



“O.S”

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 “O.S”
2. O.S Files & Folders Structure
3. Installation
4. HTML Structure
5. Inner Linking (How it works?)
6. CSS Files and Structure
7. JavaScripts
8. Window Template (HTML & JavaScript)
9. Ajax Contact Form
10. PSD Files and Structure
11. Licensing Terms

1. About “O.S”

O.S is a "Desktop Style" site template, designed by the iDangero.us team.

This template is perfect for small business websites and IT projects. The template is fully animated without the use of Flash technology. Smooth and good looking animation and Ajax features of this template are realized by using the JavaScript jQuery library and jQuery UI library.

All content loaded on the fly, your users do not need to wait the reloading of entire site.

Page content is displayed in the windows that you can drag, resize or minimize. You can put any content (flash, html, php etc.) inside the "window".

Template does not have a typical site menu with a links. Instead menu, you can see the draggable icon toolbar with the fade effect.

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

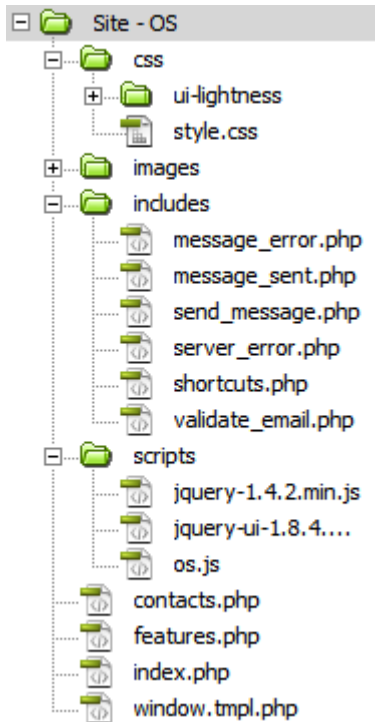
Also **O.S** is powered with an Ajax Contact Form with validation.

Here is a couple of screenshots:



2. O.S Files Structure

O.S files are located in the **os/** 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: 5 php scripts used for Ajax contact form, shortcuts, page script and popup window layout

scripts: JavaScripts used in this template

And the main **index.php**, **window template** and .php files with a content in root folder

3. Installation

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

Index.php <body> structure:

```
<body>
<div id="wrapper">
  <div id="windows"> </div>
  <?php include("includes/shortcuts.php") ?>
  <div id="footer"> --- Footer Content --- </div>
</div>
```

Container for "windows". All content windows will be added right here.

Shortcuts section located in the **includes/shortcuts.php** file

Footer section

Footer Section

Footer section is used for copyrights information of you company or project

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

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

Shortcuts HTML Structure

Now let's look at the shortcuts HTML structure. Open **includes/shortcuts.php** file:

```
<div id="shortcuts">
  <div class="w-tl">
    <div class="w-tr">
      <div class="w-tm"></div>
    </div>
  </div>
  <div class="w-m1">
    <div class="w-mr">
      <div class="w-mm">
        <div class="w-content">
          <a class="shortcut" href="index.php">
            <span></span>
          </a>
          <a class="shortcut new-window" rel="650" title="O.S Template Features" href="features.php" >
            <span></span>
          </a>
          <a class="shortcut iframe" rel="900" title="iFrame Demo" href="http://www.idangero.us" >
            <span></span>
          </a>
          <a class="shortcut new-window" title="Contacts" href="contacts.php" >
            <span></span>
          </a>
          <a class="shortcut" href="#" >
            <span></span>
          </a>
        </div>
      </div>
    </div>
  </div>
</div>
```

w-tl container is used for styling of top part of the shortcuts window.

w-m1 is the main middle container with the shortcuts inside. w-mr and w-mm blocks are used for styling

