

Selected tutorials

from the first year of jankoatwarpspeed.com

By Janko Jovanovic, April 2009

www.jankoatwarpspeed.com

Content

Shortly about the author and this eBook	2
CSS Message Boxes for different message types	3
Create MessageBox user control using ASP.NET and CSS	7
Building better web forms: Labels in form layouts	13
Building better web forms: Context highlighting using jQuery	16
Justify elements using jQuery and CSS	20
Enhance your input fields with simple CSS tricks	22
Continuous scrolling pattern using JavaScript and ASP.NET	27
Add grunge effect to text using simple CSS	30
Create apple.com-like breadcrumb using simple CSS	32
Animate your message boxes with jQuery	35
Alternate way to select ASP.NET server controls using jQuery	37
Create Vimeo-like top navigation	38
Enlarge input fields on focus	42
How to create Skype-like buttons using jQuery	44

Shortly about the author and this eBook

My name is Janko Jovanovic and I work as a solution architect, web developer and designer. Basically, I project, develop and design websites and web applications.

Besides my everyday work I am very active in local development community and lecture on almost all major public events and local .NET User Group. I run JankoAtWarpSpeed.com blog about web development, web design, usability and user experience. I finished Information Technology College in Belgrade and gained MCSD certificate. The hobby that I am most passionate about is painting and drawing.

This eBook is a collection of some of my tutorials that I published during the first year of JankoAtWarpSpeed.com. It is free which means that you can print it for yourself and share it with your colleagues, but you are not allowed to republish it in any form. However, I'd like to ask you to consider the environment before printing this eBook ☺ Help saving our planet by reading this in electronic form.

For more information you can contact me via:

www.jankoatwarpspeed.com

this@jankoatwarpspeed.com

<http://twitter.com/dzovan>

CSS Message Boxes for different message types

Published on 22. May 2008

Can you believe this: Few days ago I went to my bank to check my credit score with the Credit Bureau. The bank official typed in my personal data and sent a request. Web application responded by displaying a yellow message box with an exclamation icon saying that data processing is still in progress. He checked several more times, but he didn't notice that at one moment the message changed to "Account available". But the message box hasn't changed. He continued to check a few more times and eventually he realized that the request was successful.

I don't know what was in the minds of developers and designers who created this application, but it certainly wasn't the user. This poor bank official was really frustrated. I can't imagine what the rest of the application looks like.

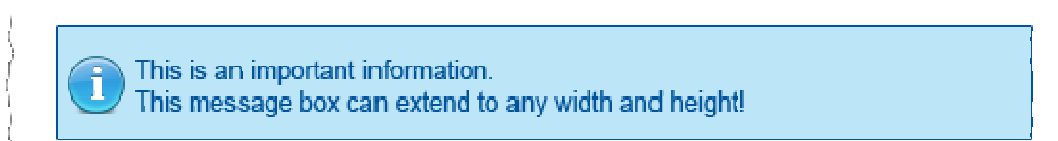
To prevent this, different message types should be displayed differently. My opinion is that every web application should handle four main message types: information, successful operation, warning and error. Each message type should be presented in a different color and different icon. A special message type represents validation messages.

I will show you a remake of CSS message boxes I used on my latest project. I changed them slightly just to make them simpler for this example. In next article, you will see how to create ASP.NET user control that can support different message types and how to style it using CSS. You will also see how to style ValidationSummary in a similar way.

Let's first take a quick look at message types.

1. Information messages

The purpose of information messages is to inform the user about something relevant. This should be presented in blue because people associate this color with information, regardless of content. This could be any information relevant to a user action.



For example, info message can show some help information regarding current user action or some tips.

2. Success messages

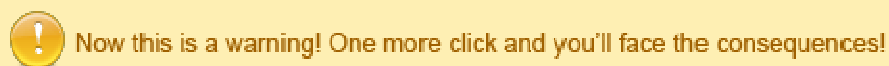
Success messages should be displayed after user successfully performs an operation. By that I mean a complete operation - no partial operations and no errors. For example, the message can say: "Your profile has been saved successfully and confirmation mail has been sent to the email address you provided". This means that each operation in this process (saving profile and sending email) has been successfully performed.



I am aware that many developers consider this as an information message type, but I prefer to show this message type using its own colors and icons - a green with a checkmark icon.

3. Warning messages

Warning messages should be displayed to a user when an operation couldn't be completed in a whole. For example "Your profile has been saved successfully, but confirmation mail could not be sent to the email address you provided.". Or "If you don't finish your profile now you won't be able to search jobs". Usual warning color is yellow and icon exclamation.



4. Error messages

Error messages should be displayed when an operation couldn't be completed at all. For example, "Your profile couldn't be saved." Red is very suitable for this since people associate this color with an alert of any kind.



Design process

Now when we know the way to present messages to users, let's see how to implement a it using CSS. Let's take a quick look at the design process.

I will keep this as simple as I can. The goal is to have a single div that implements a single CSS class. So the HTML markup will look like this:

```
<div class="info">Info message</div>
<div class="success">Successful operation message</div>
<div class="warning">Warning message</div>
<div class="error">Error message</div>
```

CSS class will add a background image to the div that will be positioned top-left. It will also create padding inside the div so that text can have enough white space around it. Note that left padding has to be wider to prevent text overlapping with the background image.



And here are the CSS classes for all four (five with validation) different message types:

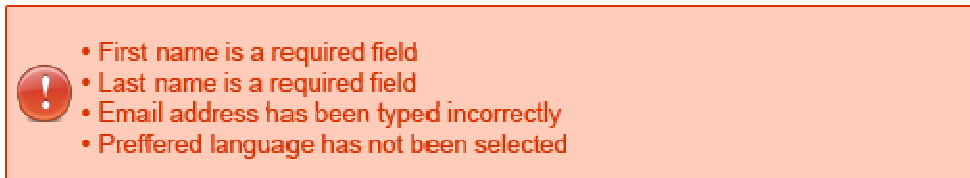
```
body{
    font-family:Arial, Helvetica, sans-serif;
    font-size:13px;
}
.info, .success, .warning, .error, .validation {
    border: 1px solid;
    margin: 10px 0px;
    padding:15px 10px 15px 50px;
    background-repeat: no-repeat;
    background-position: 10px center;
}
.info {
    color: #00529B;
    background-color: #BDE5F8;
    background-image: url('info.png');
}
.success {
    color: #4F8A10;
    background-color: #DFF2BF;
    background-image:url('success.png');
}
.warning {
    color: #9F6000;
    background-color: #FFEEB3;
    background-image: url('warning.png');
}
.error {
    color: #D8000C;
    background-color: #FFBABA;
    background-image: url('error.png');
}
```

Note: Icons used in this example are from [Knob Toolbar icons](#) collection.

Validation messages

I noticed that many developers can't distinguish between validation and other message types (such as error or warning messages). I saw many times that validation message is displayed as error message and caused confusion in the user's mind.

Validation is all about user input and should be treated that way. ASP.NET has built in controls that enable full control over user input. The purpose of validation is to force a user to enter all required fields or to enter fields in the correct format. Therefore, it should be clear that the form will not be submitted if these rules aren't matched. That's why I like to style validation messages in a slightly less intensive red than error messages and use a red exclamation icon.



CSS class for validation message is almost identical to others (note that in some attributes are defined in previous code sample):

```
.validation {  
    color: #D63301;  
    background-color: #FFCCBA;  
    background-image: url('validation.png');  
}
```

Conclusion

Messages are an important part of the user experience that is often omitted. There are many articles that show nicely styled message boxes but it is not just a matter of design. It should provide a user with meaningful information, semantically and visually.

There are two other articles I would like to recommend you:

- [CSS Message Box collection](#)
- [Create a valid CSS alert message](#)

Create MessageBox user control using ASP.NET and CSS

Published on 28. May 2008

In my previous article I wrote about an importance of handling different message types properly. In this article we'll create a simple ASP.NET user control and apply CSS styles shown in previous article.

