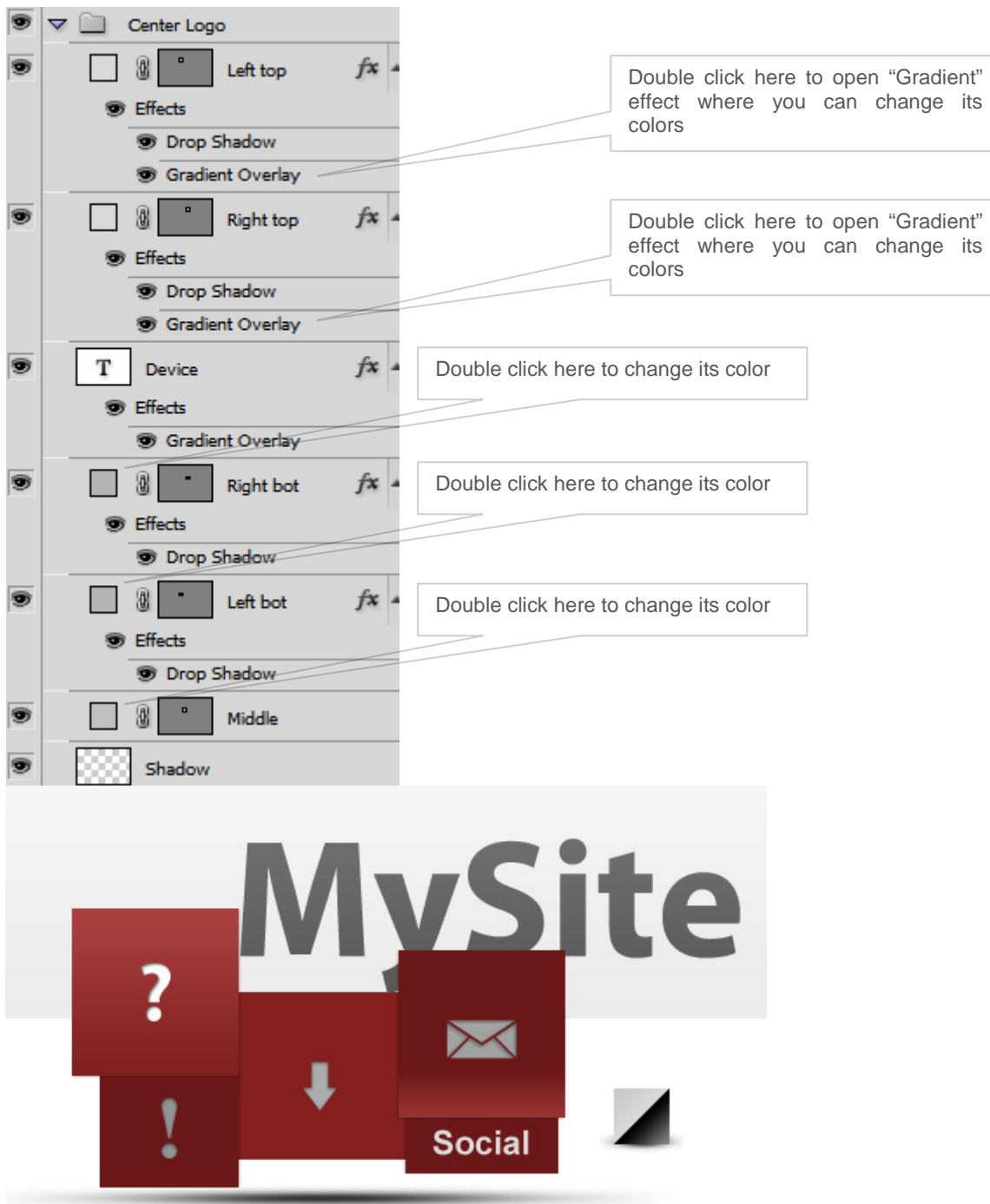
To customize Device template you need to use included psd files and make some changes in template CSS files.