Link with a shortcut image

This **span** container is used to display link title in the tip style. Value of the **alt** attribute of shortcut image will be added here automatically by JavaScript

```

<a class="shortcut" href="#" >
    <span></span>
</a>
<a class="shortcut" href="#" >
    <span></span>
</a>
<a class="shortcut" href="#" >
    <span></span>
</a>
<a class="shortcut" href="#" >
    <span></span>
</a>
<a class="shortcut" href="#" id="font_inc" >
    <span></span>
</a>
<a class="shortcut" href="#" id="font_dec" >
    <span></span>
</a>
</div>
</div>
</div>
<div class="w-bot">
    <div class="w-bl">
        <div class="w-br">
            <div class="w-bm"></div>
        </div>
    </div>
</div>
</div>
</div>

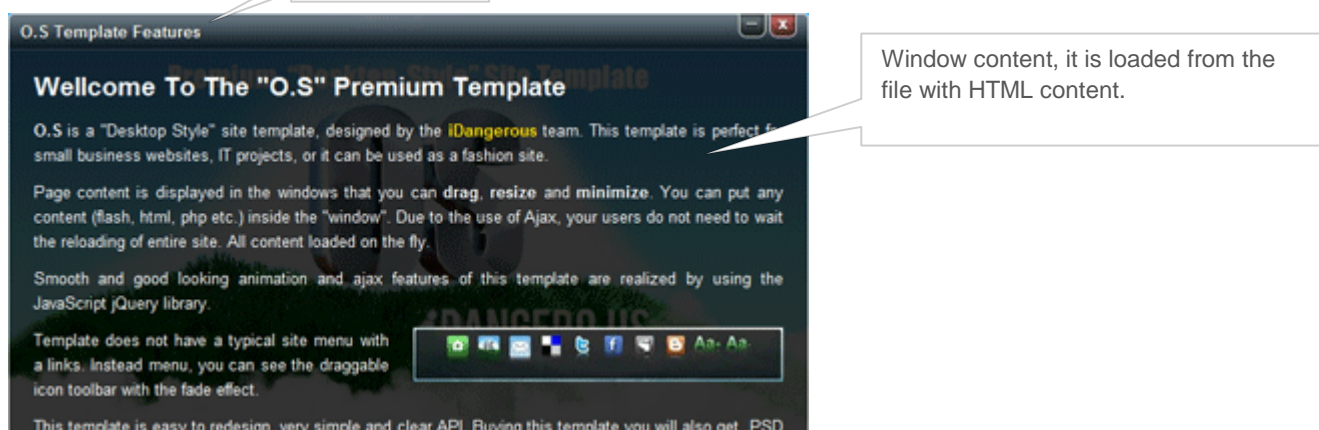
```

w-bot container is used for styling of bottom part of the shortcuts window.

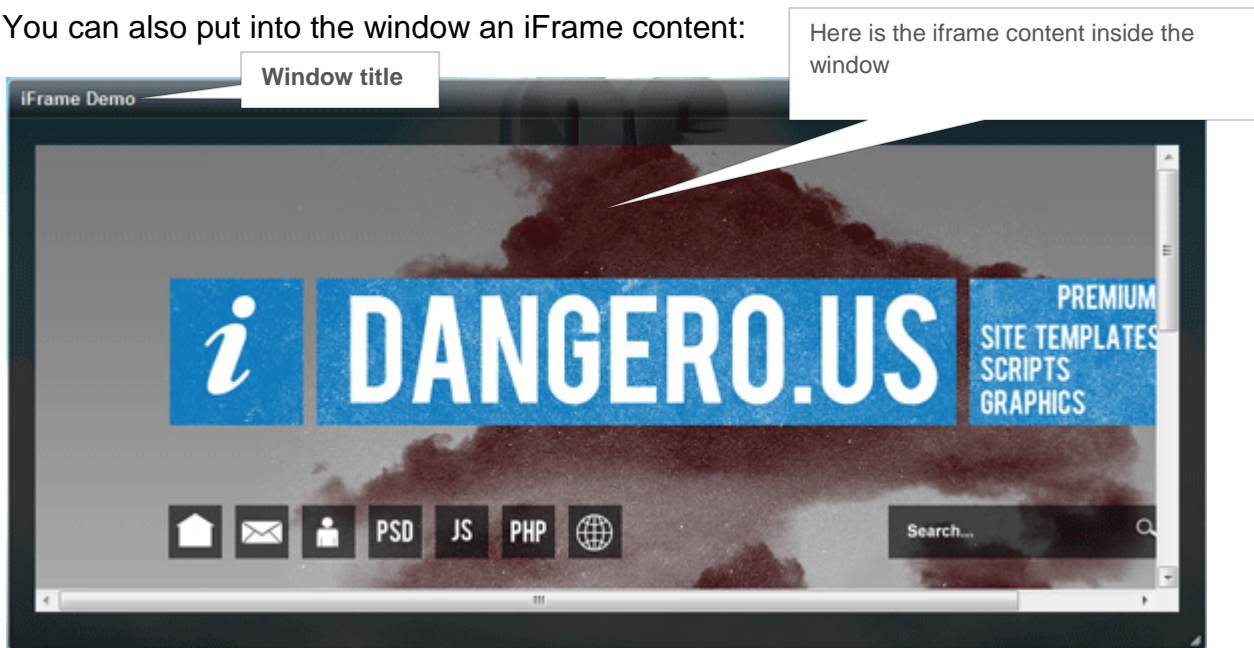
To create your own shortcut you have to create a link using the simple `<a>` tag with a *“shortcut”* class inside the *div* block with a *“w-content”* class. Then you need to put a shortcut image inside this link using the `` tag with a specified *alt* attribute. And the finally you have to add `` block after the shortcut image. That's all, new shortcut is ready to use.

