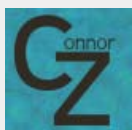




# Build An Incredible Login Form With jQuery

Aug 19th in JavaScript & AJAX by Connor Zwick

One struggle that still remains today in web design is displaying all of the redundant information on every page. For example, a login form. What if there was a way to easily make the content accessible on every page, but keep it hidden until needed? Well you can, by making a top panel that when clicked, will reveal its self and its content. But we need to make this look nice, so we'll also animate it.



Author: Connor Zwick

Hi, I'm Connor Zwick-freelance blogger and designer. I love learning and teaching new topics in areas like HTML, CSS, PHP, Ruby on Rails, and last but certainly not least, jQuery. You can follow me on [twitter](#) or visit his design blog [cyberantix](#).



In this tutorial, we'll create a sliding panel, that slides in to reveal more content, using JQuery to animate the height of the panel. In this case, we will be creating a hypothetical login for the new tutstplus area that's coming soon.

## Step 1 Photoshop Layout

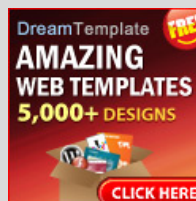
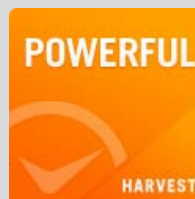
First we need to plan out our layout and make it look cool with Photoshop. Thanks to Collis, and his amazing Photoshop skills, we have a slick layout to work with. You can grab the before and after PSD file for further inspection in the source zip file. But it's pretty self-evident. It doesn't have any gradients either, so we should be able to make this fairly easily just with CSS.



Plus

Source Files, Bonus Tutorials and More for \$9 p/month

LEARN MORE & SIGNUP | LOGIN HERE





Here you can see what the demo will look like in its normal state.



Here is what the demo will look like when the panel is slid down.

## Step 2 --Planning the structure

First off, we need to build the page's structure. To create the layout above, what do we all need structurally in the HTML?

- First off, we need a header, that extends across the entire width of the page.
- We'll also need a top panel, which for now, we will pretend is permanently expanded until we insert the JQuery.
- We'll need an area for the normal page content.
- Finally we'll need a visual break between the header and page content, which we will accomplish through a horizontal ruler (hr).

Alright, so the layout of the page is pretty simple. Here it is:

```
1. <div></div> <!--Header-->
```

**Subscribe**  
Subscribe to an update feed:

FULL RSS SHORT RSS  
TWITTER EMAIL  
LINK FEED FREEBIES  
VIDEOS

45856 readers  
BY FEEDBURNER

Confused? [Click Here!](#)

**Advertise Here**

**Write for us!**  
We pay \$150 for user submitted tutorials and articles that we publish. Anyone can send in a contribution

[LEARN MORE](#)