Let's for example change logo and color of Control Panel rectangles and background image.

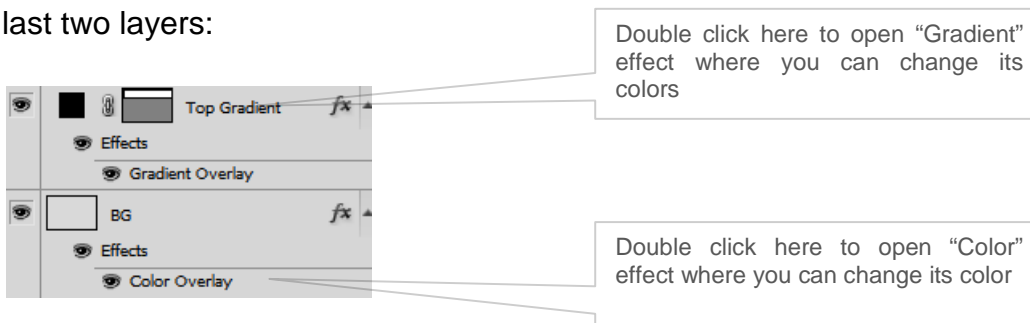
First of all open device-white.psd file. Find "Center Logo" layers group and expand it, choose "Horizontal Type Tool", click on the "Device" logo text and type your own:



Then let's change color of grey rectangles to red (for example):



Then we have to change the background color and color of logo text. Use the same rules for last two layers:



What we have:



All we have to do is to cut these images for web usage. Go to **images/layouts-white/** folder. Find the appropriate files and pay attention to their size. It will be very good if it will be the same. In this folder we should replace **bg.png** and **logo.png** images. Bg.png – is the 10px in width and full height cut “Top Gradient” layer.

After file replacement we need to open `css/style-width.css` and to find rules for body background and for Control Panel.

Body style rule is the first in file. We need to change its background color

```
body {  
    background:#fff url(../images/layouts-white/bg.png) repeat-x left top;  
    font-family:Arial, Helvetica, sans-serif;  
    margin:0px;  
    font-size:13px;  
    line-height:1.4em;  
    color:#fff;  
}
```

We have to change to:

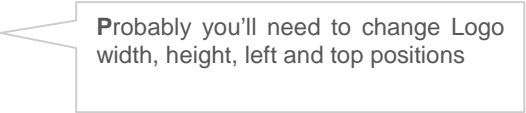
#fff (white) color is changed to
#c44c4c

```
body {  
    background:#c44c4c url(../images/layouts-white/bg.png) repeat-x left top;  
    font-family:Arial, Helvetica, sans-serif;  
    margin:0px;  
    font-size:13px;  
    line-height:1.4em;  
    color:#fff;  
}
```

Actually if you replace CP logo with a same size image, don't need to change background property for it. Otherwise you probably will need to fix its size and position:

Here is rule for Control Panel:

```
#cp {  
  background:url(../images/layouts-white/logo.png) no-repeat left top;  
  width:540px;  
  height:270px;  
  position:absolute;  
  left:80px;  
  top:170px;  
}
```



As you can see it's very easy to customize Device template, all you need is to find appropriate layer in psd file, replace existing image and make some changes in CSS file.

11. JavaScripts

Device Template uses five 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-1.5.1.min.js"></script>  
<script type="text/javascript" src="scripts/jquery-ui-1.8.10.min.js"></script>  
<script type="text/javascript" src="scripts/cufon-yui.js"></script>  
<script type="text/javascript" src="scripts/Myriad_Pro_400-Myriad_Pro_600.font.js"></script>  
<script type="text/javascript">  
  Cufon.replace('.cufoned');  
</script>  
<script type="text/javascript" src="scripts/device.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). [Visit official website.](#)

Second script file (jquery-ui-1.8.10.min.js) is the jQuery UI library. It makes available to drag and resize windows. [Visit official website.](#)

Third script file (cufon-yui.js) is the Cufon library. Used here for fast text with Myriad Pro font. [Visit official website.](#)

Forth script file (Myriad_Pro_400-Myriad_Pro_600.font.js) is the encoded Myriad Pro font.