5. Inner Linking (How it works?)

As mentioned above all content pages (files with HTML content) are loading into the window like this one:



You can also put into the window an iFrame content:



Now let's see how to put desirable type of content (HTML or iFrame) into the window.

Shortcuts Linking

Example 1. We want to load HTML content from the **features.php** file using the shortcut link. We should use the following formatting for shortcut link:

```
<a class="shortcut new-window" rel="650" title="O.S Template Features" href="features.php" >  
  <span></span>  
</a>
```

When the **"new-window"** class is specified, then the JavaScript will open this link in **window** with HTML content inside.

We should use the **"rel"** attribute to specify window width. In this case window width is equal to 650px.

"title" attribute will be used by JavaScript to specify window title.

And here is the location of the file with HTML content we want to load.

Example 2. We want to put iFrame into the window using the shortcut link. We should use the following formatting for shortcut link:

```
<a class="shortcut iframe" rel="900" title="iFrame Demo" href="http://www.idangero.us" >  
  <span></span>  
</a>
```

When the **"iframe"** class is specified, then the JavaScript will open this link in **window** with iFrame inside.

We should use the **"rel"** attribute to specify window width. In this case window width is equal to 900px.

"title" attribute will be used by JavaScript to specify window title.

And here is the URL location of the site which will be loaded inside iFrame.

*If you want to use shortcut as an external link, just don't use **"new-window"** or **"iframe"** class for this link.*

Window Linking

Example 3. We want to load new window with HTML content from the **features.php** file using the simple link from another window. We should use the following formatting for this link:

```
<a href="features.php" title="Features" rel="600" class="in-window">Template Features</a>
```

Location of the file with HTML content we want to load.

"title" attribute will be used by JavaScript to specify window title.

"rel" attribute is used to specify window width. In this case window width is equal to 600px.

When the **"in-window"** class is specified, then the JavaScript will open this link in **window** with iFrame inside.

Example 4. We want to load new window with an iFrame inside using the simple link from another window. We should use the following formatting for this link:

```
<a href="http://www.google.com" title="Google" rel="600" class="in-iframe">Google</a>
```

URL location of the site which will be loaded inside iFrame.

"title" attribute will be used by JavaScript to specify window title.

"rel" attribute is used to specify window width. In this case window width is equal to 600px.

When the **"in-iframe"** class is specified, then the JavaScript will open this link in **window** with HTML content inside.

As you can see it is very easy to use Inner Linking. All you need is:

- to specify file (or URL for iFrame) location using the **href** attribute,
- to specify the "content type" class (**new-window** and **iframe** or **in-window** and **in-iframe**),
- to specify window width with a **"rel"** attribute (*not necessary*)
- and window title with a **"title"** attribute (*not necessary*)

6. CSS Files and Structure

There are two 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.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.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.

```
/* --- Window Styling --- */
```

This section contains style rules for all window elements

```
/* --- Shortcuts Section --- */
```

Rules for shortcut icons

```
/* --- Forms --- */
```

Rules for contact form and all its elements

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

All rules for the Products Section

```
/*----- 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.

jquery-ui-1.8.4.custom.css file contains style rules used in jQuery UI library.

7. JavaScripts

O.S Template uses three JavaScript files and one inline script for the “window” (chapter 8).

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

```
<head>
-----
<script type="text/javascript" src="scripts/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="scripts/jquery-ui-1.8.4.min.js"></script>
<script type="text/javascript" src="scripts/os.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 (jquery-ui-1.8.4.min.js) is the jQuery UI JavaScript library.