Let me explain why we need a user control. If you want every aspx page to support same notification logic, you will need a same functionality on each page. Thus, a user control emerges as logic solution. This can be placed on each aspx page or on a Master Page if you use it. This way you can render any information or an error in the same manner on any place in your application. You can, for example, use this control to show your error messages.



Let's see what functionalities we want to implement:

- We want our control to render four different message types (described in previous article). Thus we'll need a way to inform our control which CSS class to implement
- We need the ability to pass the text to be displayed
- At the end, we want to have a close button on our control to enable a user to hide the message after reading. Beside this, we have to provide the developers with the ability to choose whether to show close button or not.

To do this, we'll create a new user control and apply the design that supports the requests above:

```
<div class="container">
  <asp:Panel ID="MessageBox" runat="server">
    <asp:HyperLink runat="server" id="CloseButton" >
      <asp:Image runat="server" ImageUrl="~/images/close.png"
        AlternateText="Click here to close this message" />
    </asp:HyperLink>
    <p><asp:Literal ID="litMessage" runat="server"></asp:Literal></p>
  </asp:Panel>
</div>
```

And CSS classes that style this control:

```
body{
  font-family:Arial, Helvetica, sans-serif; font-size:13px;
}
.info, .success, .warning, .error, .validation {
  border: 1px solid;
  margin: 10px 0px;
```

For more tutorials, live demos and other articles visit www.jankoatwarpspeed.com

Please consider the environment before printing this eBook!


```

        background-repeat: no-repeat;
        background-position: 10px center;
    }
    .info {
        color: #00529B;
        background-color: #BDE5F8;
        background-image: url('images/info.png');
    }
    .success {
        color: #4F8A10;
        background-color: #DFF2BF;
        background-image: url('images/success.png');
    }
    .warning {
        color: #9F6000;
        background-color: #FEEFB3;
        background-image: url('images/warning.png');
    }
    .error {
        color: #D8000C;
        background-color: #FFBABA;
        background-image: url('images/error.png');
    }
    .container
    {
    }

    .info p, .success p, .warning p, .error p {
        padding: 0px 50px;
    }
    .info a, .success a, .warning a, .error a {
        float: right;
        padding: 10px;
        cursor: pointer;
    }
    .container img {
        border: none;
    }

```

We have one DIV that implements class "container". It will act as a most outer container. I left that class blank but you can apply some styles here, e.g. width. Next, we'll have one Panel control that will be accessible from the server, one Literal control that will render a message and one HyperLink with an Image that will be our Close button. That's our design.

Let's see what is happening in the code behind of our user control.

We'll have one boolean property that will indicate whether to render Close button or not.

```
public bool ShowCloseButton { get; set; }
```

Next, we'll declare an Enum that will hold our four different message types:

```
public enum MessageType
{
```

```
Error = 1,  
Info = 2,  
Success = 3,  
Warning = 4  
}
```

Now, we need a method that will show the panel with the message and apply correct class.

```
public void Show(MessageType messageType, string message)  
{  
    CloseButton.Visible = ShowCloseButton;  
    litMessage.Text = message;  
    switch (messageType)  
    {  
        case MessageType.Error:  
            MessageBox.CssClass = "error";  
            break;  
        case MessageType.Info:  
            MessageBox.CssClass = "info";  
            break;  
        case MessageType.Success:  
            MessageBox.CssClass = "success";  
            break;  
        case MessageType.Warning:  
            MessageBox.CssClass = "warning";  
            break;  
    }  
    this.Visible = true;  
}
```

This method does a few things:

- It will show/hide Close button according to our ShowCloseButton property.
- It will apply a message to a Literal
- It will apply a CSS class based on MessageType Enum

Let me explain a few things here. First, I chose Literal instead of Label control, because Label is rendered in SPAN element and Literal to pure text.

Next, our Close button will be a HyperLink because it has to be accessible from the server. You are probably asking yourself why. There are two reasons for this. First, close button will have to be displayed ONLY if our ShowCloseButton property is set to true. If so, we have to attach "onclick" event handler to a HyperLink and pass it a ClientID of our Panel control. Why this?

There is a difference between an ID and a ClientID. Each server control will have a ClientID different from ID you gave it. ClientID consists of a control ID and all parent control's IDs. This is how JavaScript see our control. And if we want to find it on the client we'll search for a ClientID. ClientID typically looks like: ParentControl1ID_ParentControl2ID_ControlID.

If this isn't simple enough let me explain it this way - if we have more than one MessageBox panel on a page, document.getElementById won't find the right one if we try to get this element with its ID. You have to provide a full client name with all parent controls.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (ShowCloseButton)
        CloseButton.Attributes.Add("onclick", "document.getElementById('" +
            MessageBox.ClientID + "').style.display = 'none'");
}
```

That's why we'll attach onclick event handler on the server.

This will work, but I am not satisfied with that large "switch" block in our Show method. We can optimize this a little. Since our class names and Enum items are the same, we can replace a complete "switch" block with a single line of code:

```
public void Show(MessageType messageType, string message)
{
    CloseButton.Visible = ShowCloseButton;
    litMessage.Text = message;
    MessageBox.CssClass = messageType.ToString().ToLower();
    this.Visible = true;
}
```

This line will read the NAME of Enum item that we passed to our Show method and assign it to MessageBox panel as a CSS class.

What else can we do? We can create four wrapper methods to make it easier for developers:

```
public void ShowError(string message)
{
    Show(MessageType.Error, message);
}
public void ShowInfo(string message)
{
    Show(MessageType.Info, message);
}
public void ShowSuccess(string message)
{
    Show(MessageType.Success, message);
}
public void ShowWarning(string message)
{
    Show(MessageType.Warning, message);
}
```

This isn't required, but I prefer to have more choices during development. So the final code will look like this

```
public partial class MyMessageBox : System.Web.UI.UserControl
{
    #region Properties
    public bool ShowCloseButton { get; set; }
    #endregion
    #region Load
    protected void Page_Load(object sender, EventArgs e)
```

```

{
    if (ShowCloseButton)
        CloseButton.Attributes.Add("onclick", "document.getElementById('" +
            MessageBox.ClientID + "').style.display = 'none'");
    }
    #endregion
    #region Wrapper methods
    public void ShowError(string message)
    {
        Show(MessageType.Error, message);
    }
    public void ShowInfo(string message)
    {
        Show(MessageType.Info, message);
    }
    public void ShowSuccess(string message)
    {
        Show(MessageType.Success, message);
    }
    public void ShowWarning(string message)
    {
        Show(MessageType.Warning, message);
    }
    }
    #endregion
    #region Show control
    public void Show(MessageType messageType, string message)
    {
        CloseButton.Visible = ShowCloseButton;
        litMessage.Text = message;
        MessageBox.CssClass = messageType.ToString().ToLower();
        this.Visible = true;
    }
    #endregion
    #region Enum
    public enum MessageType
    {
        Error = 1,
        Info = 2,
        Success = 3,
        Warning = 4
    }
    #endregion
}

```

Now, let's see how our user control can be shown on a pages:

```

<uc1:MyMessageBox ID="MyMessageBox1" runat="server" ShowCloseButton="true" />
<uc1:MyMessageBox ID="MyMessageBox2" runat="server" ShowCloseButton="false" />
<uc1:MyMessageBox ID="MyMessageBox3" runat="server" ShowCloseButton="true" />
<uc1:MyMessageBox ID="MyMessageBox4" runat="server" />

```

You can just drag&drop a control on a form and set (or omit) ShowCloseButton property. In a code behind, you can show this control in many ways:

For more tutorials, live demos and other articles visit www.jankoatwarpspeed.com
Please consider the environment before printing this eBook!

