iDANGERO.US PREMIUM SITE TEMPLATES SCRIPTS GRAPHICS

Imagine Box

Imagine Box is a premium site template, designed by the Dangero.us team. This template is perfect for small business websites, projects, or it can be used as a fashion site.

The template is fully animated without the use of Flash technology. All content loaded on the fly, so your users do not need to wait the reloading of entire site.

This template is easy to redesign, very simple and clear API. Buying this template you will also get .PSD sources and detailed documentation.

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

So, it's time to checkout the main pages of the site:

- Template Features
- Products section
- Ajax Contact Form with validation

© 2010 "Imagine Box" Premium Template by iDangero.us. All Rights Reserved

"Imagine Box"

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 "Imagine Box"

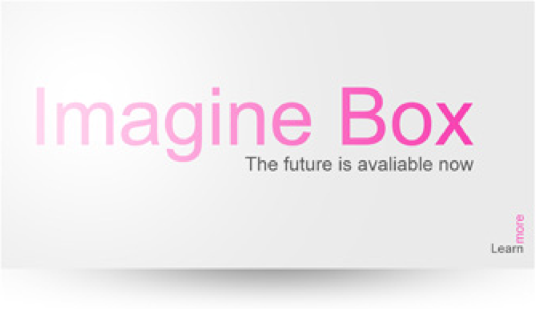
Imagine Box is the Ajax based website template designed by iDangero.us team.

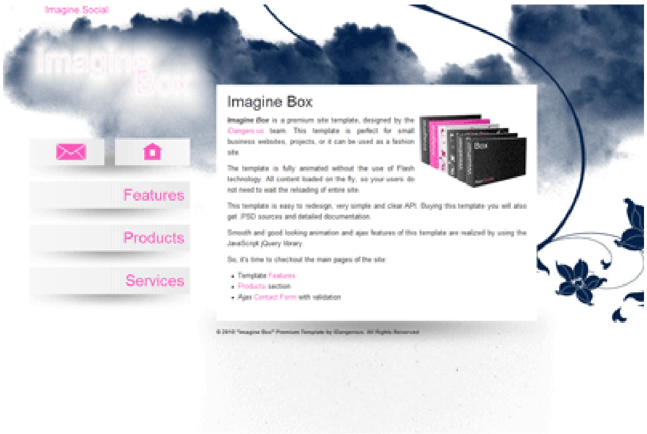
This template is perfect for small business websites, projects, or it can be used as a fashion site. The template is fully animated without the use of Flash technology. Smooth and good looking animation and Ajax features of this template are realized by using the JavaScript jQuery library.

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

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

Also **Imagine Box** is powered with an Ajax Contact Form with validation. Here is a couple of screenshots:





Imagine Box

Imagine Box is a premium 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 template is fully animated without the use of Flash technology. All content loaded on the fly, so your users do not need to wait the reloading of entire site.

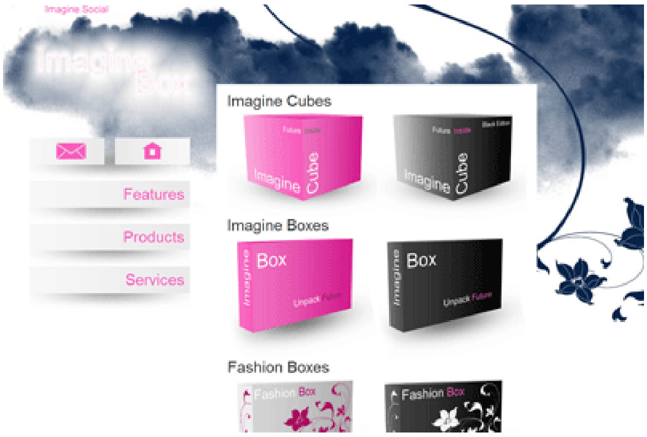
This template is easy to redesign, very simple and clear API. Buying this template you will also get .PSD sources and detailed documentation.

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

So, it's time to checkout the main pages of the site:

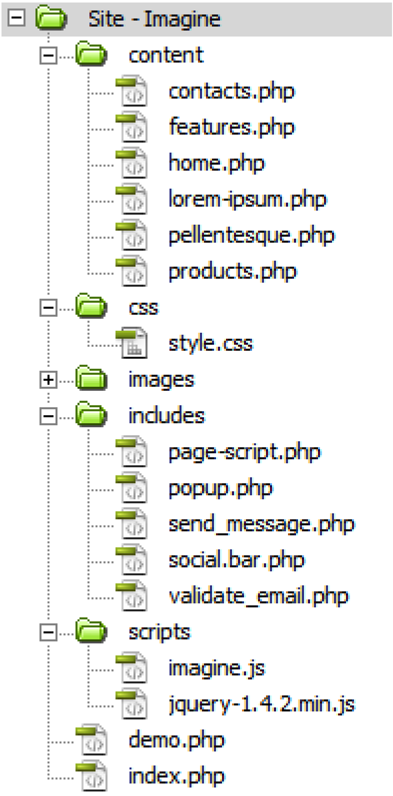
- Template Features
- Products section
- Ajax Contact Form with validation

© 2010 "Imagine Box" Premium Template by iDangero.us. All Rights Reserved



2. Imagine Box Files Structure

Imagine Box files are located in the **imagine/** folder from the downloaded archive:



```
Site - Imagine
  content
    contacts.php
    features.php
    home.php
    lorem-ipsum.php
    pellentesque.php
    products.php
  css
    style.css
  images
  includes
    page-script.php
    popup.php
    send_message.php
    social.bar.php
    validate_email.php
  scripts
    imagine.js
    jquery-1.4.2.min.js
  demo.php
  index.php
```

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: two php scripts used for Ajax contact form, social.bar, page script and popup window layout

scripts: JavaScripts used in this template

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

3. Installation

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

Index.php <body> structure:

```
<body>
<!------ Intro Section ------>
<div id="intro">
  <div class="intro-box">
     <img src="images/layouts/intro-box.png" width="1000" height="598" alt="Imagine Box" />
     <img class="blink" id="enter" src="images/layouts/learn-more.png" width="60"
     height="71" alt="learn more" title="Learn More" />
  </div>
</div>
<!------ Main Content Section ------>
<div id="main-content">
    <!------ Social Bar ------>
```

Social.Bar section (chapter 11)

```
    <div id="social-bar">
        <?php include("includes/social.bar.php")?>
    </div>
    <!------ Menu ------>
    <div id="menu">
      <ul class="menu">
        <li class="home-contacts">
           <a class="menu-home" href="content/home.php">
              <img src="images/layouts/home.png" alt="Home" title="Home" />
           </a>
           <a class="menu-contacts" href="content/contacts.php">
              <img src="images/layouts/contacts.png" alt="Contacts" title="Contacts" />
           </a>
        </li>
        <li><a href="content/features.php">Features</a></li>
        <li><a href="content/products.php">Products</a></li>
        <li>Services
            <ul class="submenu">
                <li><a href="content/lorem-ipsum.php">Lorem Ipsum</a></li>
                <li><a href="content/pellentesque.php">Pellentesque</a></li>
            </ul>
        </li>
      </ul>
    </div>
    <!------ Content Pages Section ------>
```

Content Pages section. All files with html content will be loaded here

```
    <div id="content">
        <div class="content-inner">
           <?php include('includes/page-script.php')?>
           <?php include('content/home.php')?>
        </div>
    </div>
    <div class="clear"></div>
    <!------ Footer ------>
    <div id="footer"> --- Footer Content ---</div>
</div>
<!------ Additional Background ------>
<div id="background" ></div>
<!------ Popup Layouts ------>
```

Popup layouts in chapter 9

```
<?php include("includes/popup.php")?>
</body>
```

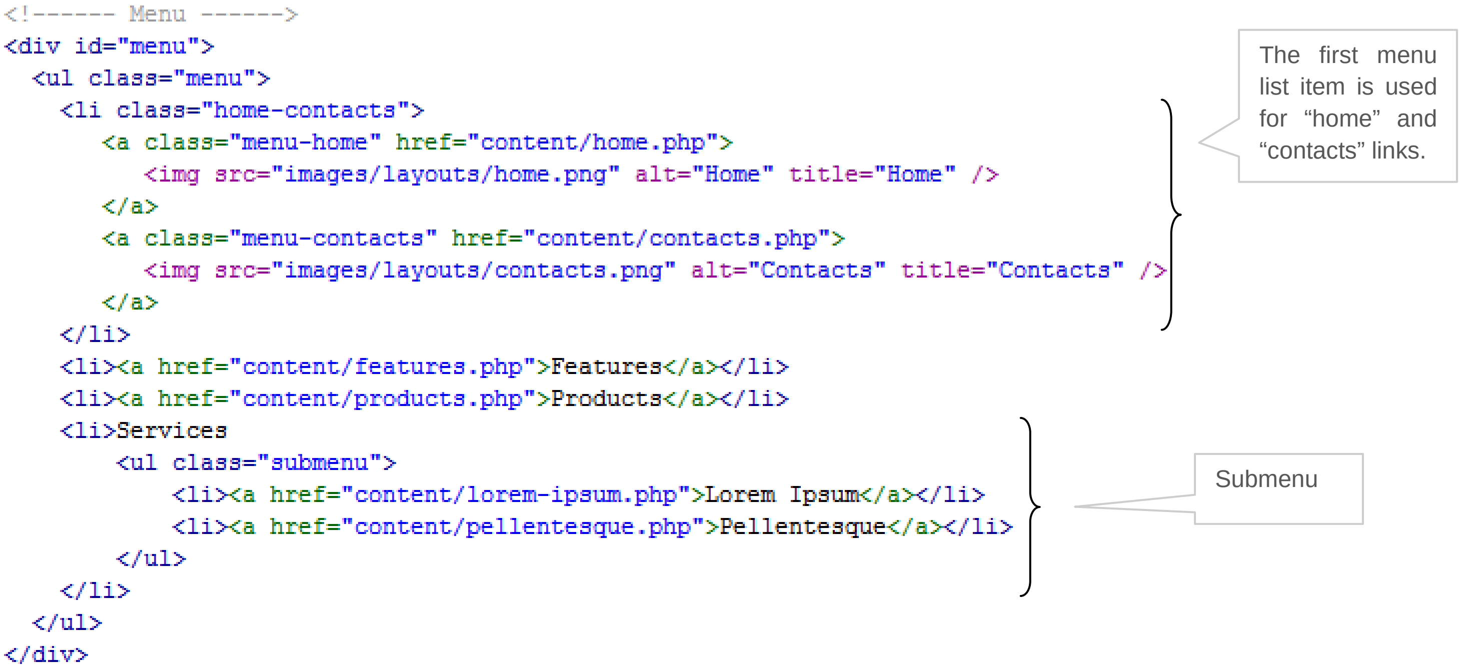
Intro Section

This section contains a large image with the “Imagine Box” title. We see this image when the site is loaded and all site content is hidden behind it.

Main Content Section

This is the main section of the site. Here is social bar, menu, content section and the footer section.

Menu

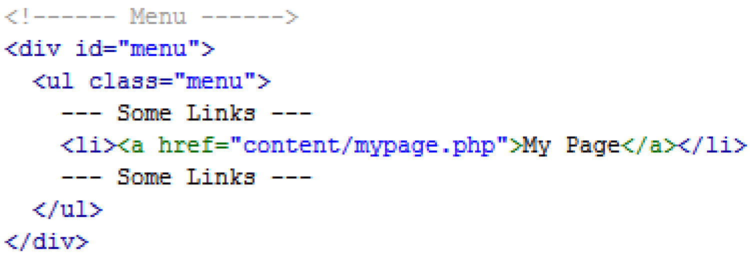


```
<!------ Menu ------>
<div id="menu">
  <ul class="menu">
    <li class="home-contacts">
       <a class="menu-home" href="content/home.php">
          <img src="images/layouts/home.png" alt="Home" title="Home" />
       </a>
       <a class="menu-contacts" href="content/contacts.php">
          <img src="images/layouts/contacts.png" alt="Contacts" title="Contacts" />
       </a>
    </li>
    <li><a href="content/features.php">Features</a></li>
    <li><a href="content/products.php">Products</a></li>
    <li>Services
        <ul class="submenu">
            <li><a href="content/lorem-ipsum.php">Lorem Ipsum</a></li>
            <li><a href="content/pellentesque.php">Pellentesque</a></li>
        </ul>
    </li>
  </ul>
</div>
```

The first menu list item is used for “home” and “contacts” links.