And the fifth script file (device.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.

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 etc.

First of all we need to disable css file with a black color theme

When we'll click on the "change-color" button the the changeColor() function will be executed

We have to fix "white-layer" top position after window is scrolled

This code means that when the Ajax request is starting, then the showLoader() function will be executed

After the Ajax request is completed, then the hideLoader() function will be executed

"Open Slideshow" animation. After the "openSS" link is clicked, then the div with a "slideshow" id attribute will fade in for 600ms after the 600ms delay. Control panel will slide to new top position (420px) for 600ms and Menu will slide to top position 700px for 600ms

"Open Slideshow" animation. After the "openSS" link is clicked, then the div with a "slideshow" id attribute will fade in for 600ms after the 600ms delay. Control panel will slide to new top position (420px) for 600ms and Menu will slide

Slide-right animation for slideshow slides

Slide-left animation for slideshow slides

"Close Slideshow" animation. Inversed to Open animation

Hover animation for CP links. When we'll take the mouse cursor on the link, then the children "img" element will fade in for 300ms.

Otherwise it will fade out for 600ms.

```

1 // JavaScript Document
2 $(function() {
3     /*-----Change Color-----*/
4     document.getElementById("blackStyle").disabled = 1;
5     $(".change-color").click(function() {changeColor() })
6     $(window).scroll(function() {
7         $("#white-layer").css({top:$ (window).scrollTop() })
8     })
9     /*-----Ajax Loader-----*/
10    $(".body").ajaxStart(function() {
11        showLoader();
12    })
13    $(".body").ajaxComplete(function() {
14        hideLoader();
15    })
16    /*-----Slideshow-----*/
17    $(".openSS").click(function(e) {
18        e.preventDefault();
19        $("#slideshow").delay(600).fadeIn(600)
20        $("#cp").animate({top:420},600)
21        $("#menu").animate({top:700},600)
22    })
23    $(".slide-right .slide-arrow").click(function() {
24        if($(".slide").css("left")!="-" + $(".slide").length-1)*700 + "px")
25        $(".slide").animate({left:"-=700px"},1000)
26    })
27    $(".slide-left .slide-arrow").click(function() {
28        if($(".slide").css("left")!="0px") $(".slide").animate({left:"+=700px"},1000)
29    })
30    $(".close-ss").click(function() {
31        $("#slideshow").fadeOut(600)
32        $("#cp").delay(600).animate({top:170},600)
33        $("#menu").delay(600).animate({top:450},600)
34    })
35    /*-----CP-----*/
36    $("#cp a").hover(
37        function() {
38            $(this).children("img").fadeIn(300)
39        },
40        function() {
41            $(this).children("img").fadeOut(300)
42        }
43    )

```

```

44  /*-----menu-----*/
45  $(".openMenu").click(function(e) {
46      e.preventDefault();
47      for (var i=1;i<= $(".menu-link").length;i++) {
48          $(".menu-link").eq(i-1).delay((i-1)*200).fadeIn(600)
49      }
50  })
51  $(".menu-link:last-child").css({marginRight:0})
52  $(".menu-link").hover(
53      function() {
54          $(this).children("img").animate({width:"89px"},150)
55      },
56      function() {
57          $(this).children("img").animate({width:"109px"},150)
58      }
59  )
60  $(".a.new-window").click(function(e) {
61      e.preventDefault();
62      var contentType;
63      if($(this).hasClass("iframe")) contentType = "iframe";
64      else if($(this).hasClass("withImg")) contentType = "image";
65      else contentType = "text";
66      newWindow({
67          url : $(this).attr('href'),
68          width : $(this).attr("rel"),
69          height : $(this).attr("rev"),
70          contentType : contentType,
71          title:$(this).hasClass("withImg")?$(this).attr('title'):"",
72          left : $(this).offset().left,
73          top : $(this).offset().top
74      })
75  })
76  })

```

"Open Menu" animation. After the "openMenu" link is clicked, then every menu link will fade in for 600ms after the delay time equal to (200ms)*(Link index number). This delay will add gradual fade effect

Hover animation for Menu links. When we'll take the mouse cursor on the link, then the width of children "img" element will be changed to 89px for 150ms

Otherwise, width of children "img" element will be changed to 109px for 150ms

Handling for links with a "new-window" class.