```
MyMessageBox1.ShowInfo("This is an information. This MessageBox works!");  
MyMessageBox2.Show(MyMessageBox.MessageType.Success, "Your profile has been  
changed");  
MyMessageBox3.ShowError("This is an error message");  
MyMessageBox4.ShowWarning("This is a warning!");
```

Conclusion

You saw how to create ASP.NET user control that can handle different message types. You also saw how to implement a Close button on a server control.

I am aware that this code isn't perfect, although it did a great job for me on several projects. Did you ever need or do something similar? Do you have any other ideas how this could be implemented? Share it!

Building better web forms: Labels in form layouts

Published on 6. June 2008

One of the requests that I often had is to refactor the presentation layer of web applications. One of the things that purports is a form layout refactoring.

What is so important about form layout?

First time when I was thinking of the best solution I remembered that I read a great article "[Web Application Form Design](#)" written by [Luke Wroblewski](#). Luke explained how label position can affect web form usability. He described the advantages and disadvantages of common label positions and how visual elements fit in this. To put it in short, there are three common ways to position a label associated with an input field:

Vertical labels

Email Address

Password

Left-aligned labels

Email Address

Password

Right-aligned labels

Email Address

Password

It is hard to say which one is better, although a great article "[Label Placement in Forms](#)" from [UXMatters](#) can answer a lot of questions. But it is easy to say that Left-aligned labels are the most common scenario in web sites and web applications.

You can find a lot of articles on how to build a "public" web form - the one that is a part of a web site. But web application forms require an extra work. Web application forms tend to be more complex than web site forms due to complex business rules and (often irrational) user requests.

For more tutorials, live demos and other articles visit www.jankcoatwarpspeed.com

Please consider the environment before printing this eBook!

An interesting article "[Functioning Form - Web Application Form Layout](#)" gives one of possible solutions using "background colors" (pay attention to comments on this article). However, if you have a large number of input fields and labels that somehow has to be ordered and grouped (which is a common scenario in web applications), this solution produces a hard-to-fill form. It looks somehow overworked. Especially if you have to apply some color schema other than gray.

Name	<input type="text"/>
Street Address	<input type="text"/> <input type="text"/>
City	<input type="text"/>
State	<input type="text" value="Enter Text"/>
Zip/Postal	<input type="text"/> <input type="text"/>
Province	<input type="text"/>

So, what is the solution?

Left-aligned labels weren't enough. Background colors were too much. Then I decided to try something in between. I tried with "underlined labels" - visual elements that can be seen on products like [Microsoft CRM](#).

<u>First name</u>	<input type="text"/>
<u>Last Name</u>	<input type="text"/>

As you can see on the example above, underlined labels are an extension to Left-aligned labels. They visually associate labels with input fields and, I believe, do not interrupt eye movement as it might have been the case with "background colors".

Now, the question is can this improve overall interactivity. By this I mean to reduce the eye movement and the time required to fill the form. These are the parameters that are being tested using Eyetracking testing. To find out more about this I strongly recommend you to read "[Introduction to Eyetracking](#)" from [UXMatters](#).

But there is a catch...

This looks fine when there is a single input field that has the common height. But take a look at the example below.

Email Address

Password

Account type ☐ Basic (Free registration)
☐ Pro 9.99\$
☒ Expert 19.99\$

Some description

If the area that contains input field(s) expands, the label associated with it just won't serve the purpose anymore. Just take a look at "Account type" label; Does it refers just to "Basic" option? And what about "Some description"? The line under the label is not aligned neither to the top nor to the bottom of input field. All of this might be confusing to users.

The only thing I could think of is to vertically align labels to the bottom of input field area.

Email Address

Password

Account type ☐ Basic (Free registration)
☐ Pro 9.99\$
☒ Expert 19.99\$

Some description

Although users liked this layout, I am really interested in finding out does this improves overall usability. I don't know if anyone tried to test this, but if answer is yes, it would be nice to see the results.

Conclusion

In this article I presented you an alternative way to design form layouts. Again, I would strongly recommend you to read following articles:

- [Web Application Form Design](#)
- [Functioning Form - Web Application Form Layout](#)
- [Label Placement in Forms](#)
- [Introduction to Eyetracking](#)

If you have any experience with form layouts or if you just agree or disagree, please share your thoughts!

Building better web forms: Context highlighting using jQuery

Published on 9. June 2008

In my previous article "[Labels in form layouts](#)" I wrote about how I have implemented "underlined labels" on web application forms. However, due to complexity of web applications I developed I often needed to find a way to focus a user on a current context. When I discovered jQuery I decided to use it because of its powerful features.

Since the forms were often very complex and had (too much I would say) controls on them, I needed to focus a user's attention on the current context. I first thought to highlight the current row.

In this example I'll show you what I tried to do.

The image above shows that whatever users do, current row will be highlighted. The html markup for the example above will look like this:

```
<h3>Sample web form</h3>
<div class="content">
  <div class="row">
    <div class="left">First name</div>
    <div class="right"><input name="Text1" type="text" class="text" /></div>
    <div class="clear"></div>
  </div>
  <div class="row">
    <div class="left">Last name</div>
```

For more tutorials, live demos and other articles visit www.jankcoatwarpspeed.com

Please consider the environment before printing this eBook!

```
        <div class="right"><input name="Text1" type="text" class="text" /></div>
        <div class="clear"></div>
    </div>
    <div class="row">
        <div class="left">Email</div>
        <div class="right"><input name="Text1" type="text" class="text" /></div>
        <div class="clear"></div>
    </div>
    <div class="row">
        <div class="left">Password</div>
        <div class="right"><input name="Text1" type="text" class="text" /></div>
        <div class="clear"></div>
    </div>
</div>
<hr />
<!-- Other rows here -->
<input name="Button1" type="button" value="Do some action" />
```

And CSS classes are pretty simple as well:

```
body{
    font-family:Arial, Helvetica, sans-serif;
    font-size: 13px;
}
.content{
    padding:10px;
    width:370px
}
.left{
    width:150px;
    float:left;
    padding:7px 0px 0px 7px;
    min-height:24px;
}
.right{
    width:200px;
    float:left;
    padding:5px;
    min-height:24px;
}
.clear{
    float:none;
    clear:both;
    height:0px;
}
.row{
    background-color:none;
    display:block;
    min-height:32px;
}
.text{
    width:190px;
}
```

For more tutorials, live demos and other articles visit www.jankoatwarpspeed.com

Please consider the environment before printing this eBook!

```
.ruler{
    width:400px; border-bottom:dashed 1px #dcdcdc;
}
tr:focus{
    background-color:#fcfcf0;
}
td{
    vertical-align:top;
}
.over{
    background-color:#f0f0f0;
}
.out{
    background-color:none;
}
```

Each "row" consists of two div's, one for a label and other for an input field(s). So what we tried to achieve it to highlight a current row whenever user click on any element inside that row. It could be an input field or a label.

Now let's see the simplicity of jQuery:

```
$(document).ready(function(){
    $('.left, .content input, .content textarea, .content select').focus(function(){
        $(this).parents('.row').addClass("over");
    }).blur(function(){
        $(this).parents('.row').removeClass("over");
    });
});
```

Whenever an input, textarea, select, or element that has "left" CSS class get focus, append "over" CSS class to all parents that have "row" CSS class. Also, whenever an input, textarea, select, or element that has "left" CSS class lose focus, remove "over" CSS class from all parents that have "row" CSS class.

This was very interesting and extremely simple solution, but in forms more complex that the one in the example above it just didn't make much improvement. What I wanted actually was to highlight a group of related fields. Thanks to jQuery it was very easy to change the rules:

```
$(document).ready(function(){
    $('.left, .content input, .content textarea, .content select').focus(function(){
        $(this).parents('.content').addClass("over");
    }).blur(function(){
        $(this).parents('.content').removeClass("over");
    });
});
```

Note that the only change is parents() target. Instead of changing only parent row, we can highlight a whole group of rows. Each div that has "content" CSS class will be highlighted whenever any of his children get focus and vice versa.

And this is what it looks like in the end:

Sample web form

First name

Last name

Email

Password

Address

Postal

City

Country

Account type ☒ Basic (Free registration)
☐ Pro 9.99\$
☐ Expert 19.99\$

Some description

Do some action

Sample web form

First name

Last name

Email

Password

Address

Postal

City

Country

Account type ☒ Basic (Free registration)
☐ Pro 9.99\$
☐ Expert 19.99\$

Some description

Do some action

In very complex web forms this enables users to focus only on a current action. By using attractive color schema and jQuery animations, this can be even more interesting.

Conclusion

You saw how you can easily improve the user experience by highlighting the current context. You also saw that it is extremely easy to do it by using jQuery. If you didn't use jQuery I can only suggest you to visit [jQuery official website](http://jquery.com) and download the library. There, you can find the documentation, examples and other stuff as well.

Justify elements using jQuery and CSS

Published on 9. July 2008

When creating a web form you have to make a functional and visually aligned layout. The simplest way to do this is to place elements in a table or by fixing width of labels. Tables stretch its cells according to width of largest element in a column. That way you can have aligned form. Fixing label width will also allow you to have a hard-coded but aligned form.

But...

But what if you don't want to use tables? Or what if you don't know how long labels could be because you are developing an application that has many localized strings? And you still want to align input elements according to the width of the longest label?

Then you can justify all labels. By justifying I mean to make them same width. And the simplest way to do this is to use [jQuery](#).

If you create a simple web form like in the example below all you have to do is to set float:left and display:block for both label and input field.

Name

Email address

Website

These are the CSS styles:

```
label, input[type="text"]{
    float:left;
    display:block;
}
label
{
    margin-right: 5px;
}
.field{
    width:100%;
    overflow:auto;
    margin:5px 0px;
}
```

Looks familiar? Usually, you would set the width of labels to some number, let's say 100px. But, let's do another thing. Let's calculate the width of a largest label and apply it to all labels. jQuery function below does exactly what I wrote:

```
$(document).ready(function() {
    var max = 0;
    $("label").each(function(){
```

For more tutorials, live demos and other articles visit www.jankcoatwarpspeed.com
Please consider the environment before printing this eBook!

```
        if ($(this).width() > max)
            max = $(this).width();
    });
    $("label").width(max);
});
```

And visually it looks like this:

Name	→	<input type="text"/>
Email address		<input type="text"/>
Website	→	<input type="text"/>

Conclusion

It's easy to do many tricks using jQuery and this was just one of them. Maybe there are simplest solutions to do this, but it was fun to see putting jQuery in most common scenarios like this one.

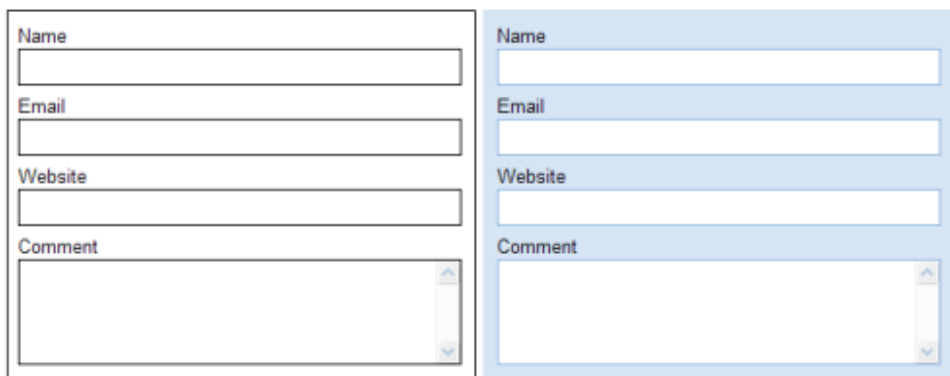
Enhance your input fields with simple CSS tricks

Publish on 27. July 2008

We're all trying to build an effective and good looking web forms. But there are always a new challenges. If you read my previous articles on how to build a better web forms, you could have noticed there are so many details. Label positioning, context highlighting or justifying elements. But, with just a few simple CSS tricks you can make a usual, boring web forms more effective and exciting.

The example you are going to see is something that you use every day: blog comment form. Ok, so what can you do to enhance a web form? You can...

...add some borders



The image displays two versions of a web form side-by-side. The left version is a standard, unenhanced form with a thin black border and no padding. The right version is an enhanced version with a light blue background, a solid blue border, and padding around the input fields and labels. The form contains four fields: Name, Email, Website, and Comment. The labels are positioned above the input fields.

At least what you can do is to add borders and padding to your input fields. Two examples above shows how you can set the border color to match your color schema. It is also a good practice to add some padding to input fields. That will make forms more clear. In this example I set the 4px padding to inputs and textarea.

```
#inputArea
{
    font-family: Arial, Sans-Serif;
    font-size: 13px;
    background-color: #d6e5f4;
    padding: 10px;
}

#inputArea input[type="text"], #inputArea textarea
{
    font-family: Arial, Sans-Serif;
    font-size: 13px;
    margin-bottom: 5px;
    display: block;
    padding: 4px;
    border: solid 1px #85b1de;
    width: 300px;
}
```

Let me review the code above shortly. This is the code for the second example, the blue one. The entire form has a light blue background #d6ee5f4 and 10px padding. Each input element is displayed as a block, which ensures that labels are positioned above input fields.

Now, this was very simple. But you can do more. You can...

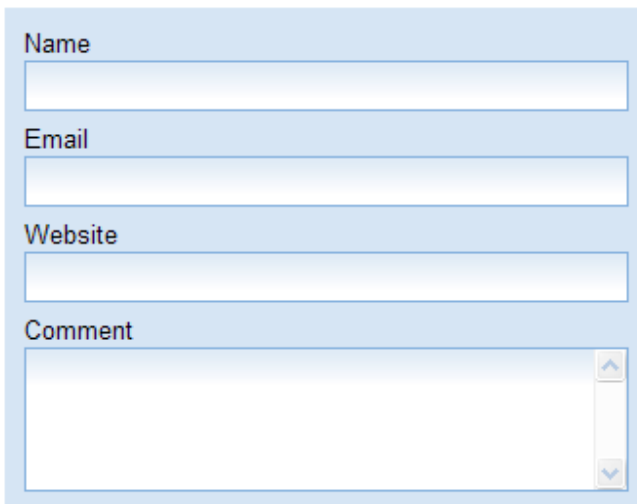
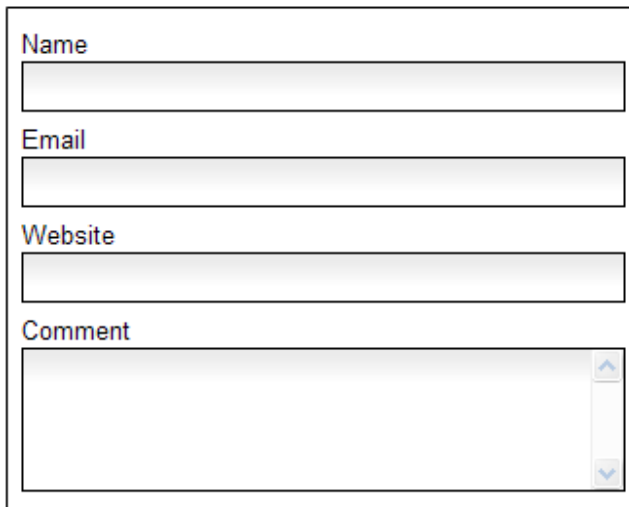
... add some background

You can also add some solid background like in the example below