And the third script file (os.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.

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 shortcut links, its animation and Ajax loader.

```
2 $(function() {
3     /*-----Ajax Loader-----*/
4     $("body").ajaxStart(function() {
5         $(this).prepend('<div class="ajax-loader"></div>');
6     })
7     $("body").ajaxComplete(function() {
8         $(".ajax-loader").remove();
9     })
10    /*-----Shortcuts-----*/
11    $("a.new-window").click(function(e) {
12        e.preventDefault();
13        newWindow($(this).attr('href'), $(this).attr('title'), $(this).attr("rel"), 0)
14    })
15    $("a.iframe").click(function(e) {
16        e.preventDefault();
17        newWindow($(this).attr('href'), $(this).attr('title'), $(this).attr("rel"), 1)
18    })
19    $("#shortcuts").css({
20        "left": ($(window).width() - $("#shortcuts").width())/2+"px" ,
21        opacity:"0.2"
22    }).draggable()
23    $("#shortcuts").hover(
24        function() {
25            $(this).fadeTo(300,1).css({"z-index":4000})
26        },
27        function() {
28            $(this).fadeTo(300,0.2).css({"z-index":100})
29        }
30    )
31    $(".shortcut").hover(
32        function() {
33            $(this).children("img").animate({width:"+=10px"},200)
34            $(this).children("img").next("span").html($(this).children("img").attr("alt")).fadeIn(200)
35        },
36        function() {
37            $(this).children("img").animate({width:"-10px"},200)
38            $(this).children("img").next("span").fadeOut(100)
39        }
40    )
41    $("#font_inc").click(function(e) {
42        e.preventDefault();
43        $("body").animate({"font-size":"+=2px"},1)
44    })
45    $("#font_dec").click(function(e) {
46        e.preventDefault();
47        $("body").animate({"font-size":"-=2px"},1)
48    })
49 })
```

This code means that when the Ajax request is starting, the HTML code of Ajax Loader will be added to the BODY section of the document.

When the Ajax request is completed, the HTML code of Ajax Loader will be removed from the BODY section of the document.

Shortcut links handling. When you'll click the shortcut link with the "new-window" class, then the newWindow() function will be executed. This case is for the window with HTML content from the file inside

Almost the same as the previous one, but for the case when you'll need to put an iFrame content inside the window.

Used for initialization of the shortcuts toolbar. It will be positioned in the center of the screen (horizontally) with opacity equal to 0.2.

draggable() method provides an opportunity to drag our shortcuts toolbar.

Fade effect for the shortcuts toolbar. When you'll hover on it, it will fade to opacity 1 for 0.3 seconds. When you'll take mouse cursor away, the toolbar will fade to opacity 0.2 for 0.3 seconds.

Animation of the shortcuts. First function will be executed when you'll hover on the shortcut icon. Its size will be increased on 10px for 0.2 seconds. And the "tip" with its title will be shown for 0.2 seconds.

The same function as a previous one, but for the mouseout action.

This function is used for "Increase Font Size" icon. When you'll click it, the font-size will be increased for 2px

This function is used for "Decrease Font Size" icon. When you'll click it, the font-size will be decreased for 2px