First of all we need to find out content type of new window: HTML content, iframe or image

Then the newWindow() function will be executed with the following parameters

Left and top positions of new window will be equal to appropriate positions of clicked link

The next part of code contains functions which could be called from any place of the document. These functions are used to load new windows, for submitting and validation of the contact form, to handle opened windows:

This function is used to show Ajax loader and to set its top position equal to browser's window scrolled top position + 200px;

```

77 function showLoader() {
78     $(".ajax-loader").css({top:$(window).scrollTop()+200}).show();
79 }
80 function hideLoader() {
81     $(".ajax-loader").hide();
82 }

```

This function is used to hide Ajax loader after the Ajax request is completed

expand() function provides an opportunity to expand minimized windows through the number of animation effects. As parameters this function accepts ID number, width and height of window which is need to be expanded. And after the window is expanded, the "expand" button will be replaced with a "minimize" button.

```

83 function expand(id,windowWidth,windowHeight) {
84     $("#window_"+(id)).resizable("enable");
85     $("#window_"+(id)).animate({
86         width: windowWidth,
87         height: windowHeight
88     },{queue:false},300)
89     $("#window_"+(id)).find("iframe").animate({height:windowHeight-80},300)
90     $("#window_"+(id)+".w-ml").animate({width:windowWidth-20,height:windowHeight-40},{queue:false},300)
91     $("#window_"+(id)+".expand").remove()
92     $("#window_"+(id)+".toolbar").append('')
93 }
94 function minimize(id) {
95     $("#window_"+id).animate({width:"290px",height:"50px",left:"10px"},300,function() {
96         $(this).children(".w-ml").css({height:"10px",width:"270px"})
97         $(this).find("iframe").css({height:"120px",width:"100%"})
98         $(this).addClass("minimized")
99         $(this).fadeTo(300,0.3)
100         $(this).resizable("disable")
101     })
102     $("#window_"+id+".toolbar").append('');
103     $("#window_"+id+".minimize").remove()
104 }
105 function closeWindow(id) {
106     $("#window_"+id).remove()
107 }
108 function newWindow(params) {
109     params.width = params.width || 600;
110     params.contentType = params.contentType || "text";
111     params.left = params.left || 0;
112     params.top = params.top || 0;
113     params.title = params.title || "";
114     var windowID;
115     if ($("#window:last").index() == "-1") windowID = 0;
116     else windowID = ($("#window:last").attr("id").split("window_")[1])*1+1;
117     if (params.contentType != "image") {
118         jQuery.get("window.tmpl.php",{id : windowID, url : params.url, contentType : params.contentType, width : params.width},function(tmpl) {
119             $("#windows").append(tmpl)
120             var windowTop = $(window).scrollTop()+100;
121             var windowHeight = params.height || ($("#window:last").height())
122             var addedWindow = $("#window_"+windowID);
123             addedWindow.css({
124                 width:290,
125                 left: params.left,
126                 top: params.top,
127                 height: 50,
128                 opacity:0.1,
129                 filter: "alpha(opacity=10)"
130             })
131             addedWindow.animate({
132                 left: ($(window).width() - params.width)/2+"px",
133                 top: $(window).scrollTop()+100,
134                 height: windowHeight,
135                 width: params.width,
136                 opacity:1
137             },400,function() {
138                 addedWindow.css({filter:""})
139                 $(".window:lt("+addedWindow.index()+")").fadeTo(300,0.3)
140             })
141         })
142     }
143 }

```

minimize() function provides an opportunity to minimize windows through the number of animation effects. And after the window is expanded, the "minimize" button will be replaced with an "expand" button.