```
#inputArea input[type="text"], #inputArea textarea
{
  font-family: Arial, Sans-Serif;
  font-size: 13px;
  margin-bottom: 5px;
  display: block;
  padding: 4px;
  border: solid 1px #85b1de;
  width: 300px;
  background-color: #EDF2F7;
}
```

Or you can add a soft gradient as a background. The examples above shows gray and blue gradients.



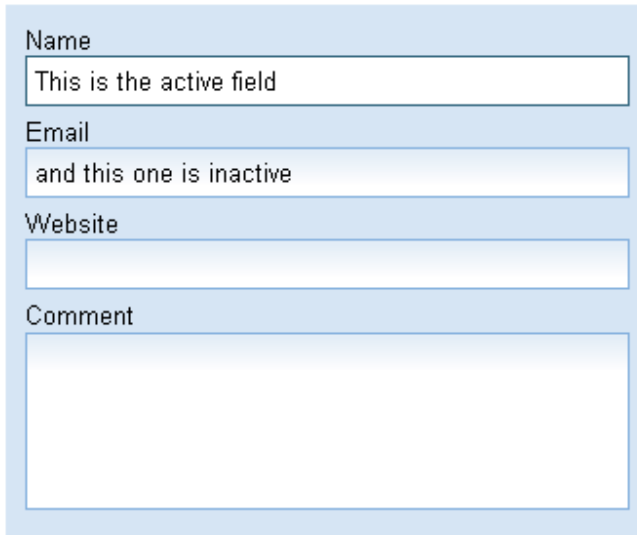
```
#inputArea input[type="text"], #inputArea textarea
{
    font-family: Arial, Sans-Serif;
    font-size: 13px;
    margin-bottom: 5px;
    display: block;
    padding: 4px;
    border: solid 1px #85b1de;
    width: 300px;
    background-image: url( 'blue_bg.png' );
    background-repeat: repeat-x;
    background-position: top;
}
```

The trick is simple and is contained in last three lines of the code. You add gradient image as background, set it to be repeated horizontally (repeat-x), and position it to the top of the field. Simple, eh?

But you can do even more! You can...

...add some behavior

But very simple. Make the active input field different. Like in the example below.



The image shows a form with four input fields. The first field, labeled 'Name', contains the text 'This is the active field' and has a blue border and white background. The second field, labeled 'Email', contains the text 'and this one is inactive' and has a light blue background and blue border. The third field, labeled 'Website', is empty and has a light blue background and blue border. The fourth field, labeled 'Comment', is empty and has a light blue background and blue border.

```
#inputArea input[type="text"]:focus, #inputArea textarea:focus
{
    background-image: none;
    background-color: #ffffff;
    border: solid 1px #33677F;
}
```

As you can see, the code is very simple. Each time a field get the focus a different styles will be applied. It changes the background and the border. But wait! This doesn't work in Internet Explorer !! So we have to call JavaScript (or jQuery) for help. Ok, you now know that I lied. You can't do this only with CSS. But, hey, it's not my false, talk to IE guys :)

Ok, to do this, we have to change our CSS slightly.

```
#inputArea input, #inputArea textarea
{
    font-family: Arial, Sans-Serif;
    font-size: 13px;
    margin-bottom: 5px;
    display: block;
    padding: 4px;
    width: 300px;
}
```

We will define all styles for all inputs and texareas except for background and border.

```
.activeField
{
    background-image: none;
    background-color: #ffffff;
    border: solid 1px #33677F;
}
.idle
{
    border: solid 1px #85b1de;
    background-image: url( 'blue_bg.png' );
    background-repeat: repeat-x;
    background-position: top;
}
```

Next, we'll define two classes that will set the styles for idle and active state. And a touch of jQuery, and voila!

```
$(document).ready(function(){
    $("input, textarea").addClass("idle");
    $("input, textarea").focus(function(){
        $(this).addClass("activeField").removeClass("idle");
    }).blur(function(){
        $(this).removeClass("activeField").addClass("idle");
    });
});
```

It is now working in IE. Of course, it is working in Firefox as well. This code does three things: initially, it adds "idle" class to all of the inputs and defined behavior for focus and blur events. Maybe not perfect, but it's working.

So, what else you can do?

Experiment

Yes, that's right, experiment with different colors, border sizes and backgrounds. You can for example also add hover functionality. Try, and get rid of boring forms!

Continuous scrolling pattern using JavaScript and ASP.NET

Posted on 19. July 2008

Continuous scrolling is very interesting pattern. If I would have to describe it in short I would say that new items are being loaded while you scroll down the content. If this sounds odd to you, don't worry. I thought it is absolutely crazy when I first read about it. But I realized people are amazed by this pattern after my session "ASP.NET patterns for better user experience" on [MSDN Day](#) in May this year. And let me just say that this pattern is used on [Google Reader](#) and [Dzone](#).

However, it is surprisingly that there is not much articles on how to do it in ASP.NET! Honestly, I found one good article [JavaScript Tutorial - Continuous Pagination](#) that reveals some details (I apologize if I missed any article, so please let me know).

Definition... a kind of

When I searched for a definition I found it under two more names: "Continuous Pagination" and "Infinite Scrolling". I couldn't agree with those names. First, this is not pagination - it is completely opposite pattern. While in pagination you click on a specific page to see some content, in Continuous scrolling you just scroll down and new content comes asynchronously. Second, I have a problem with the term infinite. It actually can't be infinite - eventually you'll come to an end of content. So let's stick with the name "Continuous scrolling".

I already mentioned that this is an opposite pattern to a well known pagination. So, when to use it?

Like pagination, this pattern should be used when there is more data than can fit in a container. And that's the only similarity. The major difference between pagination and scrolling is that user can read the content without interruption. Without clicking on a page link and waiting for response. While scrolling down, the content is being automatically loaded and rendered. So reading can go swimmingly.



Implementation

Typically, user is notified that new content is being loaded by showing indicator in the bottom right angle of a content (as you can see on the screenshot on the left).

Generally, it looks like this: Initial content is loaded on the start. When user scrolls down to the end of the content, JavaScript function calls local WebService and a set of new items is sent back to the client.

Let's make an example. Since I use Google Reader every day, let's make something that looks like it. Ok, I admit, I stole colors for this example :)

For more tutorials, live demos and other articles visit www.jankcoatwarpspeed.com

Please consider the environment before printing this eBook!

This is going to be very simple. I have a list of countries stored in SQL database. Each country has ID, Name and number of Internet users. I would like to show the complete list in a div, and to enable continuous scrolling. Image on the left shows our example.

So, I will define CountryWS that will get the data from the database. This is the first catch and might be tricky. Since new data have to be loaded on request (precisely on scroll) I have to do paging in the database. I do that by sending two parameters to my stored procedure: PageIndex and PageSize. PageIndex will tell my stored procedure which page to query, and PageSize - how many items to retrieve.

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
[ScriptService]
public class CountryWS : System.Web.Services.WebService
{
    [WebMethod]
    [ScriptMethod]
    public List<Country> GetMoreCountries(int pageIndex, int pageSize)
    {
        System.Threading.Thread.Sleep(1000);

        SampleDAO sampleDAO = new SampleDAO();
        List<Country> countries = sampleDAO.GetCountriesPaged(pageIndex, pageSize);
        return countries;
    }
}
```

That is "the server" part. Let's see what will happen on the client. All items will be rendered in a div that has onscroll handler defined.

```
<div id="content" runat="server" onscroll="OnDivScroll()">
    <!--Items will be rendered here-->
</div>
```

OnDivScroll function will do a simple job - it will determine if the user scrolled down to the near bottom of the content. Then it will call CountryWS and pass the result to HandleRetrievedData function.

```
function OnDivScroll()
{
    var content = document.getElementById('<%=content.ClientID %>');
    if(content.scrollTop < content.scrollHeight - 500)
        return;

    CountryWS.GetMoreCountriesObject(pageIndex, pageSize, HandleRetrievedData );
}
```

This is a second catch. Why near? The answer is simple - if you want your user to read smoothly, you'll have to load a new content BEFORE he/she reaches the end of the content. Ok?

And for the end, HandleRetrievedData function will parse the result and render new content.

```
function HandleRetrievedData(result)
{
```

```
var content = document.getElementById('<%=content.ClientID %>');  
  
for (var i = 0; i <= result.length - 1; i++)  
{  
    content.innerHTML += "<div class='entry'><b>" +  
        result[i].CountryName + "</b> (" + result[i].Symbol + ") - " +  
        result[i].InternetUsers + " internet user(s)</div>";  
}  
}
```

You will notice that sample code is more complex than this example, but I will let you sweat a little bit ;)

Conclusion

You saw what Continuous scrolling is and how to easily implement it using JavaScript and ASP.NET. For more information about this pattern read Ajaxpatterns.org and Continuous Scrolling pattern on ui-patterns.com.

Add grunge effect to text using simple CSS

Posted on 9. August 2008

When I first saw [CSS text gradient effect](#) I asked myself how this trick can be used to create some more effects. Actually, there are an infinite number of effects you can create - the limit is your imagination. In this short tutorial you will see how to add grunge effect to your text using just CSS and one image.