Submenu

To create your own menu link, first of all you need to create list item using <li> tag inside the list (<ul>) with a “menu” class. Then add link with an <a> tag. After that you have to create a .php file with HTML content in the **content/** folder. Name it for example **mypage.php.** Then add the **href** attribute with a path to this file. It must be look like this:

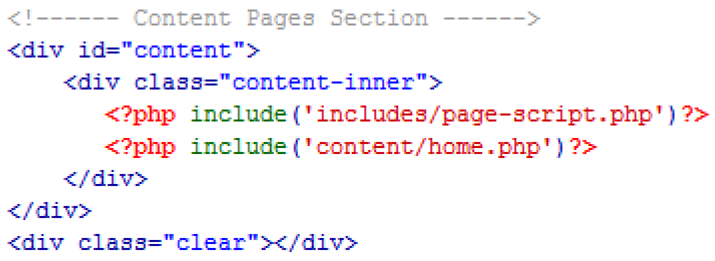


```
<!------ Menu ------>
<div id="menu">
  <ul class="menu">
    --- Some Links ---
    <li><a href="content/mypage.php">My Page</a></li>
    --- Some Links ---
  </ul>
</div>
```

To create submenu you have to add unsorted list using the <ul> tag with a “submenu” class inside the menu list item, and then you can add menu links inside it.

Content Section

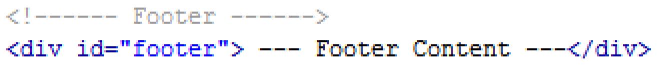
This section is used to load HTML content (by Ajax) from the .php files located in **content/** folder. Home page content and page script is included by default.



```
<!------ Content Pages Section ------>
<div id="content">
    <div class="content-inner">
       <?php include('includes/page-script.php')?>
       <?php include('content/home.php')?>
    </div>
</div>
<div class="clear"></div>
```

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**

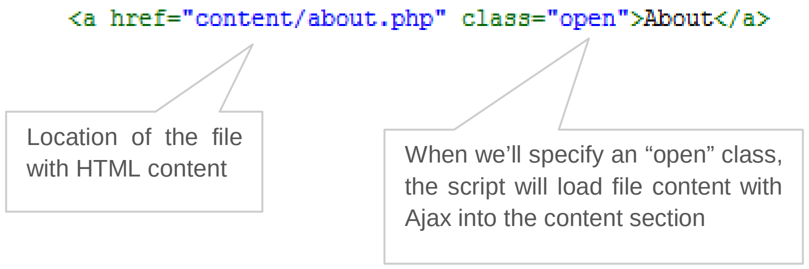
5. Inner Linking (How it works?)

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="content/about.php" class="open">About</a>
```

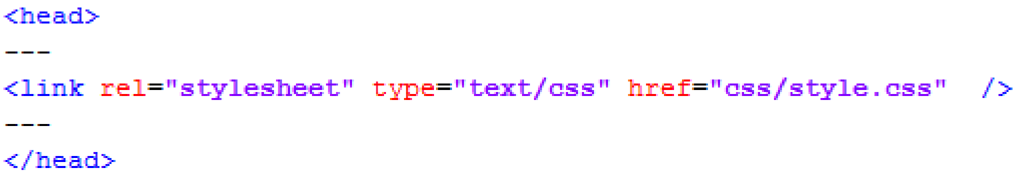
Location of the file with HTML content

When we’ll specify an “open” 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 file location using the **href** attribute and to specify the “**open**” class to load page with Ajax.

6. CSS Files and Structure

There is only 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/style.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.

`style.css` file is separated into sections:

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

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

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

This section contains rules for menu styling

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

Rules for content blocks

```
/* ---- Social.Bar ---- */
```

Social.Bar icons styling

```
/*---- Products Section ----*/
```

All rules for the Products Section

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

Rules for contact form and buttons

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

Rules for footer styling

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

Ajax Loader styling

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

Popup Window and its parts

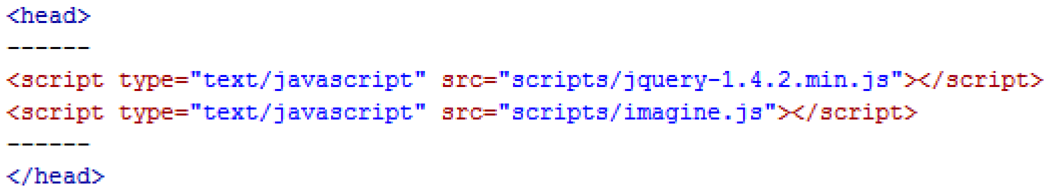
```
/*----End of CSS File----*/
```

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.

7. JavaScripts

Imagine Box template uses three JavaScript files and one inline script for the Products Section (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/imagine.js"></script>
------
</head>
```

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

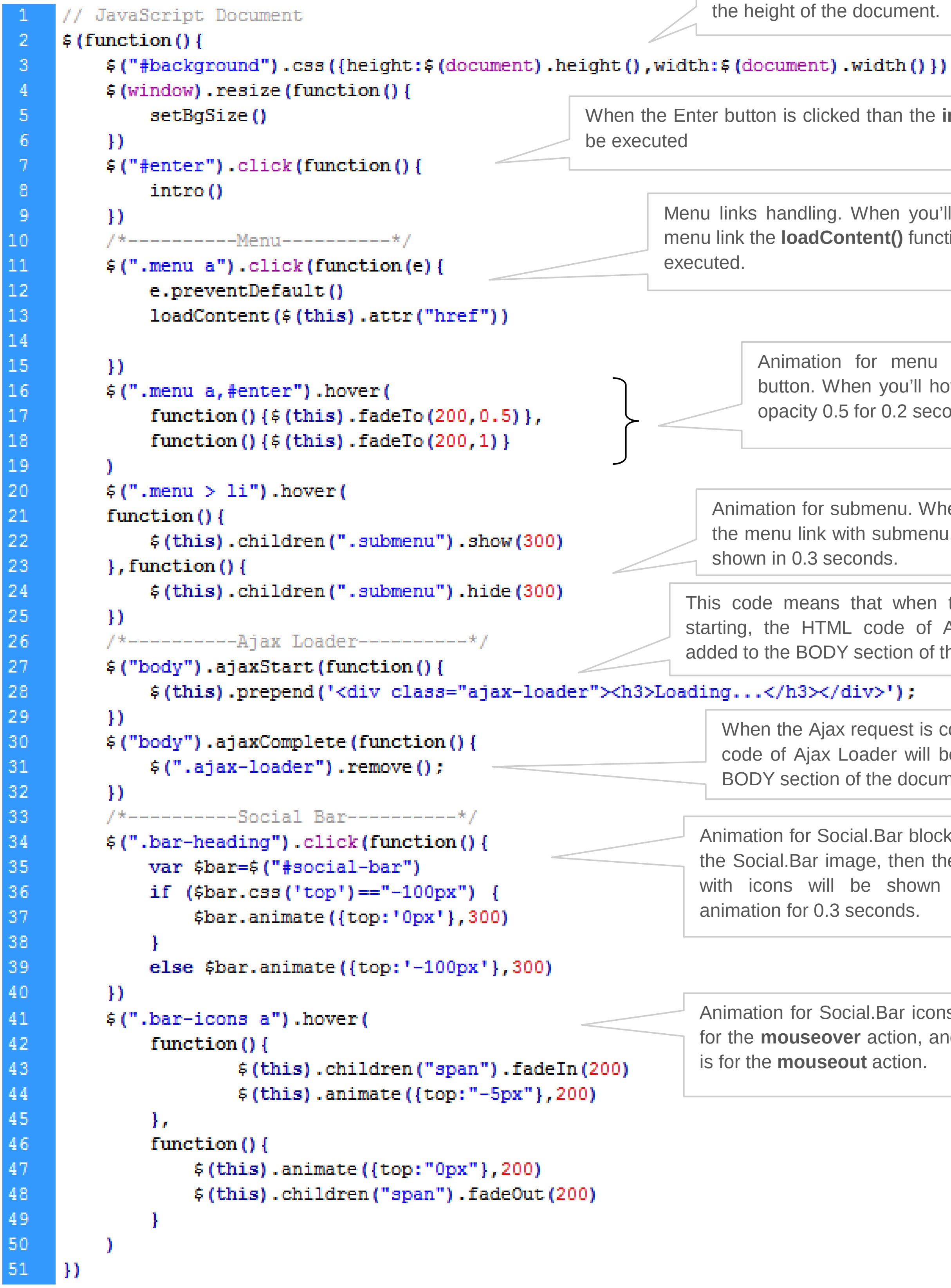
The first script file is the jQuery (JavaScript library that greatly reduces the amount of code that you must write).

The second script file (imagine.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.

The third script is the so called “page script” is loaded with every page.

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 is the effects and rules for handling menu links, animation of Social.Bar and Ajax loader.



```
// JavaScript Document
$(function(){
    $("#background").css({height:$(document).height(),width:$(document).width()})
    $(window).resize(function(){
        setBgSize()
    })
    $("#enter").click(function(){
        intro()
    })
    /*----------Menu----------*/
    $(".menu a").click(function(e){
        e.preventDefault()
        loadContent($(this).attr("href"))

    })
    $(".menu a,#enter").hover(
        function(){$(this).fadeTo(200,0.5)},
        function(){$(this).fadeTo(200,1)}
    )
    $(".menu > li").hover(
    function(){
        $(this).children(".submenu").show(300)
    },function(){
        $(this).children(".submenu").hide(300)
    })
    /*----------Ajax Loader----------*/
    $("body").ajaxStart(function(){
        $(this).prepend('<div class="ajax-loader"><h3>Loading...</h3></div>');
    })
    $("body").ajaxComplete(function(){
        $(".ajax-loader").remove();
    })
    /*----------Social Bar----------*/
    $(".bar-heading").click(function(){
        var $bar=$("#social-bar")
        if ($bar.css('top')=="-100px") {
            $bar.animate({top:'0px'},300)
        }
        else $bar.animate({top:'-100px'},300)
    })
    $(".bar-icons a").hover(
        function(){
                $(this).children("span").fadeIn(200)
                $(this).animate({top:"-5px"},200)
        },
        function(){
            $(this).animate({top:"0px"},200)
            $(this).children("span").fadeOut(200)
        }
    )
})
```

Assigning the background height equal to the height of the document.

When the Enter button is clicked than the **intro()** function will be executed

Menu links handling. When you'll click the menu link the **loadContent()** function will be executed.

Animation for menu links and the "enter" button. When you'll hover on it, it will fade to opacity 0.5 for 0.2 seconds

Animation for submenu. When you'll hover on the menu link with submenu, submenu will be shown in 0.3 seconds.