Simple function which is used to close required window.

newWindow() is the main function which is used to open new window

Default window width is 600px

Default content type

Default window positions

Default window title is empty value

New window ID number

This case for HTML and iFrame layouts

Ajax GET request to window.tmpl.php file with a list of parameters

New window will be added in the end of div container with a "windows" id attribute

We have to add this style properties to just added window before its "show" animation

Popup animation for new window. Left, top, height, width and opacity properties will be animated to the following values for 400ms

All other windows will fade to opacity 0.3 for 300ms

With window we also have to animate inner elements: "w-ml" container and iframe

```
141 $("#window_"+windowID+" .w-ml").animate({height:windowHeight-40},400)
142 $("#window_"+windowID+" iframe").animate({height:windowHeight-80},400)
143 addedWindow.draggable({ handle: 'div.toolbar' });
144 addedWindow.resizable({
145     alsoResize: '#window_'+windowID+' .w-ml, #window_'+windowID+' iframe',
146     minHeight: 50,
147     minWidth:290
148 });
149 addedWindow.mousedown(function() {
150     if($(this).css("opacity")!=1) {
151         $(this).css({"z-index":3000}).fadeTo(300,1,function(){$(this).css({"filter":""}
152     ))
153     $(this).nextAll(".window").css({"z-index":2000}).fadeTo(300,0.3)
154     $(this).prevAll(".window").css({"z-index":2000}).fadeTo(300,0.3)
155     }
156 })
157 initWindowLinks(windowID);
158 }
159 else {
160     jQuery.get("img.tmpl.php",{id : windowID, url : params.url, title : params.title},function
161     (tmpl){
162         showLoader();
163         $("#windows").append(tmpl)
164         var addedWindow = $("#window_"+windowID);
165         var windowHeight = addedWindow.height()
166         addedWindow.draggable({ handle: 'div.toolbar' });
167         if(window.Cufon !== undefined) Cufon.replace("#window_"+windowID+" .cufoned");
168         addedWindow.css({
169             width:365,
170             left: params.left,
171             top: params.top
172         })
173         $("#window_"+windowID+" img.window-image").css({width:345})
174         $("#window_"+windowID+" .window-image")[0].onload = function() {
175             hideLoader();
176             addedWindow.animate({
177                 left: ($(window).width() - windowHeight+20)/2+"px",
178                 top: $(window).scrollTop()+100,
179                 width: windowHeight
180             },{queue : false},400,
181             function(){addedWindow.css({"filter":""})})
182             addedWindow.fadeIn(400)
183             $(".window:lt("+addedWindow.index()+")").delay(600).fadeTo(300,0.3)
184             $("#window_"+windowID+" .window-image").animate({
185                 width:windowWidth-20
186             },400)
187             $("#window_"+windowID+" .image-title").delay(700).slideDown(300)
188         }
189         addedWindow.mousedown(function() {
190             if($(this).css("opacity")!=1) {
191                 $(this).css({"z-index":3000}).fadeTo(300,1,function(){
192                     $(this).css({"filter":""})
193                 })
194                 $(this).nextAll(".window").css({"z-index":2000}).fadeTo(300,0.3)
195                 $(this).prevAll(".window").css({"z-index":2000}).fadeTo(300,0.3)
196             }
197         })
198     })
199 }
```

Here we need to make new window resizable and draggable.