**User Plus**  
Community Link Feed

[25 3D Typography Tutorials](#)  
3D Typography is an art of collecting and arranging text and characters in three dimension view. So here we have collected some of the tutorials on 3D Typography

```

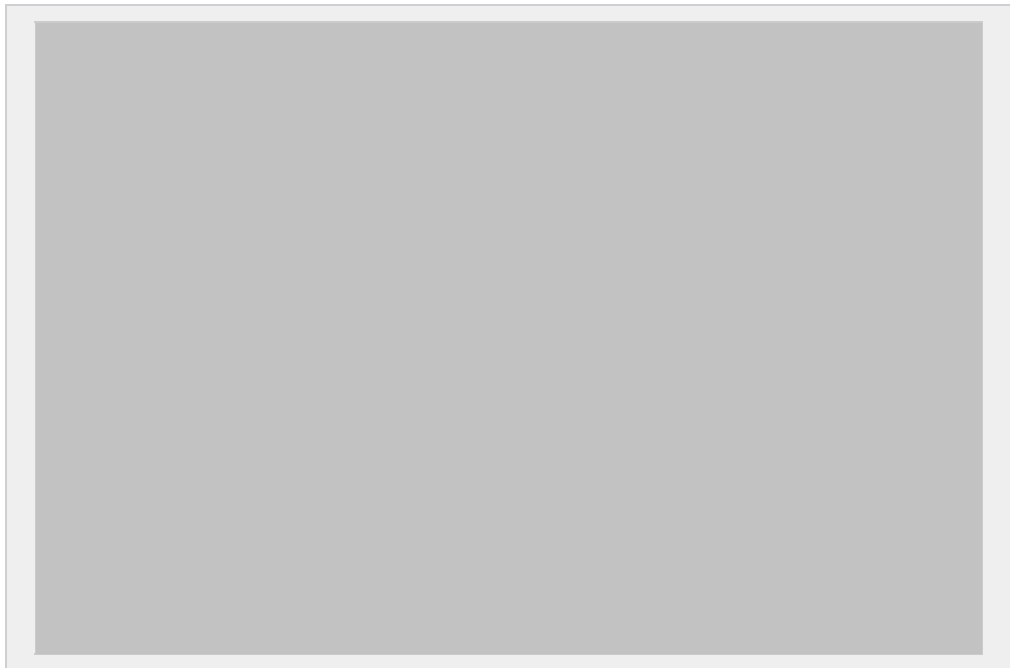
2. |
3. | <hr> <!--Header Stripe-->
4. |
5. |
6. | <div> <!--Contains the button and panel-->
7. | <div> <!--Contains the panel itself-->
8. |
9. | <div> This div will serve as the background of the panel</div>
10. |
11. | </div>
12. | <div><a>Login Here</a></div> <!--Will be the button to slide the panel down-->
13. |
14. | <div><a>Hide</a></div> <!--Toggles to this when the panel is slid down-->
15. | >
16. | </div>
17. | </div>
18. |
19. | <div>
20. |
21. | All of the Content will go here
22. | </div>

```

Wow...without classes or any content inside, it looks like a lot of pointless divs, but all will be necessary for the CSS and JQuery later on. Now we will start adding classes in preparation for the CSS.

## Step 3 -- CSS preparation: Classes & ID's

Now we've got to change the skeleton into an actual site with CSS. We'll start by adding classes and ID's to all of those divs! You can do this easily by printing out the Photoshop layout and then marking up the areas and associated classes with a sharpie. For this demonstration, I will do the same only in Photoshop. Although it may be extremely ugly, hopefully it will show you the different regions of the page.



Note: I plan on having the normal non-highlighted image on hover.

Here is the page with the added classes and ID's:

view plain copy to clipboard print ?

### 10 Of The Best Insects Kindom Photo Manipulation

Today we are going to showcase some of the interesting images that have been manipulated specifically for the insects kingdom. Without any delays here are the winning best images in this category.

### Golden Aliminium Text Effect Tutorial

Golden Aliminium Text Effect Tutorial. The tutorial is in Turkish but you can use Google Translate on my website.

### Mind Boggling And Extraordinary Wordpress Themes

Creating good Wordpress themes is becoming more popular now with the introduction of new versions of wordpress that help designers to achieve highets that were not possible. Here are some post gathered from around the internet with some exceptional and extraordinary wordpress theme designs.

### Free Bulk Sms Software

Download pocket PC to mobile mass messaging software that sends messages, business news, invitation and other important information simultaneously to number of users.

### 40+ New And Fresh Blogger Templates

Showcase of some neat, new and fresh blogger templates to pimp up your blog.

### Online CSS Sprite Builder / Generator

css-sprite.es is an online CSS Sprites Builder / Generator This tool can, given a few image files, generate a clean image sprite with corresponding valid CSS and XHTML code along with an optional hover processing for rollover effects.

### Theatrical Movie Trailer With JQuery

Don't you wonder sometimes, when looking at something, how you can create something like that yourself? I have to admit that I do. This is why I came up with the Skype for Mac and the iPhone examples in your webbrower. Today, we're trying something new in this Proof of Concept category: creating a theatrical movie trailer with jquery

### Introducing Animated JPEG

An introduction to JPEG Animated file extension.