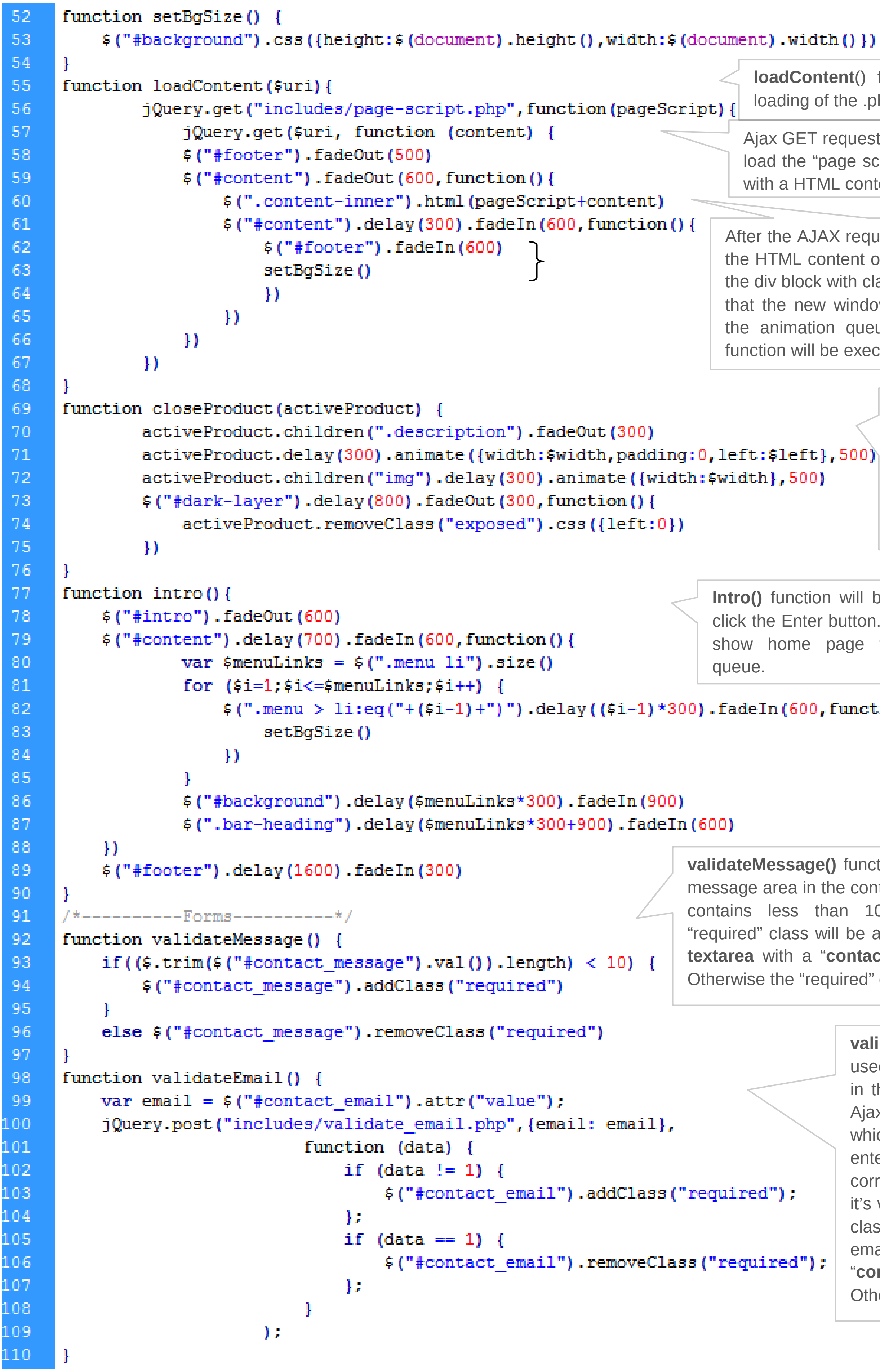
This code means that when the Ajax request is starting, the HTML code of Ajax Loader will be added to the BODY section of the document.

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

Animation for Social.Bar block. When we'll click the Social.Bar image, then the social bar block with icons will be shown using the slide animation for 0.3 seconds.

Animation for Social.Bar icons. First function is for the **mouseover** action, and the second one is for the **mouseout** action.

The next part of code contains functions which could be called from any place of the document. These functions are used for Ajax requests, for submitting and validation of the contact form, for products section, for popup window.



```
function setBgSize() {
    $("#background").css({height:$(document).height(),width:$(document).width()})
}
function loadContent($uri){
        jQuery.get("includes/page-script.php",function(pageScript){
            jQuery.get($uri, function (content) {
            $("#footer").fadeOut(500)
            $("#content").fadeOut(600,function(){
                $(".content-inner").html(pageScript+content)
                $("#content").delay(300).fadeIn(600,function(){
                    $("#footer").fadeIn(600)
                    setBgSize()
                    })
                })
            })
        })
}
function closeProduct(activeProduct) {
        activeProduct.children(".description").fadeOut(300)
        activeProduct.delay(300).animate({width:$width,padding:0,left:$left},500)
        activeProduct.children("img").delay(300).animate({width:$width},500)
        $("#dark-layer").delay(800).fadeOut(300,function(){
            activeProduct.removeClass("exposed").css({left:0})
        })
}
function intro(){
    $("#intro").fadeOut(600)
    $("#content").delay(700).fadeIn(600,function(){
            var $menuLinks = $(".menu li").size()
            for ($i=1;$i<=$menuLinks;$i++) {
                $(".menu > li:eq("+($i-1)+")").delay(($i-1)*300).fadeIn(600,function(){
                    setBgSize()
                })
            }
            $("#background").delay($menuLinks*300).fadeIn(900)
            $(".bar-heading").delay($menuLinks*300+900).fadeIn(600)
    })
    $("#footer").delay(1600).fadeIn(300)
}
/*----------Forms----------*/
function validateMessage() {
    if(($.trim($("#contact_message").val()).length) < 10) {
        $("#contact_message").addClass("required")
    }
    else $("#contact_message").removeClass("required")
}
function validateEmail() {
    var email = $("#contact_email").attr("value");
    jQuery.post("includes/validate_email.php",{email: email},
                        function (data) {
                            if (data != 1) {
                                $("#contact_email").addClass("required");
                            };
                            if (data == 1) {
                                $("#contact_email").removeClass("required");
                            };
                        }
                    );
}
```

loadContent() function is used for Ajax loading of the .php files with HTML content.

Ajax GET request. First of all, the script will load the “page script” file, then the php file with a HTML content

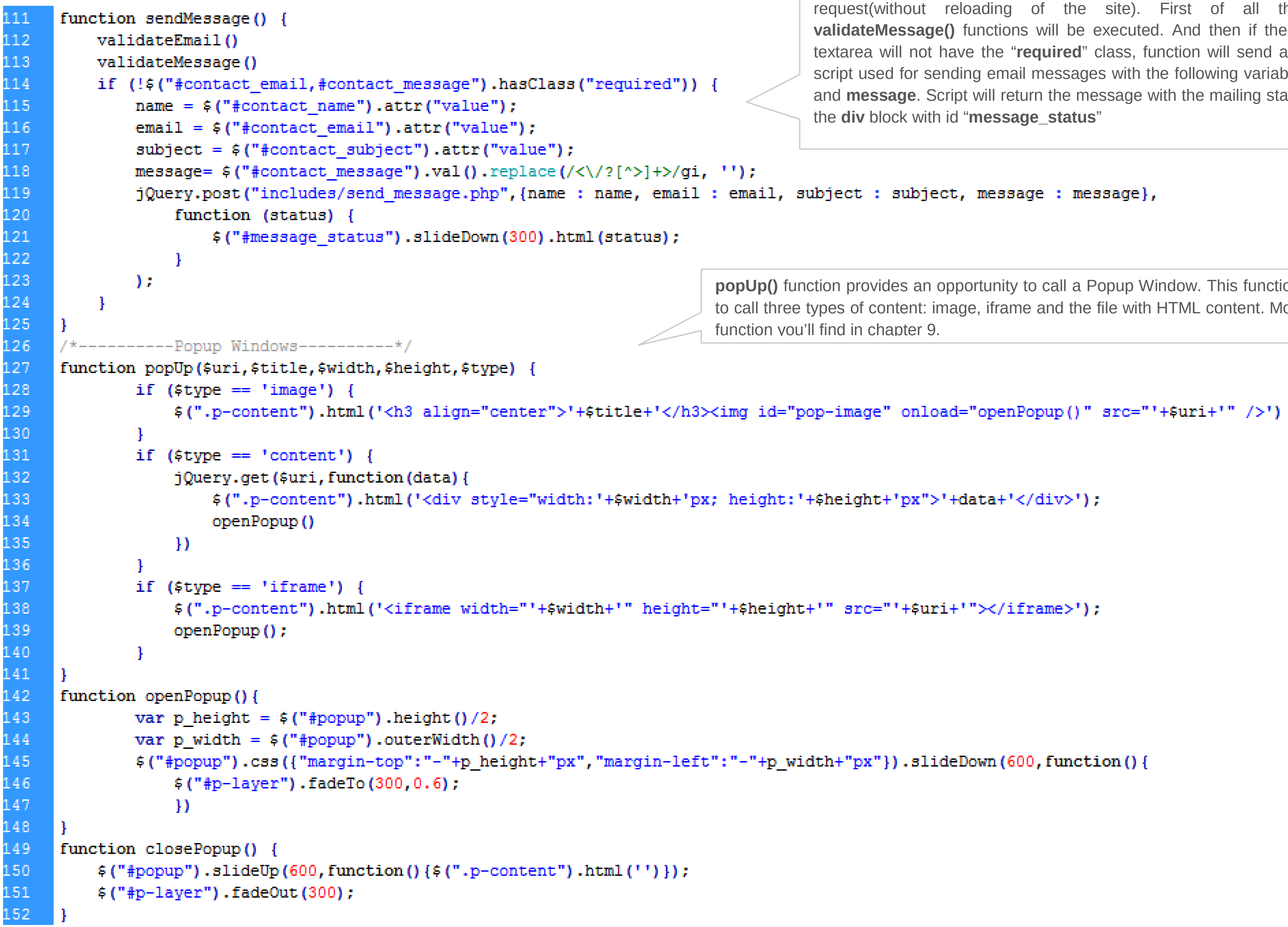
After the AJAX request, the “page script” and the HTML content of the file will be loaded to the div block with class “**content-inner**”. After that the new window will be shown through the animation queue. And the setBgSize() function will be executed.

closeProduct () function is used to close the exposed product through the animation queue

Intro() function will be executed when you’ll click the Enter button. This function is used to show home page through the animation queue.

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.

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.