This function will initialize window' links handling. See below

This case for new window with Image layout. First of all it will send Ajax GET request to **img.tmpl.php** file with a list of parameters

New window will be added in the end of div container with a "windows" id attribute

We need to replace such elements with Myriad Pro (used for image title)

Startup position for window before animation

After image is loaded window will be animated and shown

All other windows will fade to opacity 0.3 for 300ms

Image title will slide down for 300ms after the 700ms delay

When this window is not active (transparent) and we'll click on it, it will fade in and other windows will fade out

Handle for "zoom-in" button. When it is clicked, then the window width will be increased on 100px for 300ms

```
197     $("#window_"+windowID+" .zoom-in").click(function() {
198         addedWindow.animate({
199             width:"+=100px",
200             left:"-=50px"
201         },300)
202         $("#window_"+windowID+" .window-image").animate({width:"+=100px"},300)
203     })
204     $("#window_"+windowID+" .zoom-out").click(function() {
205         addedWindow.animate({
206             width:"-=100px",
207             left:"+=50px"
208         },300);
209         $("#window_"+windowID+" .window-image").animate({width:"-=100px"},300)
210     })
211 })
212 }
213 }
```

Image width will be increased too

Handle for "zoom-out" button. When it is clicked, then the window width will be decreased on 100px for 300ms

Image width will be decreased too

This function is used when we are loading new content into already opened window

```
214 function replaceWindowContent(params) {
215     var replacedWindow = $("#window_"+params.windowID);
216     var replacedContent = replacedWindow.find(".w-content");
217     replacedContent.fadeOut(300,function() {
218         jQuery.get(params.url,function(newContent) {
219             replacedContent.html(newContent);
220             initWindowLinks(params.windowID);
221             if(params.width && params.height) {
222                 replacedWindow.animate({
223                     width:params.width,
224                     left:($("window").width()-params.width)/2,
225                     height:params.height
226                 },300)
227                 replacedWindow.find(".w-ml").animate({
228                     height:params.height-40,
229                     width:params.width-20
230                 },300)
231             }
232             replacedContent.delay(600).fadeIn(300)
233         })
234     })
235 }
```

Window content will fade out for 300ms

Then – Ajax GET request to file with a new content

Window content will be replaced

If the Width and Height for new window are specified the window will be animated to new size

New content will fade in for 300ms after the 600ms delay

This function will be executed with every new window. It is used to add handling for the link with special classes to provide an opportunity to open new windows from other windows

```
236 function initWindowLinks(windowID){
237     $("#window_"+windowID+" a.new-window").click(function(e) {
238         e.preventDefault();
239         var contentType;
240         if($(this).hasClass("iframe")) contentType = "iframe";
241         else if($(this).hasClass("withImg")) contentType = "image";
242         else contentType = "text";
243         $(this).attr({"href":$(this).attr("href").split(document.location.href)[1]})
244         newWindow({
245             url : $(this).attr('href'),
246             width : $(this).attr("rel"),
247             height : $(this).attr("rev"),
248             contentType : contentType,
249             title:$(this).hasClass("withImg")?$(this).attr('title'):"",
250             left : $(this).offset().left,
251             top : $(this).offset().top
252         })
253     })
254     $("#window_"+windowID+" .closeWindow").click(function(e) {
255         e.preventDefault();
256         closeWindow(windowID)
257     })
258 }
```