### FireQuery Addon Screencast

Digging into the FireQuery Add-on: a screencast by Elijah Manor

SUBMIT MORE SUBSCRIBE



```

1. | <div id="header">
2. | </div>
3. |
4. | <hr id="header_stripe"/>
5. |
6. | <div id="wrapper">
7. |   <div id="toppanel">
8. |
9. |     <div id="panel">
10. |       <div id="panel_contents"> </div>
11. |     </div>
12. |     <div class="panel_button"><a href="#">Login Here</a></div>
13. |
14. |     <div class="panel_button"><a href="#">Hide</a></div>
15. |   </div>
16. | </div>
17. | <div id="content">
18. |
19. | </div>

```

Right now, I'd show you a screenshot of what we have so far, but we don't have anything except a horizontal ruler and two unstyled links. You get the idea. Now we can style the page.

## Step 4 -- Linking the files together

Before we go any further though, we need to introduce the CSS file to the skeleton. I created a stylesheet entitled "style.css". While we're adding code to the head, we might as well add the javascript and jQuery as well. Here is the head of the page:

```

view plain copy to clipboard print ?
1. | <head>
2. | <title>Nettuts JQuery Sliding Panel</title>
3. | <style type="text/css">
4. | @import url(style.css);
5. | </style>
6. | <script src="jquery.js" type="text/javascript"></
7. | <br>
8. | script> <script src="javascript.js" type="text/javascript"></><br><script>
9. | </head>

```

## Step 5 -- Styling the Skeleton: header

Now we have to style that skeleton of divs. Let's start from the top down. First we need to style the header as well as the body tag:

```

view plain copy to clipboard print ?
1. | body {
2. |   background: #202020;
3. |   text-align: center;
4. |   margin: 0px;
5. | }
6. |
7. | #header {
8. |   margin-left: auto;
9. |   margin-right: auto;
10. |   width: 100%;
11. |   height: 135px;

```

```
12. | background: #3f3f3f url(images/header.png) no-repeat center ;
13. | position: relative;
14. | border-bottom: 1px solid #4a4a4a;
15. | }
```

Fortunately, we have no gradients to worry about here. But we do still have a background image. I also added a 1px border to the bottom of the header for a visual break.

The background image is optional. I liked the Bell Gothic BT font so much, I decided to make it into an image. Alternatively, you can choose to just style plain text by adding styling to h1, and h2 tags:

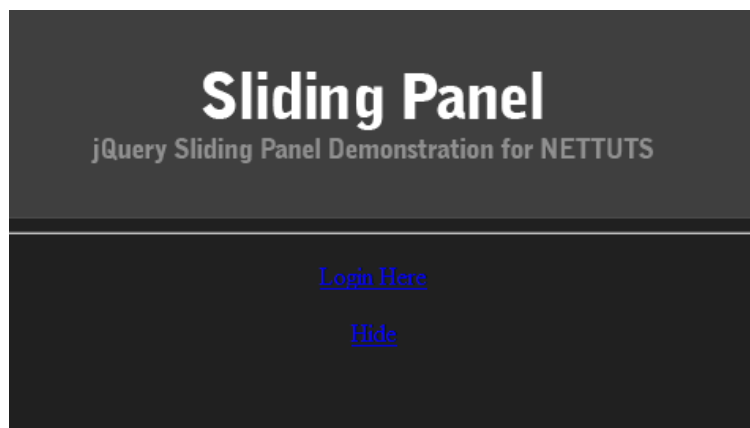
```
view plain copy to clipboard print ?
1. | #header h1{
2. | font-family: Arial, Helvetica, sans-serif;
3. | font-weight: bold;
4. | position: relative;
5. | top: 30px;
6. | font-size: 40px;
7. | color: white;
8. | }
9. |
10. | #header h2{
11. | font-family: Arial, Helvetica, sans-serif;
12. | font-weight: bold;
13. | font-size: 16px;
14. | color: #7e7e7e;
15. | }
```

And then modifying the header to this:

```
view plain copy to clipboard print ?
1. | <div id="header">
2. |
3. | <h1>Sliding Panel</h1>
4. | <br />
5. | <h2>jQuery Sliding Panel Demonstration for NETTUTS</h2>
6. | </div>
```

So now the page should look like this:

You can view step 5 [here](#).