Step 1: Create a transparent PNG image

You can use [Gimp](#) or [Photoshop](#) and some of the free brushes:

- [Free Grunge Style Brushes](#)
- [BrushKing Grunge Set 1 : 14 Hi-Res Brushes](#)
- [GetBrushes.com](#)

Experiment with the brushes and color, but keep in mind that the color you use must match the color of your background. The image we are going to use in this example looks like this:



So, we are going to make a grunge effect on a black background.

Step 2: Create simple CSS

Now we are going to add grunge effect on H1 element. In order to make this work we need a little trick: place an empty span tag inside each H1 element. Be sure to add it after the text like in the example below.

```
<h1>This is one very long title with grunge effect<span></span></h1>
```

Now the CSS magic. We are going to use absolute positioning inside relative positioning (I mentioned this in my article [How to create web application user interface that looks like Outlook](#) several weeks ago). So, we are going to place absolutely positioned span element inside the relatively positioned H1 element. That way a grunge image that is actually a background image of span element will make an overlay on H1 element.

```
h1 {
    position:relative;
    color:#aeef33;
    font-family: Georgia;
}

h1 span{
    position:absolute;
    display:block;
    top:0;
    left:0;
    height:100%;
```

```
width:100%;  
background:url('grunge.png');  
}
```

And voila! This is the result:



Step 3: Experiment

Try to find some interesting brushes or patterns which you can use to create different effects. For more information about CSS text effects, read the following articles:

- [CSS gradient text effect](#) on [Web Designer Wall](#)
- [Css Text Gradient](#) on [CSS Globe](#)

Create apple.com-like breadcrumb using simple CSS

Posted on 14. August 2008

I believe you all know what breadcrumbs are. If I would have to make a simplest, rough definition I could say it is a navigation pattern that shows a path from the home page to the current one. If you are still unsure what breadcrumbs are (or my "definition" is too rough) be sure to read [Breadcrumbs](#) pattern explanation on [Welie.com](#)



In its simplest form it usually looks like this:

You are here: Home > Sample page 1 > Sample page 2 > Current page

But we are going to enhance this simple form and to create a breadcrumb that is similar to the one on [apple.com](#). You can see the example on [store.apple.com](#).

Let's create something that looks like this

As you can see on the screenshot above, breadcrumb has a soft gray gradient and a separator that looks like a right arrow. Those are the only two images that we are going to use: background gradient and a separator. The design we want to create looks like on the image below.



So what would be the best way to create a breadcrumb? Personally I think it's the best to use unordered list (UL element). So we can have a HTML structure like this:

```
<ul id="breadcrumb">
  <li><a href="#" title="Home">
    </a></li>
  <li><a href="#" title="Sample page 1">Sample page 1</a></li>
  <li><a href="#" title="Sample page 2">Sample page 2</a></li>
  <li><a href="#" title="Sample page 3">Sample page 3</a></li>
  <li>Current page</li>
</ul>
```

In the example above we have unordered list that has four list items. Each list item has a link that will point to a single page in the path, except for the last one. There is no need to point to a current page since user is already viewing it.

If you try to see this example in a browser you would see a simple bulleted list. So we have to use some CSS here to make things right.

First we'll style UL element:

```
#breadcrumb
{
    font: 11px Arial, Helvetica, sans-serif;
    background-image:url('bc_bg.png');
    background-repeat:repeat-x;
    height:30px;
    line-height:30px;
    color:#9b9b9b;
    border:solid 1px #cacaca;
    width:100%;
    overflow:hidden;
    margin:0px;
    padding:0px;
}
```

There are some attributes here that I would like to explain.

- We applied a background image to entire UL element in order to cover the entire breadcrumb area.
- Next, we set the height to 30px because that is the background image height.
- We also set line height to 30px. That will make the text appear centered vertically.
- We set the light gray color for any text inside breadcrumb. Since links will use other settings, this is going to be a color only for a current page.
- The last two lines resets the default UL settings.

Next thing we have to do is to style our list items.

```
#breadcrumb li
{
    list-style-type:none;
    float:left;
    padding-left:10px;
}

#breadcrumb a
{
    height:30px;
    display:block;
    background-image:url('bc_separator.png');
    background-repeat:no-repeat;
    background-position:right;
    padding-right: 15px;
    text-decoration: none;
    color:#454545;
}
```

For more tutorials, live demos and other articles visit www.jankoatwarpspeed.com

Please consider the environment before printing this eBook!

```
}  
  
.home  
{  
    border: none;  
    margin: 8px 0px;  
}
```

Again, I will explain these three classes. We first styled each LI element inside unordered list:

- We set list-style-type:none for each LI element in order to remove bullets
- Also, we set float:left so that our list items appear next to each other.
- We added 10px left padding to move text from the left edge of each list item

Each link inside list item will have this settings:

- 30px height and display:block will make entire list item area clickable.
- background related attributes will set the background image for each link which is separator image. It will be placed on the right side of each link.
- 15px padding will move separator to the right

The last class (home) will set the attributes for home icon.

The last thing we have to do is to add some hover effect on links:

```
#breadcrumb a:hover  
{  
    color:#35acc5;  
}
```

Believe it or not that's all.

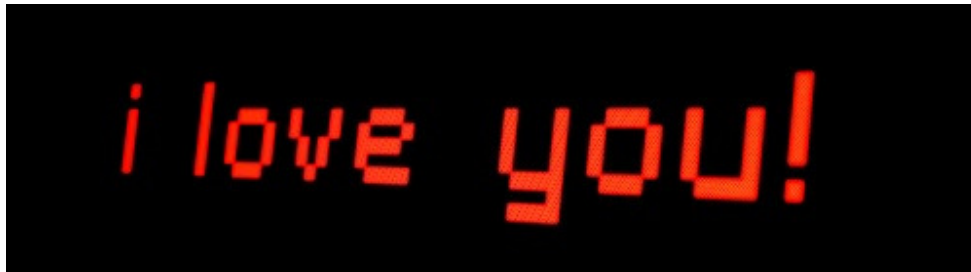
Update 17.08.2008: Some of my readers complained this isn't working in IE6 (I really think it's time to move to IE7 or Firefox). Thanks to [@lostsock](#) this is working now. You can preview [revised version](#).

Conclusion

If you review the code carefully, you can notice that you can create a variety of designs by changing background and separator images. Try and see the results!

Animate your message boxes with jQuery

Posted on 13. December 2008



A few months ago I posted an article on [creating message boxes](#). No question, you should really take care about it. But this time I want to make fun with it and add some animations using (you guess) jQuery.

The idea is more than simple. Let's animate message box that is being shown, and blow it away after we read it.

So this would be the HTML markup:

```
<div id="info">  
  This is an animated info message. Do you see it? <a href="#" class="close">Now  
  click here to make it go away!</a>  
</div>
```

And nothing more. We'll need some CSS to make it looks like something, just like in the article I mentioned earlier:

```
#info{  
  border: 1px solid;  
  margin: 10px 0px;  
  padding:15px 10px 15px 50px;  
  background-repeat: no-repeat;  
  background-position: 10px center;  
  position:relative;  
  color: #00529B;  
  background-color: #BDE5F8;  
  background-image: url('info.png');  
}
```

Cool. Now let's animate this. Although there are plenty of possibilities, I will show you just two. But I'd like you to experiment and try achieve various effects.

Blink effect

Ok, this one is pretty simple. The message will blink a few times just to let you know you should read it. Here is the code:

```
$(document).ready(function(){
    $(".close").click(function(){
        $("#info").animate({left:"+=10px"}).animate({left:"-5000px"});
    });
    $("#info").fadeOut(800).fadeIn(800).fadeOut(400).fadeIn(400)
        .fadeOut(400).fadeIn(400);
});
```

Nudge effect

Do you use MSN Messenger? You know how the chat window behaves when you send "nudge" to your colleague? We are going to do the same. Here is the code:

```
$(document).ready(function(){
    $(".close").click(function(){
        $("#info").animate({left:"+=10px"}).animate({left:"-5000px"});
    });
    $("#info").animate({left:"+=5px"},40).animate({top:"+=5px"},40)
        .animate({top:"-10px"},40).animate({left:"-10px"},40)
        .animate({top:"+=5px"},40).animate({left:"+=5px"},40)
        .animate({left:"+=5px"},40).animate({top:"+=5px"},40)
        .animate({top:"-10px"},40).animate({left:"-10px"},40)
        .animate({top:"+=5px"},40).animate({left:"+=5px"},40);
});
```

Those were my two samples. Got some other ideas?