Handling for links with a "new-window" class. The same as above for common links

Handling for links with a "closeWindow" class. Used to close window

```

258 $( "#window_"+windowID+" a.openHere").click(function(e) {
259     e.preventDefault();
260     replaceWindowContent({
261         windowID : windowID,
262         url : $(this).attr('href'),
263         width : $(this).attr("rel"),
264         height : $(this).attr("rev")
265     })
266 })
267 if(window.Cufon !== undefined) Cufon.replace("#window_"+windowID+" .cufoned");
268 }
269 function rotate(degree,windowID) {
270     $( "#window_"+windowID).css({
271         "-o-transition-property":"-o-transform",
272         "-o-transition-duration":"1s",
273         "-o-transform":"rotate("+degree+"deg)",
274         "-webkit-transition-property":"-webkit-transform",
275         "-webkit-transition-duration":"1s",
276         "-webkit-transform":"rotate("+degree+"deg)",
277         "-moz-transition-property":"-moz-transform",
278         "-moz-transition-duration":"1s",
279         "-moz-transform":"rotate("+degree+"deg)",
280         "transition-property":"transform",
281         "transition-duration":"1s",
282         "transform":"rotate("+degree+"deg)"
283     })
284     $( "#window_"+windowID+" .rotate-cw").attr({"onClick":"rotate ("+(degree+5)+" ,"+windowID+" )"})
285     $( "#window_"+windowID+" .rotate-ccw").attr({"onClick":"rotate ("+(degree-5)+" ,"+windowID+" )"})
286 }
287 function changeColor() {
288     var whiteLayer = $( "#white-layer");
289     whiteLayer.css({
290         width:$(window).width(),
291         height:$(window).height(),
292         top:$(window).scrollTop()
293     })
294     setTimeout(cufonRestart,1000);
295     whiteLayer.fadeIn(600,function() {
296         var newBodyClass = $("body").hasClass("white")?"black":"white";
297         var blackCSS = document.getElementById("blackStyle");
298         if (newBodyClass=="black") {
299             $("body").removeClass("white").addClass("black")
300             blackCSS.disabled = 0;
301             $(".menu-link img").each(function() {
302                 $(this).attr({"src" : $(this).attr("src").replace(/white/, "black")})
303             })
304             $(".slide-left img, .slide-right img").each(function() {
305                 $(this).attr({"src" : $(this).attr("src").replace(/white/, "black")})
306             })
307         }
308         else {
309             $("body").removeClass("black").addClass("white")
310             blackCSS.disabled = 1;
311             $(".menu-link img").each(function() {
312                 $(this).attr({"src" : $(this).attr("src").replace(/black/, "white")})
313             })
314             $(".slide-left img, .slide-right img").each(function() {
315                 $(this).attr({"src" : $(this).attr("src").replace(/black/, "white")})
316             })
317         }
318     }).delay(400).fadeOut(600,function() {cufonRestart()})
319 }

```

Handling for links with an "openHere" class. After it is clicked then the replaceContent() function will be executed with a list of parameters

We have to replace all elements with a "cufoned" class with Myriad Pro font in just opened window

rotate() function provides an opportunity to rotate windows with Image layout by means of CSS3 transition property

After the window is rotated we have to change onClick attributes for "rotate" elements

This function is used to "invert" color theme

Style rules for white-layer to make it full width and full height

cufonRestart() function will be executed after the 1000ms delay

White layer will fade in for 600ms

New color class for Body

Change Body class from "white" to "black"

Enabling black-style CSS file

Change "src" attribute for menu images and for slide-show images from "white" to "black"

This case is used for counter color change


```

320 function cufonRestart() {
321     if(window.Cufon !== undefined) {
322         $(".cufoned").each(function() {
323             var cufonContent = "";
324             var cufonedParts = $(this).find("cufontext")
325             for (var i=1; i <= cufonedParts.length; i++) {
326                 cufonContent+=cufonedParts[i-1].innerHTML
327             }
328             $(this).html(cufonContent);
329         })
330         Cufon.replace('.cufoned');
331     }
332 }
333 function validateText(fieldId,length) {
334     field = $("input#" + fieldId);
335     text = field.val();
336     m_length = ($.trim(text)).length;
337     if (m_length < length) field.addClass("required");
338     else field.removeClass("required");
339 }
340 function validateMessage() {
341     message = $("textarea#contact_message").val();
342     m_length = ($.trim(message)).length;
343     if (m_length < 10) $("textarea#contact_message").addClass("required");
344     else $("textarea#contact_message").removeClass("required");
345 }
346 function validateEmail() {
347     var email = $("input#contact_email").attr("value");
348     var re = /^\\w+([\\.-]?\\w+)*@(((([a-z0-9]{2,})|([a-z0-9]{-}[a-z0-9]+))\\.[a-z0-9])|([a-z0-9]+[-
349 ]?)+[a-z0-9]\\.([a-z]{2}|(com|net|org|edu|int|mil|gov|arpa|biz|aero|name|coop|info|pro|museum|
350 mobi|travel)))$/i;
351     if(re.test(email)) $("input#contact_email").removeClass("required");
352     else $("input#contact_email").addClass("required");
353 }
354 function sendMessage() {
355     validateText("contact_name",6)
356     validateText("contact_subject",6)
357     validateEmail()
358     validateMessage()
359     if ($("#contact_email,#contact_message,#contact_name,#contact_subject").hasClass("required"))
360     {
361         newWindow({url:"includes/message_error.php",width:480,height:150})
362     }
363     else {
364         name = $("#contact_name").attr("value");
365         email = $("#contact_email").attr("value");
366         subject = $("#contact_subject").attr("value");
367         message= $("#contact_message").val().replace(/<\\/?[>]+>/gi, '');
368         jQuery.post("includes/send_message.php",{name : name, email : email, subject : subject,
369 message : message},
370         function (status) {
371             if(status==1) newWindow({url:"includes/message_sent.php",width:550})
372             else newWindow({url:"includes/server_error.php",width:500})
373         }
374     );
375     }
376 }