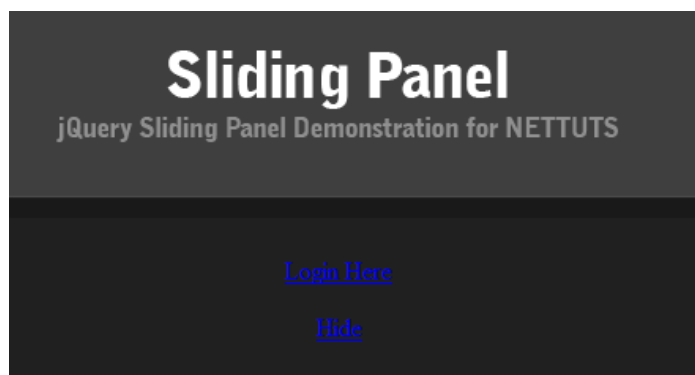
## Step 6 -- Styling the Horizontal Ruler

Although we have the bottom border of the header to visually separate the sections, we also need a thicker more visual border as well. Since we cannot apply two bottom borders to the header, we can just stylize the horizontal ruler (hr):

```

view plain copy to clipboard print ?
1. hr#header_stripe{
2. height: 12px;
3. position: relative;
4. top: -7px;
5. background-color: #191919;
6. border: none;
7. color: #191919;
8.
9. }
```

We now have a thicker separation to add to the 1px border:



You can view step 6 [here](#).

## Step 7 -- Styling the Panel

Now we need to stylize the panel. Until we add the JQuery, we're going to stylize the panel like it was expanded. When we're done with the CSS, we're going to animate the height of the panel to zero, and then back to full height; so we need to make sure that when we change the height, it stays the same.

Here is the CSS code, I'll explain it afterwards:

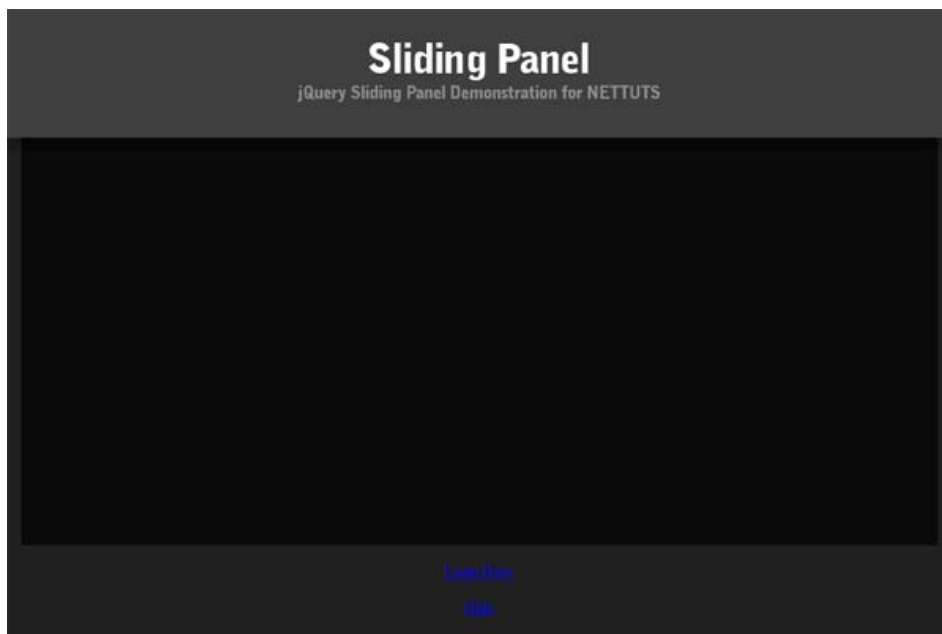
```

view plain copy to clipboard print ?
1. #wrapper{
2. margin-left: auto;
3. margin-right: auto;
4. width: 900px;
5. text-align: center;
6. }
7. #toppanel {
8. position: absolute;
9. top: 135px;
10. width: 900px;
11. z-index: 25;
12. text-align: center;
13. margin-left: auto;
14. margin-right: auto;
15. }
16. #panel {
17. width: 900px;
```

```

18. | position: relative;
19. | top: 1px;
20. | height: 400px;
21. | margin-left: auto;
22. | margin-right: auto;
23. | z-index: 10;
24. | overflow: hidden;
25. | text-align: left;
26. | }
27. | #panel_contents {
28. | background: black;
29. | filter:alpha(opacity=70);
30. | -moz-opacity:0.70;
31. | -khtml-opacity: 0.70;
32. | opacity: 0.70;
33. | height: 100%;
34. | width: 904px;
35. | position: absolute;
36. | z-index: -1;
37. | }

```



Ok, that's a lot of code for one box. Well it's more than that. Try inspecting it with either Firefox Extension Firebug or Web Developer, and you will see what all that CSS does.