Alternate way to select ASP.NET server controls using jQuery

Published on 8. January 2009



I guess that you know by now that if you use MasterPages in your applications client ID of controls will differ from "server" ID. That's because ASP.NET creates a new ID for controls on a page. So this is probably very familiar to you:

```
ctl00_cphContent_txtFirstName
```

You set the ID of the textbox to "txtFirstName" and ASP.NET adds "ctl00_cphContent_". Although there were some tries to prevent ASP.NET from generating unique ID's, I think it's better to use quick and "dirty" solutions :-)

So, how to select a server control using JS/jQuery? The usual way to select an element in JavaScript is to use server tags:

```
document.getElementById("<%=txtFirstName.ClientID %>");
```

You can use the same approach with jQuery:

```
$("#'<%=txtFirstName.ClientID %>'");
```

I always hated this. But, luckily, jQuery enables you to avoid server tags completely. Since we can search for element just by the part of the name, we can do the next:

```
$("[id$='_txtFirstName']");
```

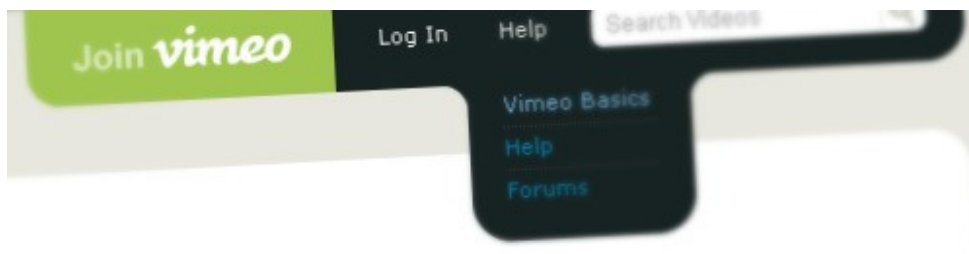
This will find elements which id's ends with "_txtFirstName". But why the dash in the front of a name? This will ensure you selected the element by its full control ID, not just a part of it.

Pretty simple, isn't it? The only issue here would be if you have controls with the same name placed in different content pages.

So, what do you think? Is this too dirty for you?

Create Vimeo-like top navigation

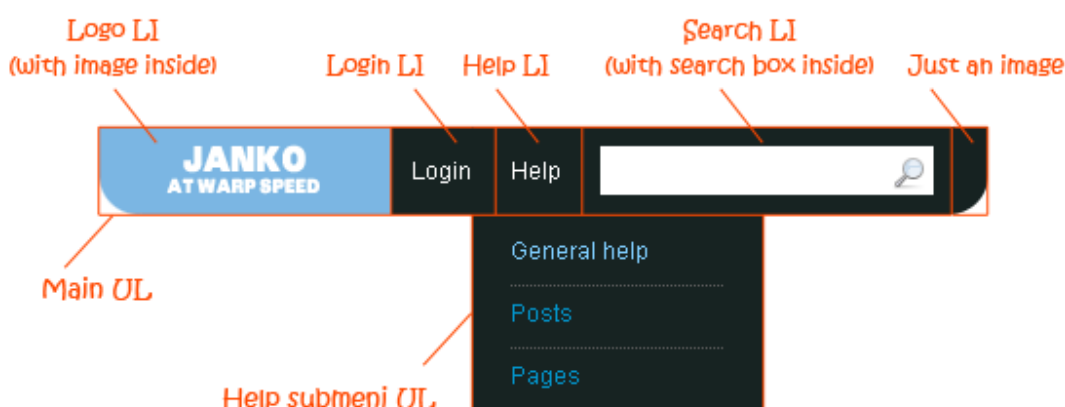
Posted on 19. January 2009



I really like the top navigation implemented on Vimeo.com. First time I saw it I wanted to recreate it. And this is exactly what I am going to do in this tutorial.

What I like the most is the menu that drops down when you hover search box. It offers you different search options that you can choose and narrow your search.

The base for this tutorial is simple CSS drop down menu based on unordered list. The structure is visually described in the image below:



As you can see we have UL with four items. The first one is logo with short submenu. Then comes login link, Help link with submenu and search item with submenu. Each submenu is being shown when hover corresponding link.

So this is the base structure that we'll use here:

```
<ul id="menu">
  <li class="logo">
    
    <ul id="main">
      <li>Welcome to <b>Create Vimeo-like top navigation</b> tutorial!</li>
    </ul>
  </li>
```

For more tutorials, live demos and other articles visit www.jankotwarpsspeed.com
Please consider the environment before printing this eBook!

```

<li><a href="#">Login</a>
</li>
<li><a href="#">Help</a>
  <ul id="help">
    <li><a href="#">General help</a></li>
    <li><a href="#">Posts</a></li>
    <li><a href="#">Pages</a></li>
  </ul>
</li>
<li class="searchContainer">
  <div>
    <input type="text" id="searchField" />
    </div>
    <ul id="search">
      <li><input id="cbxAll" type="checkbox" />All</li>
      <li><input id="Articles" type="checkbox" />Articles</li>
      <li><input id="Tutorials" type="checkbox" />Tutorials</li>
      <li><input id="Reviews" type="checkbox" />Reviews</li>
      <li><input id="Resources" type="checkbox" />Resources</li>
    </ul>
  </li>
</ul>


```

And CSS styles:

```

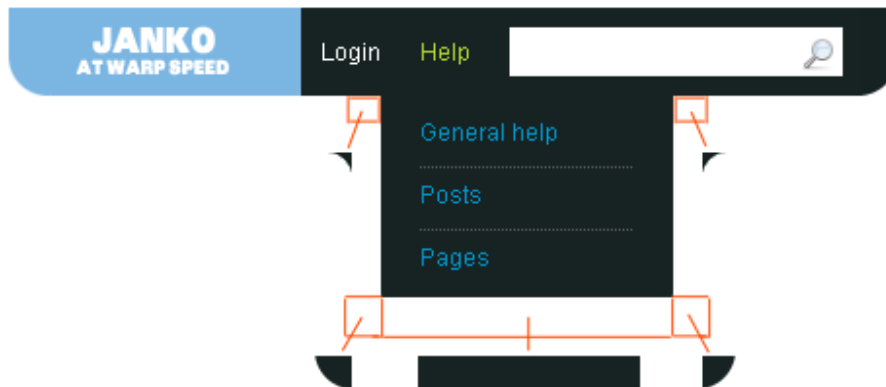
/* menu */
#menu{ margin:0px; padding:0px; list-style:none; color:#fff; line-height:45px;
display:inline-block; float:left; z-index:1000; }
#menu a { color:#fff; text-decoration:none; }
#menu > li {background:#172322 none repeat scroll 0 0; cursor:pointer; float:left;
position:relative; padding:0px 10px;}
#menu > li a:hover {color:#B0D730;}
#menu .logo {background:transparent none repeat scroll 0% 0%; padding:0px;
background-color:Transparent;}
/* sub-menus*/
#menu ul { padding:0px; margin:0px; display:block; display:inline;}
#menu li ul { position:absolute; left:-10px; top:0px; margin-top:45px; width:150px;
line-height:16px; background-color:#172322; color:#0395CC; /* for IE */ display:none;
}
#menu li:hover ul { display:block;}
#menu li ul li{ display:block; margin:5px 20px; padding: 5px 0px; border-top: dotted
1px #606060; list-style-type:none; }
#menu li ul li:first-child { border-top: none; }
#menu li ul li a { display:block; color:#0395CC; }
#menu li ul li a:hover { color:#7FCDFE; }
/* main submenu */
#menu #main { left:0px; top:-20px; padding-top:20px; background-color:#7cb7e3;
color:#fff; z-index:999;}
/* search */
.searchContainer div { background-color:#fff; display:inline; padding:5px;}
.searchContainer input[type="text"] {border:none;}
.searchContainer img { vertical-align:middle;}

```

For more tutorials, live demos and other articles visit www.jankowarpspeed.com

Please consider the environment before printing this eBook!