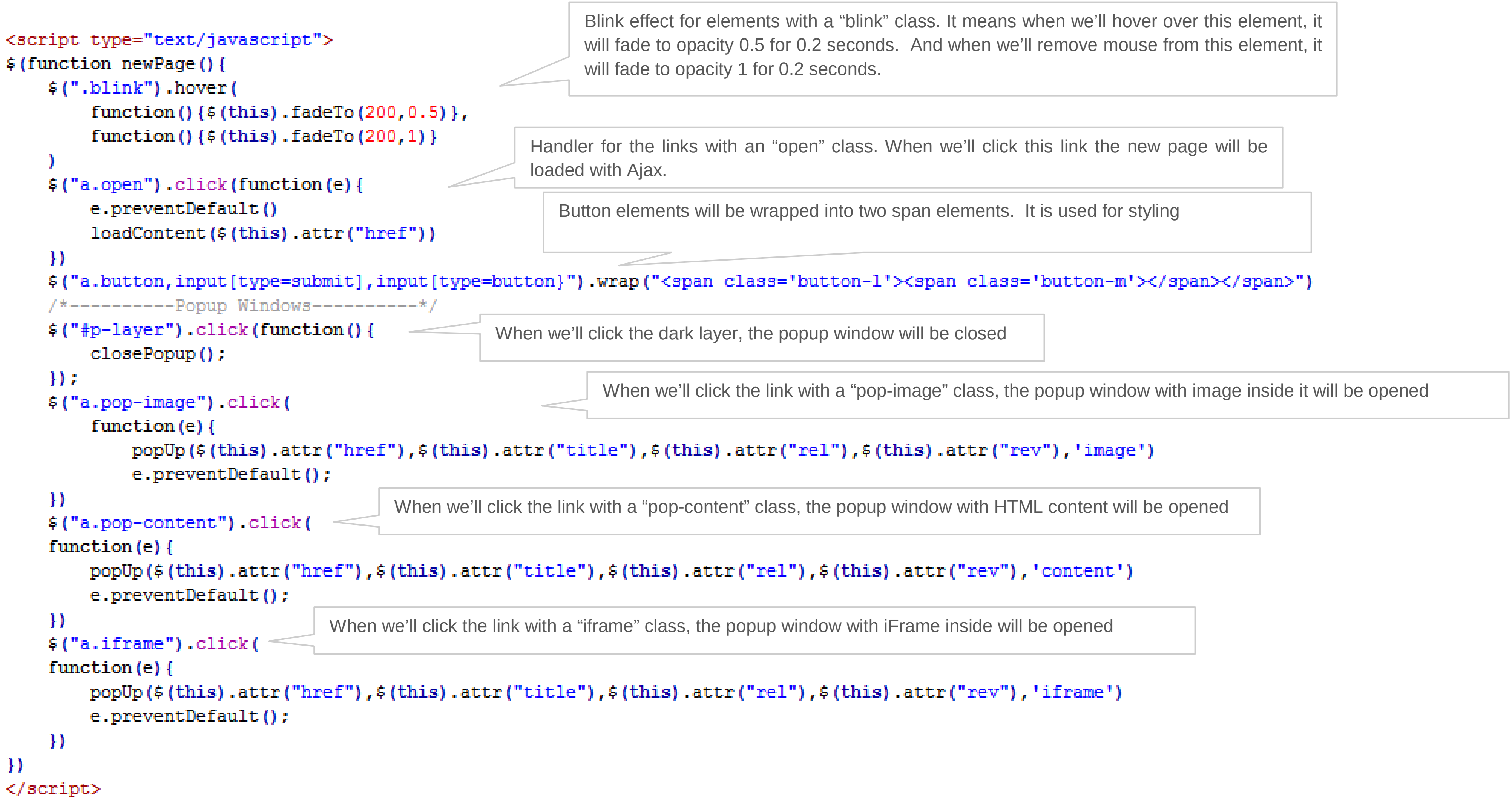


```
function sendMessage() {
    validateEmail()
    validateMessage()
    if (!$("#contact_email,#contact_message").hasClass("required")) {
        name = $("#contact_name").attr("value");
        email = $("#contact_email").attr("value");
        subject = $("#contact_subject").attr("value");
        message= $("#contact_message").val().replace(/<\/?[^>]+>/gi, '');
        jQuery.post("includes/send_message.php",{name : name, email : email, subject : subject, message : message},
            function (status) {
                $("#message_status").slideDown(300).html(status);
            }
        );
    }
}
/*----------Popup Windows----------*/
function popUp($uri,$title,$width,$height,$type) {
        if ($type == 'image') {
            $(".p-content").html('<h3 align="center">'+$title+'</h3><img id="pop-image" onload="openPopup()" src="'+$uri+'" />')
        }
        if ($type == 'content') {
            jQuery.get($uri,function(data){
                $(".p-content").html('<div style="width:'+$width+'px; height:'+$height+'px">'+data+'</div>');
                openPopup()
            })
        }
        if ($type == 'iframe') {
            $(".p-content").html('<iframe width="'+$width+'" height="'+$height+'" src="'+$uri+'"></iframe>');
            openPopup();
        }
}
function openPopup(){
        var p_height = $("#popup").height()/2;
        var p_width = $("#popup").outerWidth()/2;
        $("#popup").css({"margin-top":"-"+p_height+"px","margin-left":"-"+p_width+"px"}).slideDown(600,function(){
            $("#p-layer").fadeTo(300,0.6);
            })
}
function closePopup() {
    $("#popup").slideUp(600,function(){$(".p-content").html('')});
    $("#p-layer").fadeOut(300);
}
```

sendMessage() function provides an opportunity to send e-mail message by the Ajax request(without reloading of the site). First of all the **validateEmail()** and **validateMessage()** functions will be executed. And then if the email field and message textarea will not have the “**required**” class, function will send an Ajax request to the php script used for sending email messages with the following variables: **name**, **email**, **subject** and **message**. Script will return the message with the mailing status, which will be added to the **div** block with id “**message_status**”

popUp() function provides an opportunity to call a Popup Window. This function allows you to call three types of content: image, iframe and the file with HTML content. More about this function you’ll find in chapter 9.

Now let’s look at the page script located in the **includes/page-script.php** file:



```
<script type="text/javascript">
$(function newPage(){
    $(".blink").hover(
        function(){$(this).fadeTo(200,0.5)},
        function(){$(this).fadeTo(200,1)}
    )
    $("a.open").click(function(e){
        e.preventDefault()
        loadContent($(this).attr("href"))
    })
    $("a.button,input[type=submit],input[type=button}").wrap("<span class='button-l'><span class='button-m'></span></span>")
    /*----------Popup Windows----------*/
    $("#p-layer").click(function(){
        closePopup();
    });
    $("a.pop-image").click(
        function(e){
            popUp($(this).attr("href"),$(this).attr("title"),$(this).attr("rel"),$(this).attr("rev"),'image')
            e.preventDefault();
    })
    $("a.pop-content").click(
    function(e){
        popUp($(this).attr("href"),$(this).attr("title"),$(this).attr("rel"),$(this).attr("rev"),'content')
        e.preventDefault();
    })
    $("a.iframe").click(
    function(e){
        popUp($(this).attr("href"),$(this).attr("title"),$(this).attr("rel"),$(this).attr("rev"),'iframe')
        e.preventDefault();
    })
})
</script>
```

Blink effect for elements with a “blink” class. It means when we’ll hover over this element, it will fade to opacity 0.5 for 0.2 seconds. And when we’ll remove mouse from this element, it will fade to opacity 1 for 0.2 seconds.

Handler for the links with an “open” class. When we’ll click this link the new page will be loaded with Ajax.

Button elements will be wrapped into two span elements. It is used for styling

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

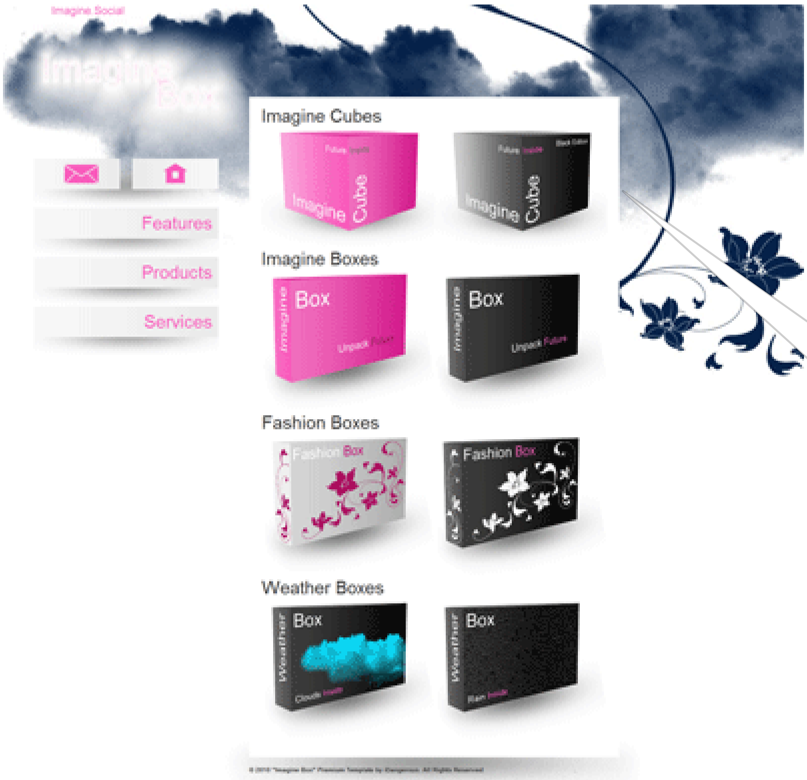
When we’ll click the link with a “pop-image” class, the popup window with image inside it will be opened

When we’ll click the link with a “pop-content” class, the popup window with HTML content will be opened

When we’ll click the link with a “iframe” class, the popup window with iFrame inside will be opened

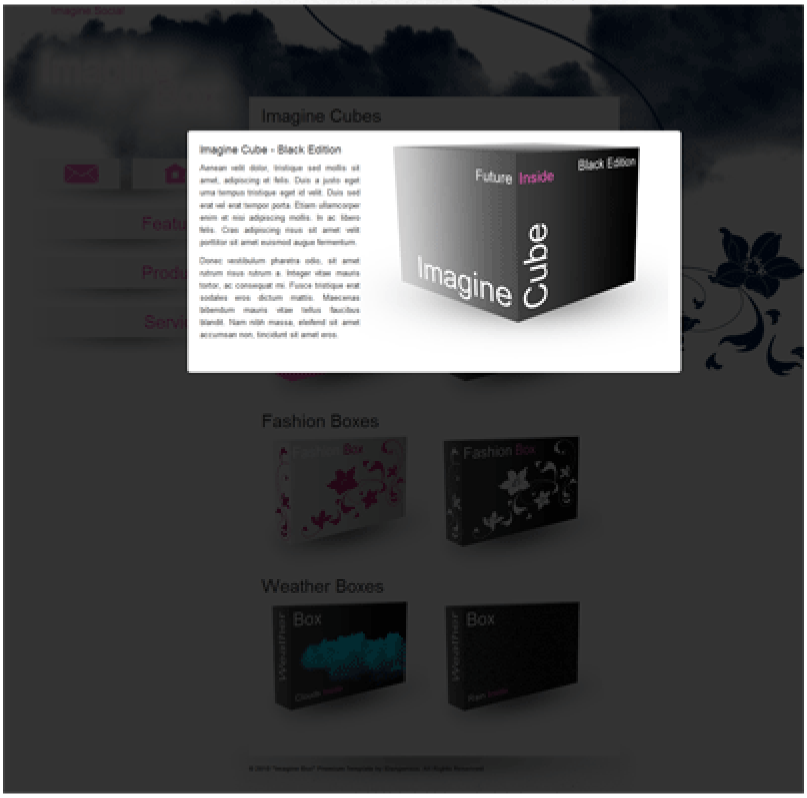
8. Products Section (HTML & JavaScript)

Here is how the Products Section looks:



Products Page

Click the product and you'll get …



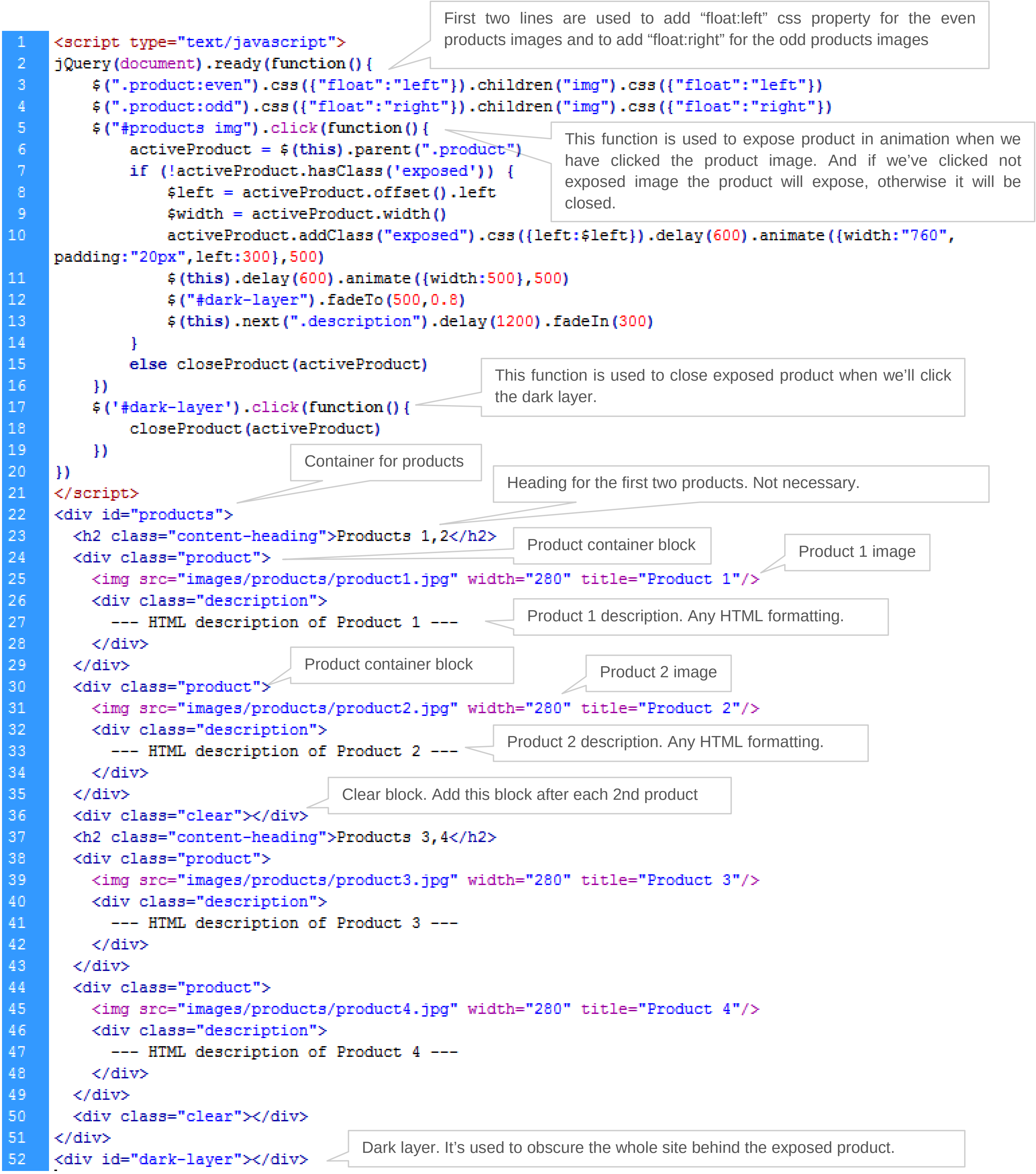
Imagine Cube - Black Edition

Aenean velit dolor, tristique sed mollis sit amet, adipiscing et felis. Duis a justo eget urna tempus tristique eget id velit. Duis sed erat vel erat tempor porta. Etiam ullamcorper enim et nisi adipiscing mollis. In ac libero felis. Cras adipiscing risus sit amet velit porttitor sit amet euismod augue fermentum.

Donec vestibulum pharetra odio, sit amet rutrum risus rutrum a. Integer vitae mauris tortor, ac consequat mi. Fusce tristique erat sodales eros dictum mattis. Maecenas bibendum mauris vitae tellus faucibus blandit. Nam nibh massa, eleifend sit amet accumsan non, tincidunt sit amet eros.

…exposed product with its description

Let's see how it's works and how to create your own products:



```
<script type="text/javascript">
jQuery(document).ready(function(){
    $(".product:even").css({"float":"left"}).children("img").css({"float":"left"})
    $(".product:odd").css({"float":"right"}).children("img").css({"float":"right"})
    $("#products img").click(function(){
        activeProduct = $(this).parent(".product")
        if (!activeProduct.hasClass('exposed')) {
            $left = activeProduct.offset().left
            $width = activeProduct.width()
            activeProduct.addClass("exposed").css({left:$left}).delay(600).animate({width:"760", padding:"20px",left:300},500)
            $(this).delay(600).animate({width:500},500)
            $("#dark-layer").fadeTo(500,0.8)
            $(this).next(".description").delay(1200).fadeIn(300)
        }
        else closeProduct(activeProduct)
    })
    $('#dark-layer').click(function(){
        closeProduct(activeProduct)
    })
})
</script>
<div id="products">
  <h2 class="content-heading">Products 1,2</h2>
  <div class="product">
    <img src="images/products/product1.jpg" width="280" title="Product 1"/>
    <div class="description">
      --- HTML description of Product 1 ---
    </div>
  </div>
  <div class="product">
    <img src="images/products/product2.jpg" width="280" title="Product 2"/>
    <div class="description">
      --- HTML description of Product 2 ---
    </div>
  </div>
  <div class="clear"></div>
  <h2 class="content-heading">Products 3,4</h2>
  <div class="product">
    <img src="images/products/product3.jpg" width="280" title="Product 3"/>
    <div class="description">
      --- HTML description of Product 3 ---
    </div>
  </div>
  <div class="product">
    <img src="images/products/product4.jpg" width="280" title="Product 4"/>
    <div class="description">
      --- HTML description of Product 4 ---
    </div>
  </div>
  <div class="clear"></div>
</div>
<div id="dark-layer"></div>
```

First two lines are used to add “float:left” css property for the even products images and to add “float:right” for the odd products images

This function is used to expose product in animation when we have clicked the product image. And if we’ve clicked not exposed image the product will expose, otherwise it will be closed.

This function is used to close exposed product when we’ll click the dark layer.

Container for products

Heading for the first two products. Not necessary.

Product container block

Product 1 image

Product 1 description. Any HTML formatting.

Product container block

Product 2 image

Product 2 description. Any HTML formatting.

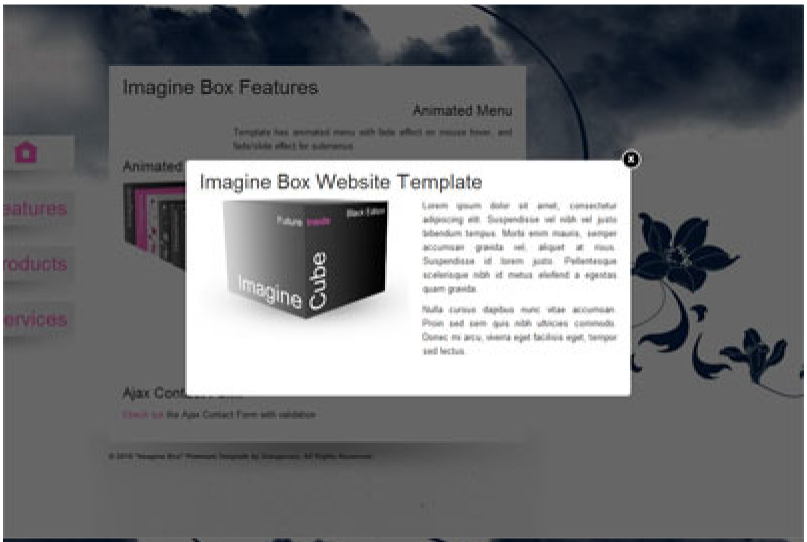
Clear block. Add this block after each 2nd product

Dark layer. It’s used to obscure the whole site behind the exposed product.

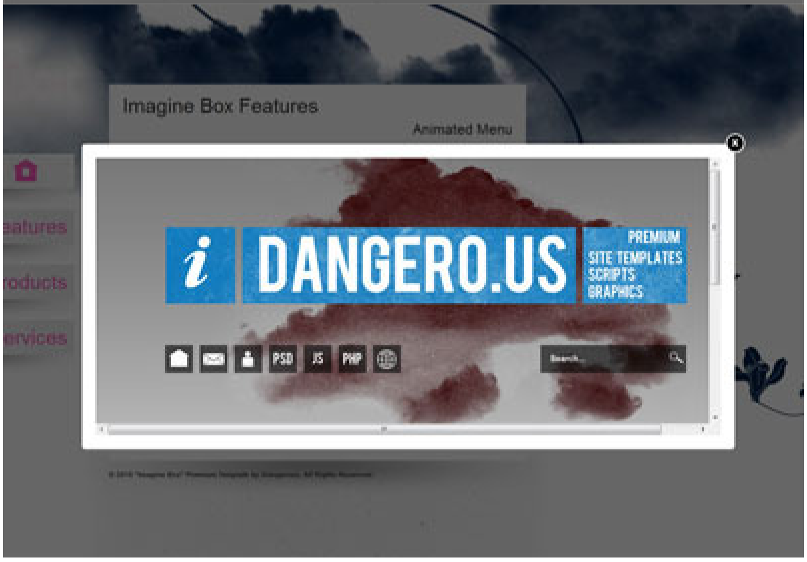
As you can see it’s not difficult to create your own products. First of all you have to create container for your product (<div> block with a “product” class). Then you need to put product image (using simple <img /> tag) and the product HTML description (using <div> block with a “description” class) into this container. You’ll need to do this for every product. And don’t forget to add **`<div class=”clear”></div>`** after each second product.

9. Popup Window

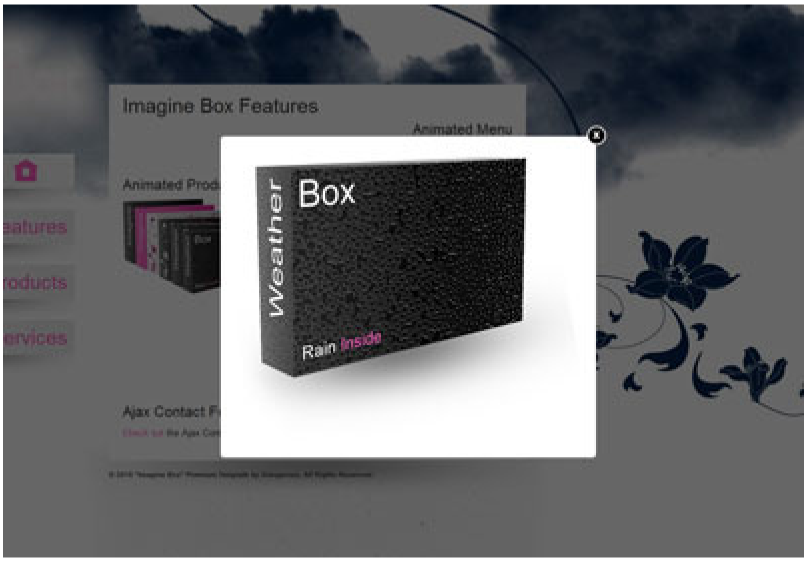
Imagine Box template has its own engine to call popup windows. Here are the screenshots:



Popup window with HTML content inside



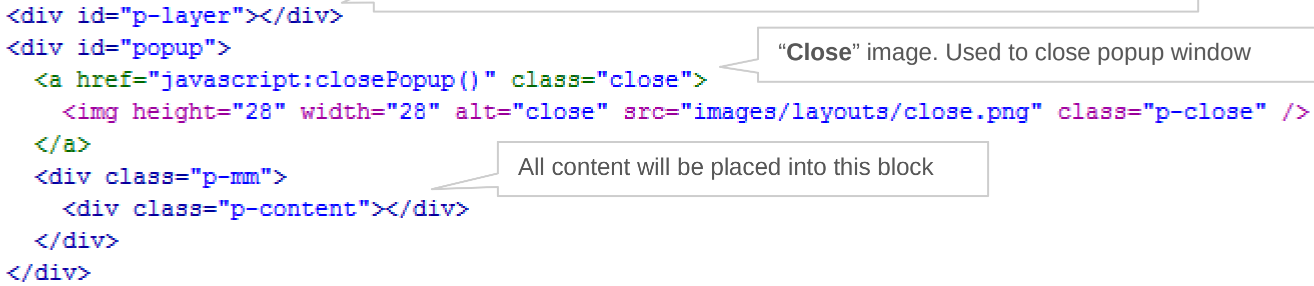
Popup window with iFrame inside



Popup window with Image inside

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

Popup Window Layout is located in the **includes/popup.php** file:



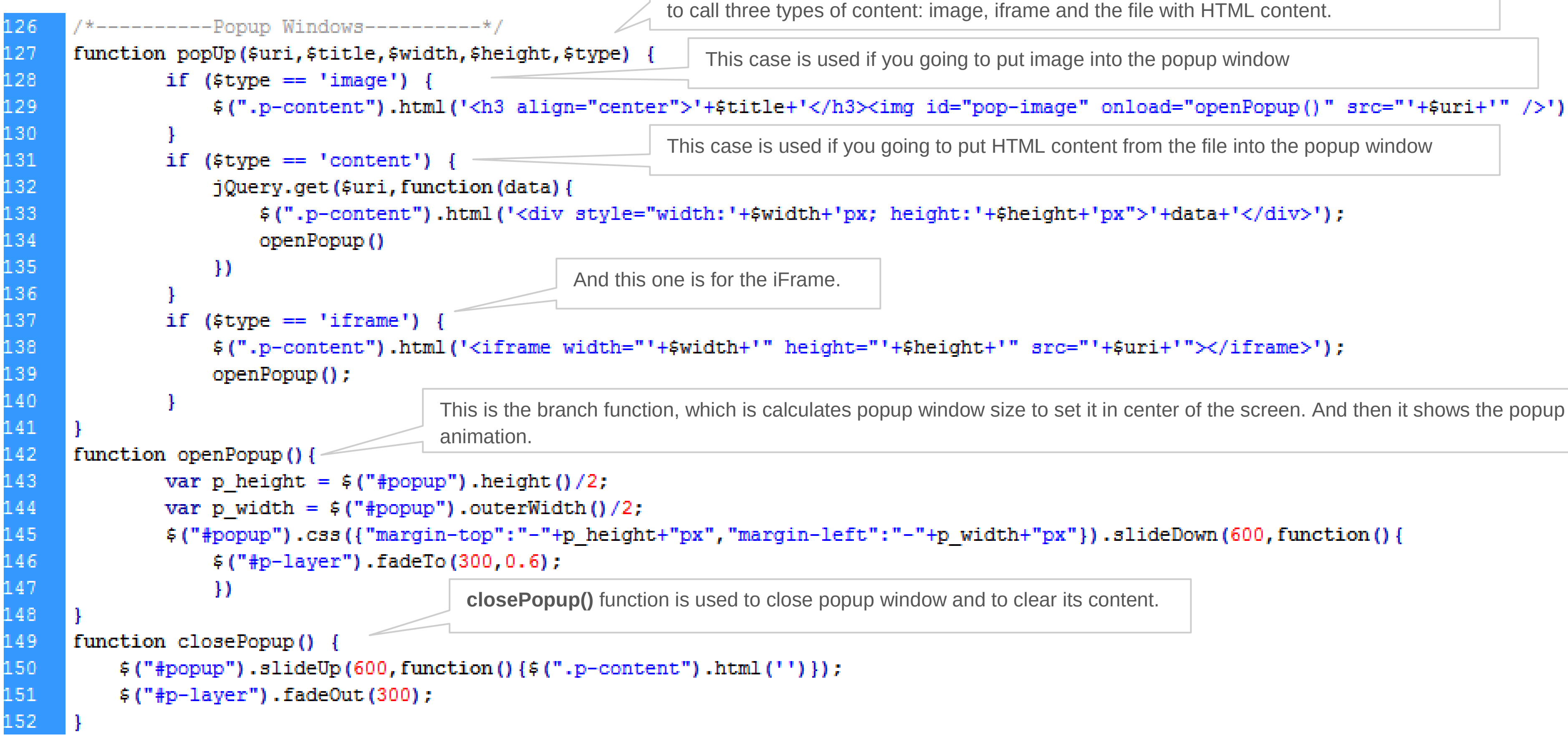
```
<div id="p-layer"></div>
<div id="popup">
  <a href="javascript:closePopup()" class="close">
    <img height="28" width="28" alt="close" src="images/layouts/close.png" class="p-close" />
  </a>
  <div class="p-mm">
    <div class="p-content"></div>
  </div>
</div>
```

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

"**Close**" image. Used to close popup window

All content will be placed into this block

As mentioned above there is the popUp() function which allows you to popup three type of content: HTML content from the file, iframe and image.



```
/*----------Popup Windows----------*/
function popUp($uri,$title,$width,$height,$type) {
        if ($type == 'image') {
            $(".p-content").html('<h3 align="center">'+$title+'</h3><img id="pop-image" onload="openPopup()" src="'+$uri+'" />')
        }
        if ($type == 'content') {
            jQuery.get($uri,function(data){
                $(".p-content").html('<div style="width:'+$width+'px; height:'+$height+'px">'+data+'</div>');
                openPopup()
            })
        }
        if ($type == 'iframe') {
            $(".p-content").html('<iframe width="'+$width+'" height="'+$height+'" src="'+$uri+'"></iframe>');
            openPopup();
        }
}
function openPopup(){
        var p_height = $("#popup").height()/2;
        var p_width = $("#popup").outerWidth()/2;
        $("#popup").css({"margin-top":"-"+p_height+"px","margin-left":"-"+p_width+"px"}).slideDown(600,function(){
            $("#p-layer").fadeTo(300,0.6);
            })
}
function closePopup() {
    $("#popup").slideUp(600,function(){$(".p-content").html('')});
    $("#p-layer").fadeOut(300);
}
```

popUp() function provides an opportunity to call a Popup Window. This function allows you to call three types of content: image, iframe and the file with HTML content.

This case is used if you going to put image into the popup window

This case is used if you going to put HTML content from the file into the popup window

And this one is for the iFrame.

This is the branch function, which is calculates popup window size to set it in center of the screen. And then it shows the popup window with a slide animation.

closePopup() function is used to close popup window and to clear its content.

As you can see above the popUp() function has the 5 variables:

$uri: path to image or file with HTML content, or URL for the iFrame content

$title: used only for popup image to set the title of popup window

$width: width of popup window in px (not used for image)

$height: height of popup window in px (not used for image)

$type: used to specify type of content for popup. Could take three values: **content**, **image** or **iframe**

So there are the two ways to call a popup window.

1. The first way is to use the **popUp()** function directly from, for example, **onclick** action:

$uri

$title – blank value if this is not image

$width=600 px

$height=300 px

$type=content. It's mean that we'll put into popup window the HTML content from file

```
<a href="#" onclick="popUp('content/features.php','',600,300,'content')">Features</a>
or
<a href="javascript:popUp('http://www.idangero.us','',600,300,'iframe')">iDangero.us</a>
or
<img src="images/small_image.jpg" onclick="popUp('images/big_image.jpg','Big Image','','','image')" />
```

$uri

$title ="Big Image". Used to specify popup window title

$width='' – blank value for image

$height='' – blank value for image

$type=image. It's mean that we'll put into popup window an image

2. The second way is much easier. We could use the simple link with a special class to call popup windows with a desired type of content:

a) To call a popup window with HTML content inside we should use this formatting for links:

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

Specify file location in the **href** attribute

Use **rel** attribute to specify window width (not necessary)

And **rev** attribute for window height (not necessary)

```
<a class="pop-content" href="content/features.php" rel="300" rev="200">Features</a>
```

b) To call a popup window with iFrame inside we should use this formatting for links:

Use "iframe" class to call a popup window with iFrame inside

Specify iFrame URL in the **href** attribute

Use **rel** attribute to specify window width (not necessary)

And **rev** attribute for window height (not necessary)

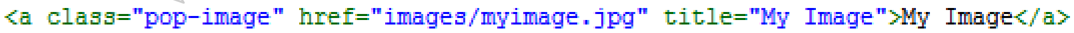
```
<a class="iframe" href="http://idangero.us" rel="300" rev="200">iDangero.us</a>
```

c) To call a popup window with an image inside we should use this formatting for links:

Use “pop-image” class to call a popup window with an Image inside

Specify image location in the **href** attribute

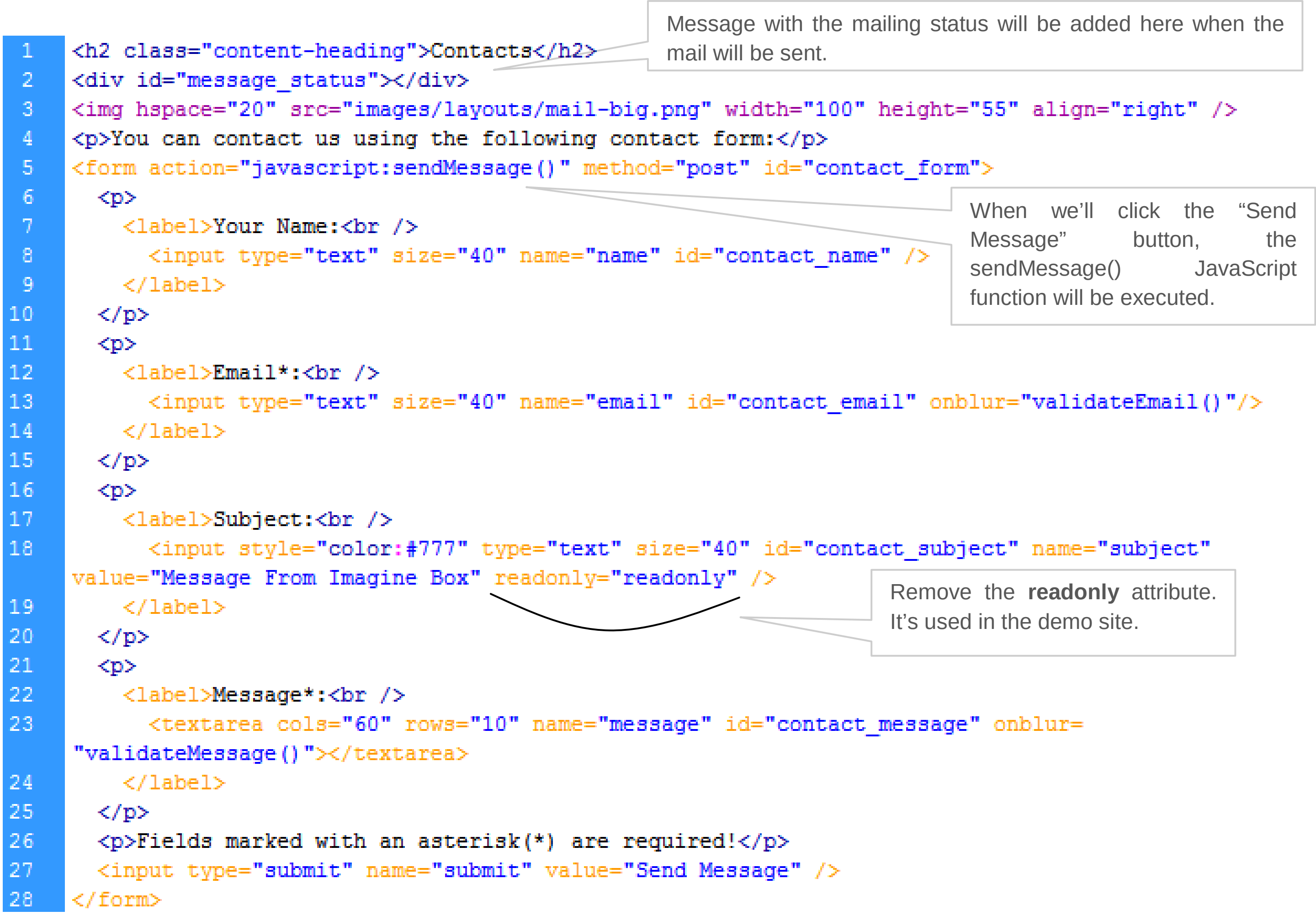
Use **title** attribute to specify window title (not necessary)



```
<a class="pop-image" href="images/myimage.jpg" title="My Image">My Image</a>
```