Check out what [Step 7](#) currently looks like.

- First off, we need the panel to be positioned absolute, or else, when expanded, it will push all of the content below it, down. So we add a wrapper, which is wrapped around everything else, and then centered. If we left the wrapper out, the panel, which is positioned absolute, would not be able to be centered as easily.
- Next, we add the style info for the toppanel as a whole. As you can see, this includes the panel buttons.
- After that, we add the style info for just the panel which is hidden normally. This is the box that you see expanded now. I made the height 100%, so that if we increase or decrease the height of #toppanel, then the height of the #panel will be the same. Also, the overflow is hidden, so that if the height of the #toppanel is lowered, it will cut of the content of the panel.
- If you examine the earlier HTML, you will see the div with the ID of panel\_contents. This div, although empty, allows us to have the background transparent, while still keeping the content inside opaque.

## Step 8 -- Add content to the Panel

Before we test out the panel, we need to add some content, to see if it hides it properly. In this example, we are making a login area, so we need to add a form, and we're also adding an image to balance it. This step is just to add content for

the demo. It is less important and is more basic, so I will not explain it as much as everything else. Here is the code:

#### CSS:

```

view plain copy to clipboard print ?
1. .border {
2.     border: 15px #1d1d1d solid;
3. }
4.
5. img.border_pic {
6.     border: 15px #1d1d1d solid;
7.     position: absolute;
8.     top: 110px;
9.     float: left;
10.    margin-left: 150px;
11.    width: 250px;
12.    height: 150px;
13.    z-index: 30;
14. }
15. div#login {
16.    width: 240px;
17.    height: 150px;
18.    position: absolute;
19.    rightright: 150px;
20.    top: 110px;
21.    background: #46392f;
22.    text-align: left;
23.    padding-left: 10px;
24. }
25. div#login p {
26.    color: #CCCCCC;
27.    font-family: Century Gothic, Georgia, "Times New Roman", Times, serif;
28.    line-height: 25px;
29. }
30. div#login input#password {
31.    position: relative;
32.    rightright: -6px;
33. }
34. div#login input#login_btn {
35.    border: 1px #899690 solid;
36.    cursor: pointer;
37.    position: relative;
38.    top: 30px;
39.    left: 86px;
40. }

```

#### HTML:

```

view plain copy to clipboard print ?
1. 
2. <div class="border" id="login">
3.
4.     <p>Username:
5.         <input type="text" size="15" name="username" id="username" />
6.
7.     <br />
8.     Password:
9.         <input type="password" size="15" name="password" id="password" />
10.

```



```
11. | <br />  
12. | <input type="button" accesskey="l" id="login_btn" name="login" value="Login" />  
13. |  
14. | </p>  
15. | </div>
```



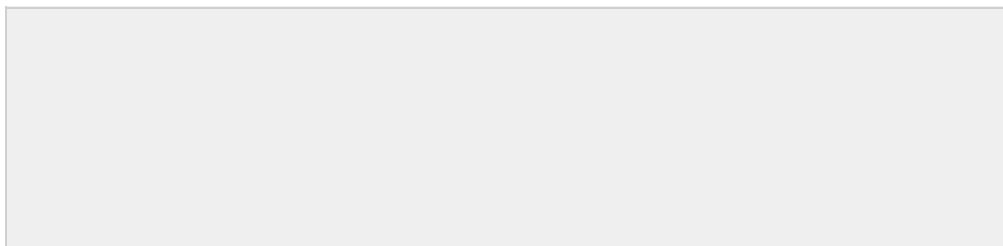
Step 8 is available [here](#).