But as you can see this is far away from the good looking Vimeo navigation. It is functional, of course, but we need that pretty rounded corners everywhere. The solution is actually simple and it is described in the image below:



Now that looks fine. What we actually did can be seen in the code below. Let's take Help submenu for the example:

```
<li><a href="#">Help</a>
  <ul id="help">
    <li>
      
      <a href="#">General help</a>
      
    </li>
    <li><a href="#">Posts</a></li>
    <li><a href="#">Pages</a></li>
    <li class="last">
      
      
      
    </li>
  </ul>
</li>
```

We added two absolutely positioned images inside the first LI that will create “shoulders”. Also, we added one more LI to the end of the list that contains two absolutely positioned corners and one 1x1px stretched image to fill the empty space. And this is the additional styles that you need:

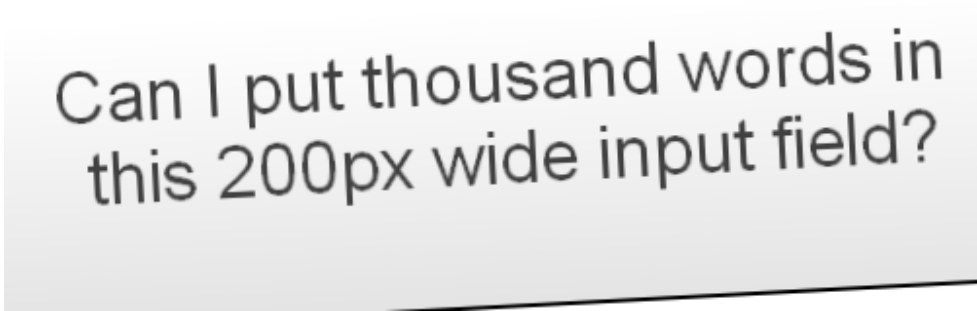
```
/* corners*/
#menu .corner_inset_left { position:absolute; top:0px; left:-12px;}
#menu .corner_inset_right { position:absolute; top:0px; left:150px;}
#menu .last { background:transparent none repeat scroll 0% 0%; margin:0px;
padding:0px;
  border:none; position:relative; border:none; height:0px;}
#menu .corner_left { position:absolute; left:0px; top:0px;}
#menu .corner_right { position:absolute; left:132px; top:0px;}
#menu .middle { position:absolute; left:18px; height: 20px; width: 115px; top:0px;}
```

Now we have functional AND good looking top navigation. This might be optimized, so if you have free time or you need it for your projects, step on it. Or you can use this one as is :)

Did you implement something similar in any project? Do you find this one useful for some of your future projects?

Enlarge input fields on focus

Published on 6. March 2009



Some time ago client insisted to be able to see all 500 characters while typing the text in input fields. Ok, reasonable. But they also insisted that web form should take the minimum space and that input fields should be "single line". Errr... "Oh I see", I said "so you want to make an oil painting using pencil, right? You can actually do it with JavaScript."

Doesn't matter if request sounds stupid or not, a solution is always needed. So I suggested enlarging input fields on focus so that entire text could be visible during typing. I wasn't sure about usability, though, but the client was satisfied. The solution was more than simple, and this is the example of it (I didn't make a demo because this is soooo simple):

```
$("#textarea").focus(function(){
    $(this).width = 600; // enlarge width
}).blur(function(){
    $(this).width = 200; // return to original value
});
```

You can animate this as well:

```
$("#textarea").focus(function(){
    $(this).animate({ width:"600px"}, 1000); // enlarge width
}).blur(function(){
    $(this).animate({ width:"200px"}, 1000); // return to original value
});
```

Now, this might be a solution for many comment forms on various blogs because your website URL might not fit entirely in input field and this can be frustrating. The reason is simple - I, for example, submit my articles to many websites and my browser remembers my inputs, of course. So when I want to enter URL on some comment form I get this:

A screenshot of a web form for submitting a comment. It consists of three input fields and a submit button. The first field is labeled 'Name (required)' and contains the text 'Janko'. The second field is labeled 'email (will not be published) used for Gravatars (required)' and contains the text 'this@jankoatwarpspeed.c'. The third field is labeled 'Website' and contains the text 'http://www.jankoatwarpspe'. A dropdown menu is open over the Website field, showing a list of identical URLs: 'http://www.jankoatwarps...'. Below the fields is a button labeled 'Submit Comment'.

Which one to choose?

Oh, yes, input fields on these forms could be simply just larger, but hey... any suggestions?

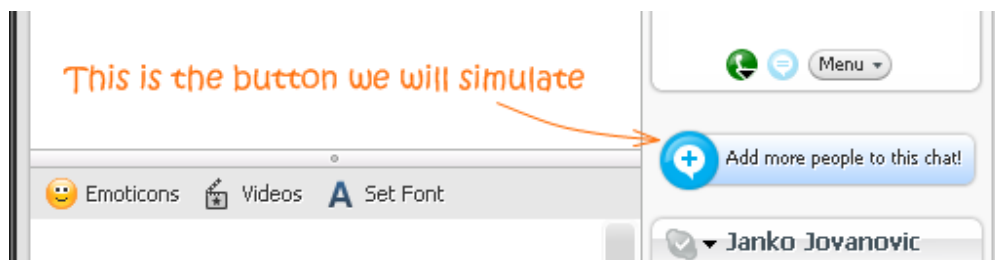
How to create Skype-like buttons using jQuery

Published on 11. March 2009



If you use Skype I am sure that you noticed that animated button for adding more people to a chat. When you click on it the icon on the left "jumps" for a few times. I love that animation. And that's why I'm going to show you how to create the same button using jQuery and some simple CSS.

If you are not sure what button am I talking about, image below might help you.



And this is how our button will look like:



Ok, the task here is quite simple - I want the icon to jump for a few times when I hover the button. (In Skype window this icon jumps when you click on it, but I think it would be much better to have it jump on hover)

The button itself consists of an image and text placed inside of a link.

```
<a class="button" href="#">
  Send info</a>
  or <a href="#">cancel</a>
</a>
```

Let's style the button. CSS code for this is simple and won't go through all the classes here. The key is that the icon is absolutely positioned inside its relatively positioned container - the link. The position of the icon is set to simulate Skype button, and that is to cover the left side of the button. Please note that you will see the rounded corners on the button only in Firefox thanks to `-moz-border-radius-` properties.

```
.button {
    padding: 4px 10px 3px 25px;
    border: solid 1px #8AB134;
    position: relative;
    cursor: pointer;
    display: inline-block;
    background-image: url( 'bkg.png' );
    background-repeat: repeat-x;
    font-size: 11px;
    height: 16px;
    text-decoration: none;
    color: #40740D;
    -moz-border-radius-bottomleft: 5px;
    -moz-border-radius-bottomright: 5px;
    -moz-border-radius-topleft: 5px;
    -moz-border-radius-topright: 5px;
}
.button img {
    position: absolute;
    top: -4px;
    left: -12px;
    border: none;
}
.button:hover {
    color: #8AB134;
}
```

Now the animation. We have three jumps on original Skype button - one large and two small jump. In first jump, icon will go up for 6px, in second 3px and in the last jump 2px. After each jump, the icon will go back to its original position defined in CSS. I guess it is pretty much similar to the Skype button.

```
$(document).ready(function(){
    $(".button").hover(function(){
        $(".button img")
            // first jump
            .animate({top:"-10px"}, 200).animate({top:"-4px"}, 200)
            // second jump
            .animate({top:"-7px"}, 100).animate({top:"-4px"}, 100)
            // the last jump
            .animate({top:"-6px"}, 100).animate({top:"-4px"}, 100);
    });
});
```

Code is tested in Firefox, Safari and IE7. If this doesn't work in IE6, I simply [don't care](#).

The icon used in this tutorial is from [Developpers Icons](#) set. Now try the demo, download the source code and enjoy styling your new buttons!