The next part of code contains functions which could be called from any place of the document. These functions are used for Ajax requests (to load new windows), for submitting and validation of the contact form, to minimize and expand windows.

```

50 function expand(id) {
51     $("#window_"+(id)).resizable("enable");
52     $("#window_"+(id)).animate({
53         width: "600px",
54         height: "200px"
55     }, 300)
56     $("#window_"+(id)+".w-ml").animate({width:"599px",height:"160px"},100)
57     $("#window_"+(id)+".expand").remove()
58     $("#window_"+(id)+".w-tbm").append('')
59 }
60 function minimize(id) {
61     $("#window_"+(id)).animate({width:"200px",height:"50px",left:"10px"},300,function() {
62         $(this).children(".w-ml").css({height:"10px",width:"198px"})
63         $(this).find("iframe").css({height:"120px",width:"100%"})
64         $(this).addClass("minimized")
65         $(this).fadeTo(300,0.3)
66         $(this).resizable("disable")
67     })
68     $("#window_"+(id)+".w-tbm").append('');
69     $("#window_"+(id)+".minimize").remove()
70 }
71 function newWindow($uri,$title,$width,$iframe) {
72     var $id = $(".window:last").index()+1;
73     jQuery.get("window.tmpl.php",{title : $title, id : $id, width : $width, uri : $uri, iframe : $iframe},
function(tmpl) {
74         $("#windows").append(tmpl)
75         var $top = ($("#window").height() - $(".window:last").height())/2;
76         if($top<0) $top = 50;
77         $(".window:last").css({
78             left: ($("#window").width() - $(".window:last").width())/2+"px",
79             top: $top+"px"
80         })
81         $(".window:last").fadeIn(300)
82         $(".window:not(:last)").fadeTo(300,0.3)
83     })
84 }
85 function strlen(string) {
86     return string.length;
87 }
88 function validateText($id) {
89     if(strlen($.trim($("#"+$id).val())) < 6) {
90         $("#"+$id).addClass("required")
91     }
92     else $("#"+$id).removeClass("required")
93 }
94 function validateMessage() {
95     if(strlen($.trim($("#contact_message").val())) < 10) {
96         $("#contact_message").addClass("required")
97     }
98     else $("#contact_message").removeClass("required")
99 }

```

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

minimize() function provides an opportunity to minimize windows through the number of animation effects. As a parameter this function accepts an ID number of window which is need to be minimized.

And after the window is minimized, the "minimize" button will be replaced with a "expand" button.

newWindow() function used to load new windows

Ajax GET request used to load window template with the following parameters: **title**, **id**, **width**, **uri** and **iframe**

After the AJAX request is completed, new window with its content will be added to the **div** block with a **"windows"** id attribute. Then new window will be positioned in the center of the screen (horizontally),

And after that it will be faded to opacity 1 for 0.3 seconds and all other windows will be faded to opacity 0.3 for 0.3 seconds

This function returns the number of characters in the string

validateText() function is used for validating text fields in the contact form (such a **subject** or **name** fields). If the field contains less than 6 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 **"contact_message"** id attribute. Otherwise the "required" class will be removed.

```

100 function validateEmail() {
101     email = $("#contact_email").attr("value");
102     jQuery.post("includes/validate_email.php",{email: email},
103         function (data) {
104             if (data != 1) {
105                 $("#contact_email").addClass("required");
106             };
107             if (data == 1) {
108                 $("#contact_email").removeClass("required");
109             };
110         }
111     );
112 }
113 function sendMessage() {
114     validateText("contact_name")
115     validateText("contact_subject")
116     validateEmail()
117     validateMessage()
118     if ($("#contact_email,#contact_message,#contact_name,#contact_subject").hasClass("required")) {
119         newWindow("includes/message_error.php","Sending Error",400)
120     }
121     else {
122         name = $("#contact_name").attr("value");
123         email = $("#contact_email").attr("value");
124         subject = $("#contact_subject").attr("value");
125         message= $("#contact_message").val().replace(/<\/?[>]+>/gi, '');
126         jQuery.post("includes/send_message.php",{name : name, email : email, subject : subject, message :
message},
127             function (status) {
128                 if(status==1) newWindow("includes/message_sent.php","Message Successfully Sent",400)
129                 else newWindow("includes/server_error.php","Sending Error",400)
130             }
131         );
132     }
133 }
134 }

```

validateEmail() function is used for validating email field in the contact form. It uses Ajax request to the php script which returns "1" if the entered email address is correct and "0" if it's wrong. If it's wrong then the "required" class will be added to the email field (input field with a "contact_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 **Name**, **Subject**, **Email** and **Message**). And then if some of the fields has the "required" class, the new window will be opened with an error message. 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. If everything is ok, the new window will be opened with the "successfully" message status. Otherwise new window will contain an "error" message status.

Couple of words about **newWindow()** function. This function accepts four parameters:

- **\$uri** – location of the file with a HTML content or URL for window with an iFrame content
- **\$title** – window title
- **\$width** – window width in px
- **\$iframe** – Use 0 if you want to open window with a HTML content from the file inside, or use 1 if you want to open window with an iFrame inside.