10. Ajax Contact Form

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



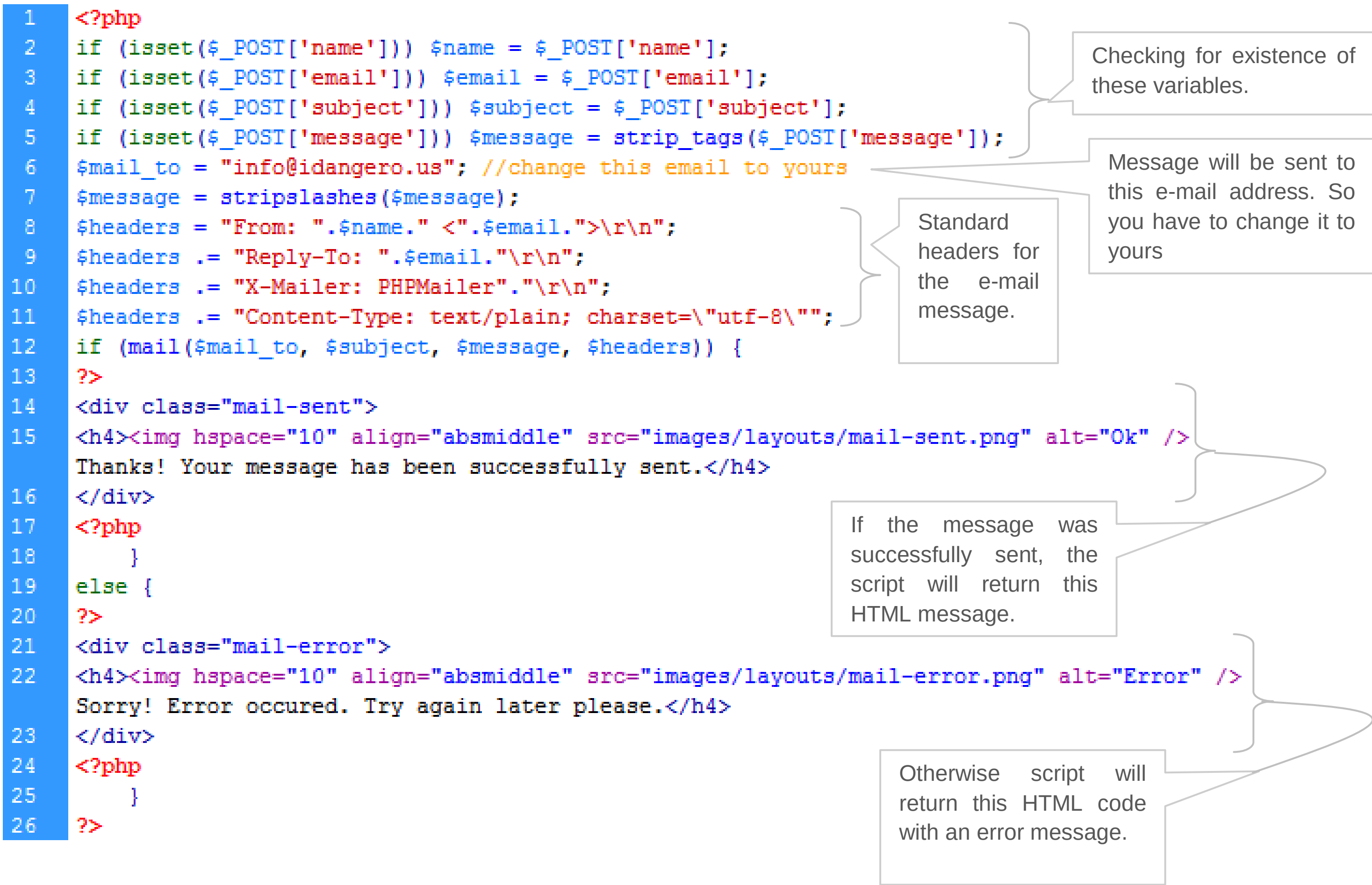
```
<h2 class="content-heading">Contacts</h2>
<div id="message_status"></div>
<img hspace="20" src="images/layouts/mail-big.png" width="100" height="55" align="right" />
<p>You can contact us using the following contact form:</p>
<form action="javascript:sendMessage()" method="post" id="contact_form">
  <p>
    <label>Your Name:<br />
      <input type="text" size="40" name="name" id="contact_name" />
    </label>
  </p>
  <p>
    <label>Email*:<br />
      <input type="text" size="40" name="email" id="contact_email" onblur="validateEmail()"/>
    </label>
  </p>
  <p>
    <label>Subject:<br />
      <input style="color:#777" type="text" size="40" id="contact_subject" name="subject" value="Message From Imagine Box" readonly="readonly" />
    </label>
  </p>
  <p>
    <label>Message*:<br />
      <textarea cols="60" rows="10" name="message" id="contact_message" onblur="validateMessage()"></textarea>
    </label>
  </p>
  <p>Fields marked with an asterisk(*) are required!</p>
  <input type="submit" name="submit" value="Send Message" />
</form>
```

Message with the mailing status will be added here when the mail will be sent.

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

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 will return mailing status message. Let's look at this script more closely:



```php
<?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'])) $message = strip_tags($_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)) {
?>
<div class="mail-sent">
<h4><img hspace="10" align="absmiddle" src="images/layouts/mail-sent.png" alt="Ok" /> Thanks! Your message has been successfully sent.</h4>
</div>
<?php
    }
else {
?>
<div class="mail-error">
<h4><img hspace="10" align="absmiddle" src="images/layouts/mail-error.png" alt="Error" /> Sorry! Error occured. Try again later please.</h4>
</div>
<?php
    }
?>
```

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.

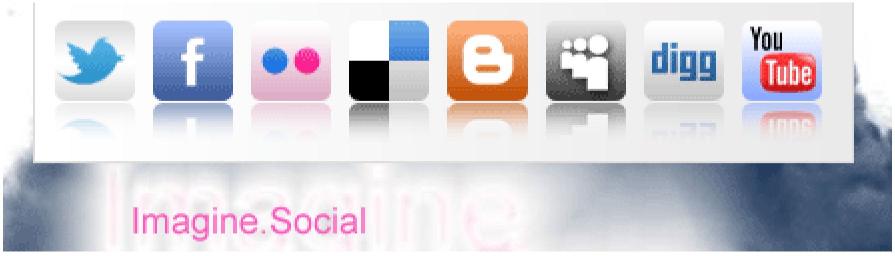
If the message was successfully sent, the script will return this HTML message.

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

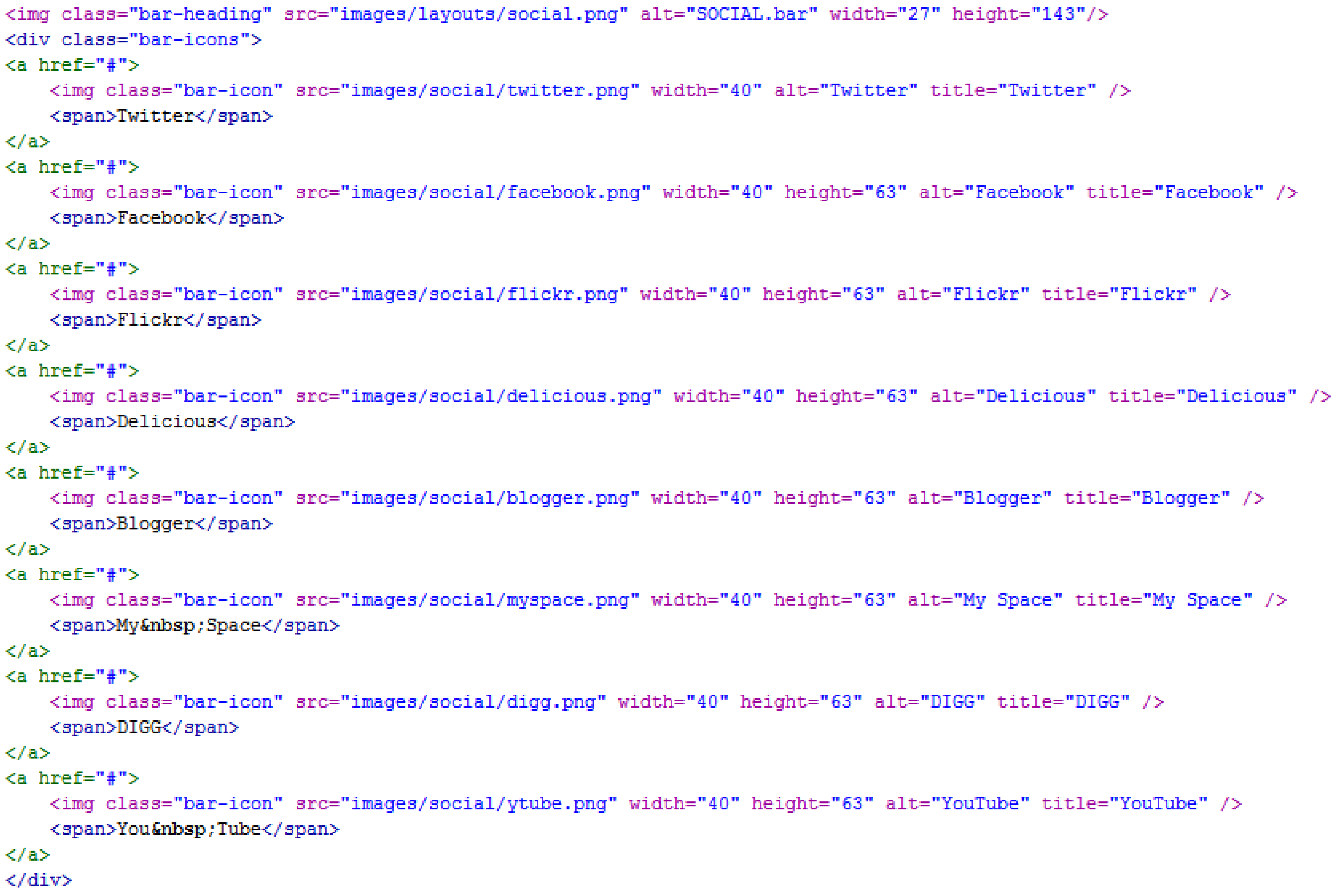
You can easily edit this status messages using the any HTML formatting.

11. Social.Bar

SOCIAL.BAR is a panel at the left top of the home page, where you can see the animated social icons which you can link to your relevant social pages (profiles).



To edit Social.Bar just open the **includes/social.bar.php** file:

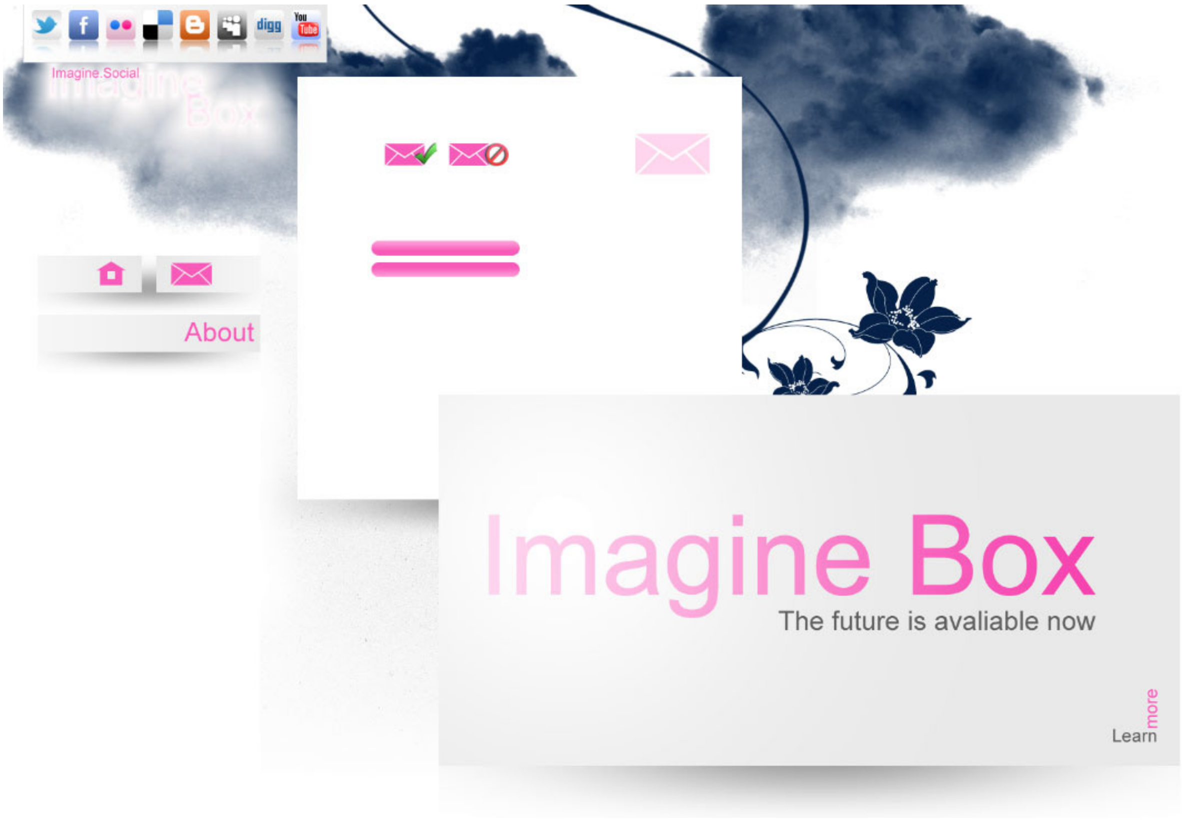


```
<img class="bar-heading" src="images/layouts/social.png" alt="SOCIAL.bar" width="27" height="143"/>
<div class="bar-icons">
<a href="#">
    <img class="bar-icon" src="images/social/twitter.png" width="40" alt="Twitter" title="Twitter" />
    <span>Twitter</span>
</a>
<a href="#">
    <img class="bar-icon" src="images/social/facebook.png" width="40" height="63" alt="Facebook" title="Facebook" />
    <span>Facebook</span>
</a>
<a href="#">
    <img class="bar-icon" src="images/social/flickr.png" width="40" height="63" alt="Flickr" title="Flickr" />
    <span>Flickr</span>
</a>
<a href="#">
    <img class="bar-icon" src="images/social/delicious.png" width="40" height="63" alt="Delicious" title="Delicious" />
    <span>Delicious</span>
</a>
<a href="#">
    <img class="bar-icon" src="images/social/blogger.png" width="40" height="63" alt="Blogger" title="Blogger" />
    <span>Blogger</span>
</a>
<a href="#">
    <img class="bar-icon" src="images/social/myspace.png" width="40" height="63" alt="My Space" title="My Space" />
    <span>My&nbsp;Space</span>
</a>
<a href="#">
    <img class="bar-icon" src="images/social/digg.png" width="40" height="63" alt="DIGG" title="DIGG" />
    <span>DIGG</span>
</a>
<a href="#">
    <img class="bar-icon" src="images/social/ytube.png" width="40" height="63" alt="YouTube" title="YouTube" />
    <span>You&nbsp;Tube</span>
</a>
</div>
```

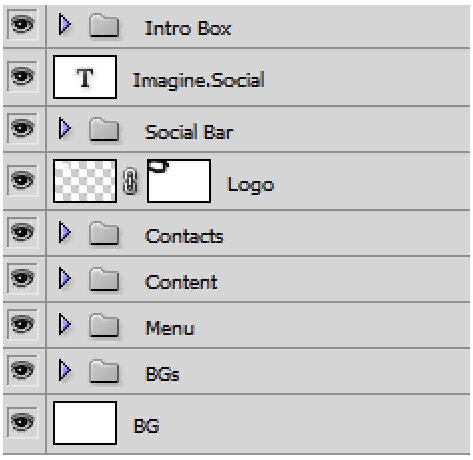
12. PSD Files & Structure

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

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

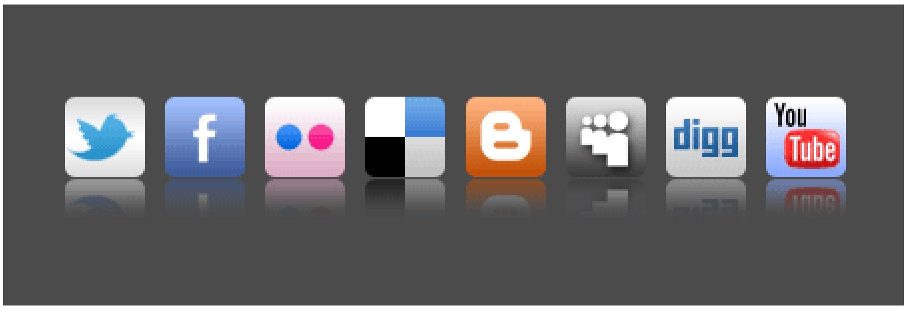


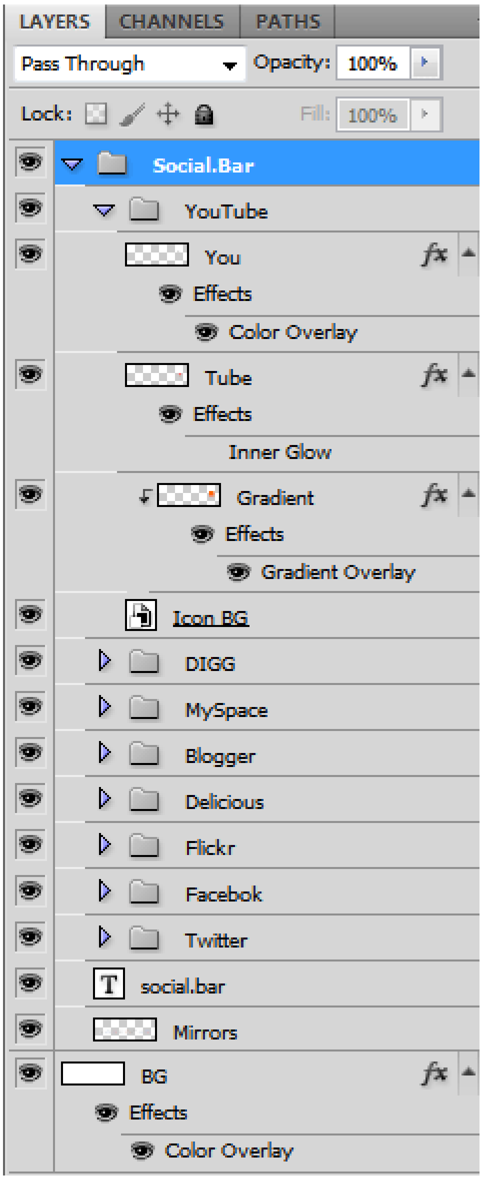
All layers in this file are well organized and easy to understand and redesign:



Intro Box
Imagine.Social
Social Bar
Logo
Contacts
Content
Menu
BGs
BG

Second .psd file is the **social-bar.psd**. This file contains the Social.Bar layered icons



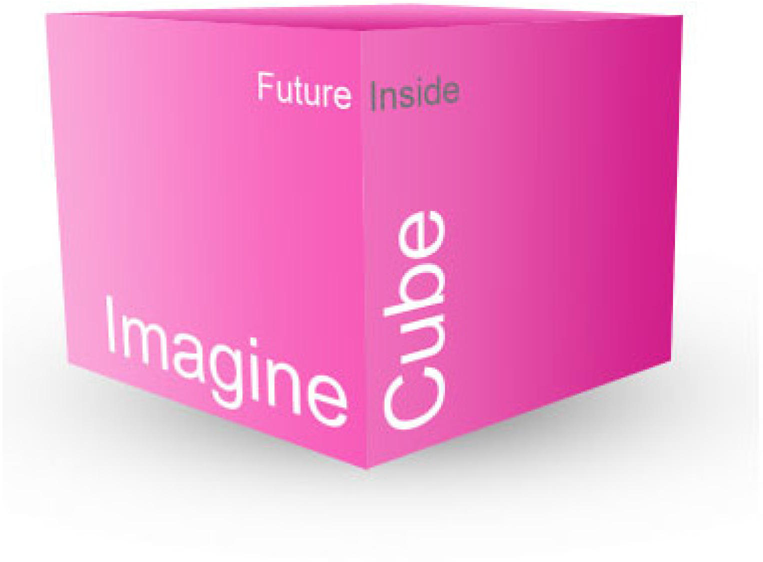


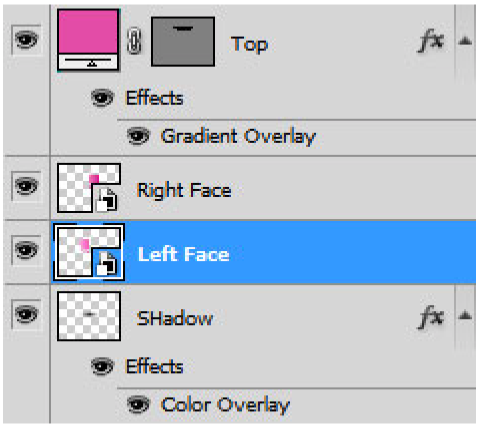
Pass Through
Opacity: 100%
Lock:
Fill: 100%

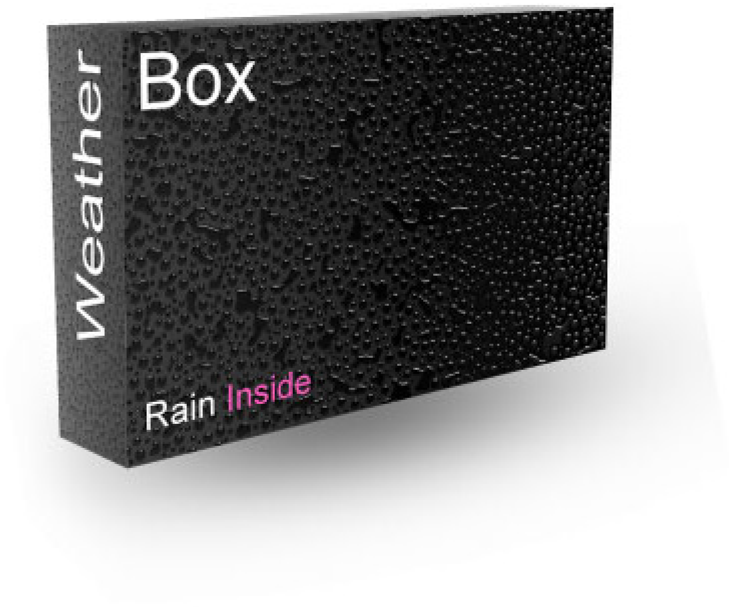
Social.Bar
YouTube
You
Effects
Color Overlay
Tube
Effects
Inner Glow
Gradient
Effects
Gradient Overlay
Icon BG
DIGG
MySpace
Blogger
Delicious
Flickr
Facebok
Twitter
social.bar
Mirrors
BG
Effects
Color Overlay

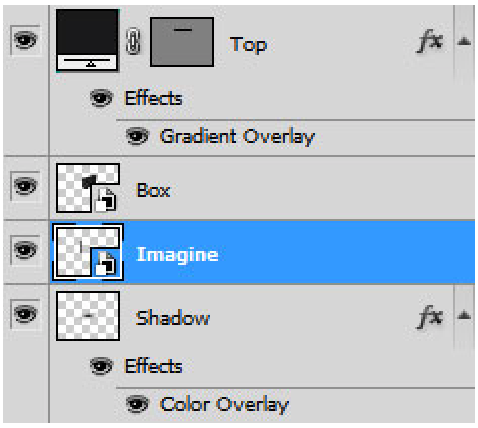
:

And the last two .psd files are the **product_box.psd** and the **product_cube.psd**. These files contain editable product boxes used in this template:




Top
Effects
Gradient Overlay
Right Face
Left Face
SHadow
Effects
Color Overlay




Top
Effects
Gradient Overlay
Box
Imagine
Shadow
Effects
Color Overlay

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 **Imagine Box 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**.