```

This function will replace all canvas elements (Texts with Myriad Pro font) to simple text elements. And will replace them again to canvas elements. We need such action because when color theme is changing, canvas elements will not change their color. So we need to do such transformations

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 "required" class will be added to this field. Otherwise the "required" class 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 "required" class will be added to the contact form **textarea** with a "message" id attribute. Otherwise the "required" class will be removed.

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 "required" class will be added to the email field (input field with an "email" id attribute). Otherwise it 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 "required" class, new window with error message will be opened

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 status number (0 or 1). If status=1 then the new window will be opened with a successful message status, otherwise it will be opened with a server error message

12. Ajax Contact Form

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

```
<h2 class="content-heading cufoned">Contacts</h2>
<p>You can contact us using the following contact form:</p>
<form name="contact" class="contact-form" action="javascript:sendMessage()" method="post">
  <p>
    <label> Your Name:*<br>
    <input id="contact_name" onkeyup="validateText('contact_name',6)" type="text" class="text"
name="name" size="30" />
    </label>
  </p>
  <p>
    <label> Your Email:*<br>
    <input id="contact_email" onkeyup="validateEmail()" type="text" class="text" name="email"
size="30" />
    </label>
  </p>
  <p>
    <label> Subject:*<br>
    <input id="contact_subject" onkeyup="validateText('contact_subject',6)" type="text" class=
"text" name="subject" size="30" />
    </label>
  </p>
  <p>
    <label> Message:*<br>
    <textarea id="contact_message" onkeyup="validateMessage()" name="message" class="textarea"
cols="30" rows="10"></textarea>
    </label>
  </p>
  <p><em>Fields marked with an asterisk(*) are required</em></p>
  <input type="submit" name="Submit" value="Send Message" />
</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 return a status message. Let's look at this script more closely:

```
<?php
if (isset($_POST['name'])) $name = $_POST['name'];
if (isset($_POST['email'])) $email = $_POST['email'];
if (isset($_POST['subject'])) $subject = $_POST['subject'];
if (isset($_POST['message'])) $message = $_POST['message'];
$mail_to = "info@idangero.us"; // change this email to yours
$message = stripslashes($message);
$headers = "From: ".$name." <".$email.">\r\n";
$headers .= "Reply-To: ".$email."\r\n";
$headers .= "X-Mailer: PHPMailer".".\r\n";
$headers .= "Content-Type: text/plain; charset=\\"utf-8\\"";
if (mail($mail_to, $subject, $message, $headers)) echo 1;
else echo 0;
?>
```

Checking for existence of these variables.

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

If the message was successfully sent, the script will return "1", otherwise it will return "0".

Now let's look at the status messages located in the following files:

includes/message_sent.php

```
<h4 class="notification"><span class="message-sent"></span>Thanks! Your message has been  
successfully sent.</h4>
```

includes/server_error.php

```
<h4 class="notification"><span class="message-sent"></span>Thanks! Your message has been  
successfully sent.</h4>
```

includes/message_error.php

```
<h4 class="notification">Required fields are empty or filled not correctly</h4>
```

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

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 **Device** (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**.