So you can call this function manually from any place of your site. For example, the following function will open new window with a HTML content from the content/about.php file, with "About my Company" window title, width of the window will be 600px in width:

newWindow("content/about.html","About my Company",600,0)

\$uri="content/about.html"

\$title="About my Company"

\$width=600

\$iframe=0

8. Window Template (HTML & JavaScript)

Window template file (*window.tmpl.php*) is located in the root folder of web-site. It consists of three parts. The first part is the clean php script, second is the inline window script and the last one is the HTML layout of window. Let's look at these parts more closely.

8.1 PHP Script

This php code is used to get the window's parameters, which were sent with the *newWindow()* JavaScript function described above.

```
<?php
$id = $_GET['id'];
$uri = $_GET['uri'];
if (!empty($_GET['title'])) $title = '<h3 class="window-title">'. $_GET['title']. '</h3>';
else $title="";
if (!empty($_GET['width'])) $width = "width:". $_GET['width']. "px";
else $width="";
$iframe = $_GET['iframe'];
?>
```

If the **width** parameter is not empty, \$width variable will be equal to this. Otherwise it will be empty

If the **title** parameter is not empty, \$title variable will be equal to this. Otherwise it will be empty

8.2 Inline JavaScript

```
<script type="text/javascript">
$(function() {
    var $id=<?php echo $id ?>;
    $(".in-window").removeClass("in-window").addClass("in-window-"+$id);
    $(".in-window-"+$id).click(function(e) {
        e.preventDefault();
        newWindow($(this).attr('href'), $(this).attr('title'), $(this).attr("rel"), 0)
    })
    $(".in-iframe").removeClass("in-iframe").addClass("in-iframe-"+$id);
    $(".in-iframe-"+$id).click(function(e) {
        e.preventDefault();
        newWindow($(this).attr('href'), $(this).attr('title'), $(this).attr("rel"), 1)
    })
    $("#window_"+$id).resizable({ alsoResize: '#window_'+$id+' .w-m1, #window_'+$id+' iframe',
minHeight: 50 });
    $(".window").draggable({ handle: 'div.w-tbl' });
    $(".window").mousedown(function() {
        $(this).css({"z-index":3000}).fadeTo(300,1)
        $(this).nextAll(".window").css({"z-index":2000}).fadeTo(300,0.3)
        $(this).prevAll(".window").css({"z-index":2000}).fadeTo(300,0.3)
    })
    $(".img.close").click(function() {
        $(this).closest(".window").fadeOut(300, function() {
            $(this).remove()
        })
    })
})
</script>
```

Initialization of the \$id variable

Used to replace the **in-window** class with, for example, **in-window-3** class (if \$id =3)

The same handling as for the **"new-window"** class in the main script

Used to replace the **in-iframe** class with, for example, **in-iframe-3** class (if \$id =3)

The same handling as for the **"iframe"** class in the main script

Provides an opportunity to resize window

Provides an opportunity to drag window

When you'll click the not active (faded) window, it will fade to opacity 1 for 0.3 seconds and all other windows will fade to opacity 0.3 for 0.3 seconds

This function provides an opportunity to close a window with a fade effect, after the **close** button is clicked

This function provides an opportunity to minimize window through the number of animation effects. And after the window is minimized, the “minimize” button will be replaced with a “expand” button.

```
$( "img.minimize" ).click(function() {
    var $window = $(this).closest(".window");
    $window.animate({width:"200px",height:"50px",left:"10px",},300,function(){
        $(this).children(".w-ml").css({height:"10px",width:"198px"})
        $(this).find("iframe").css({height:"120px",width:"100%"})
        $(this).addClass("minimized")
        $(this).fadeTo(300,0.3)
        $(this).resizable("disable")
    })
    $(this).after('');
    $(this).remove();
})
})
</script>
```

8.3 Window HTML Layout

Now let's look at the HTML code of the window:

If window width parameter is not empty, then the its value will be added here

Window **ID** will be added here

```
<div class="window" style="display:none;z-index:4000;<?php echo $width ?>" id="window_<?php
echo $id ?>">
    <div class="w-tbl">
        <div class="w-tbr">
            <div class="w-tbm">
                <?php echo $title?>
                
                 </div>
            </div>
        </div>
        <div class="w-ml">
            <div class="w-mr">
                <div class="w-mm">
                    <div class="w-content">
                        <?php
                        if ($iframe == 0 ) include($uri) ;
                        if ($iframe == 1) { ?>
                        <iframe frameborder="0" src="<?php echo $uri ?>" width="100%" height="350"></iframe>
                        <?php }?>
                    </div>
                </div>
            </div>
        </div>
        <div class="w-bot">
            <div class="w-bl">
                <div class="w-br">
                    <div class="w-bm"></div>
                </div>
            </div>
        </div>
    </div>
```

Top container for window **title** and toolbar with **minimize**, **expand** and **close** buttons.

If the **\$iframe** parameter is equal to 0, then here will be placed HTML content of the file, specified by **newWindow()** function. And if the **\$iframe** is equal to 1, then the iframe will be added here with the **src** attribute equal to the **\$uri** variable.

9. Ajax Contact Form

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

When we'll click the "Send Message" button, the `sendMessage()` JavaScript function will be executed.

```
<h2 class="content-heading">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" onblur="validateText($(this).attr('id'))" type="text" class=
"text" name="name" size="30" />
  </label>
</p>
<p>
  <label> Your Email:*<br>
  <input id="contact_email" onblur="validateEmail()" type="text" class="text" name="email"
size="30" />
</label>
</p>
<p>
  <label> Subject:*<br>
  <input id="contact_subject" onblur="validateText($(this).attr('id'))" style="color:#777"
type="text" class="text" name="subject" size="30" value="Message From O.S Template" readonly=
"readonly" />
</label>
</p>
<p>
  <label> Message:*<br>
  <textarea id="contact_message" onblur="validateMessage()" name="message" class=
"textarea" cols="30" rows="10"></textarea>
</label>
</p>
<span class="button-l"><span class="button-m">
  <input type="submit" name="Submit" value="Send Message" />
</span></span>
</form>
```

Remove the **readonly** attribute.
It's used in the demo site.

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 then will return mailing 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

Standard headers for the e-mail message.

Otherwise script will return the "0" value

If the message was successfully sent, the script will return the "1" value

After the mail is sent (or the error is occurred), JavaScript will open the new window with a message status inside. You can edit these statuses inside the appropriate .php files located in the **includes/** folder:

- **message_error.php**
- **message_sent.php**
- **server_error.php**

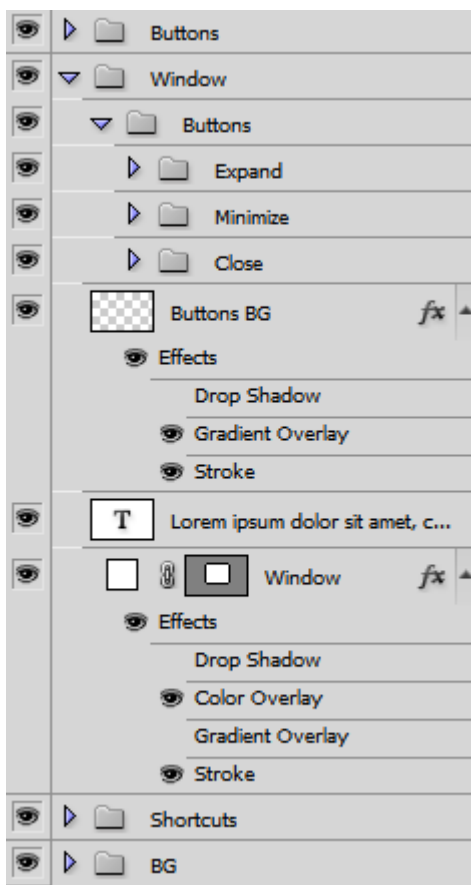
These files contain simple HTML formatting messages. So you'll change them with no problem.

10. PSD Files & Structure

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

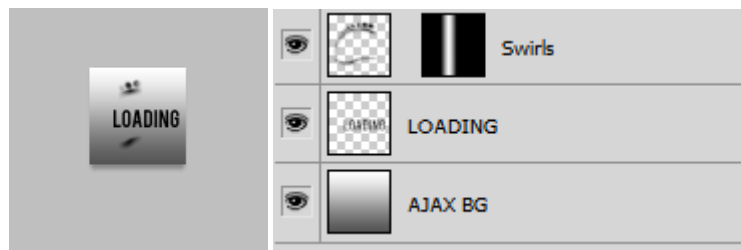


First .psd file is the **os-layouts.psd**. It contains all general layouts for template styling:



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

Second .psd file is the **ajax-loader.psd**. This file can be used to redesign animated Ajax Loader:



11. 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 **O.S Template** (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**.