## Step 9 -- Test out the CSS

We now need to make sure that if we use jQuery to animate the height of the top panel, it will work smoothly. Now that we have content, we are going to change the height of #panel to 200 and see what happens:



Wonderful. You can view step 9 [here](#). Now we're going to change it to 0:





Perfect. Now we know that the design will work with JQuery.

## Step 10 -- Styling the Button

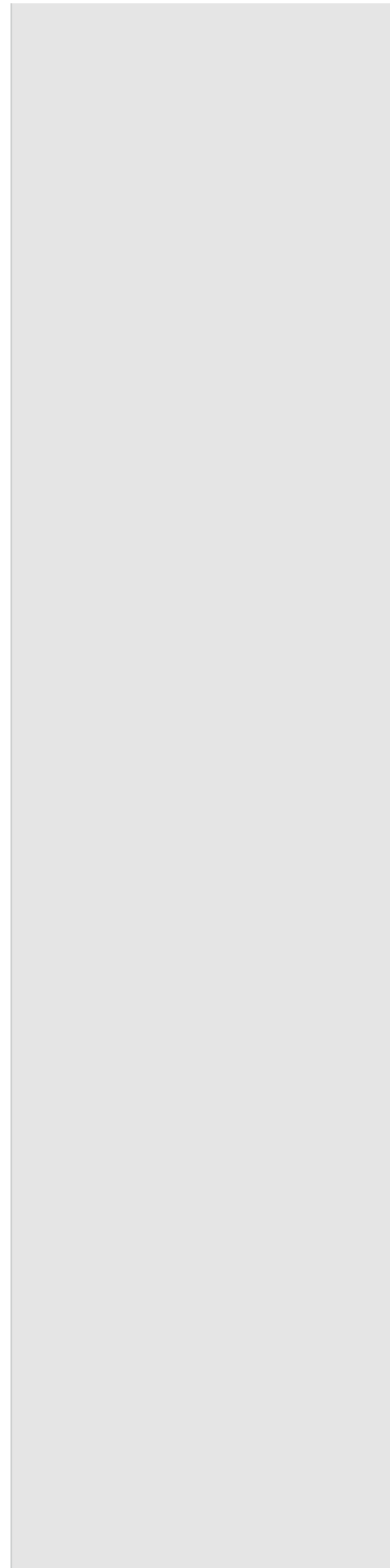
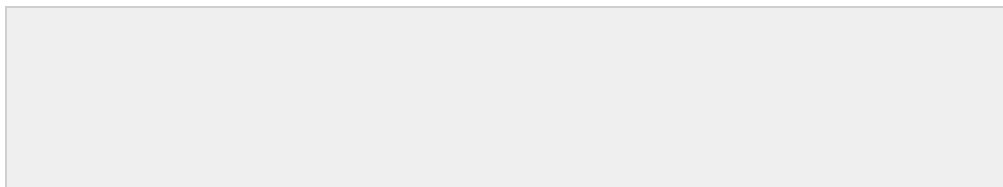
If you examine the finished product, you can see that the button that slides the panel down, changes once you click it once. This means it toggles. Therefore, we need two buttons, and we will toggle their visibility. Before we hide one of them, though, we need to add CSS to them.

If you remember, we added the class ".panel\_button" to them. Here is the style information. I will explain it after:

```

view plain copy to clipboard print ?
1. .panel_button {
2.     margin-left: auto;
3.     margin-right: auto;
4.     position: relative;
5.     top: 1px;
6.     width: 173px;
7.     height: 54px;
8.     background: url(images/panel_button.png);
9.     z-index: 20;
10.    filter:alpha(opacity=70);
11.    -moz-opacity:0.70;
12.    -khtml-opacity: 0.70;
13.    opacity: 0.70;
14.    cursor: pointer;
15. }
16. .panel_button a {
17.     text-decoration: none;
18.     color: #545454;
19.     font-size: 20px;
20.     font-weight: bold;
21.     position: relative;
22.     top: 5px;
23.     left: 10px;
24.     font-family: Arial, Helvetica, sans-serif;
25. }
26. .panel_button a:hover {
27.     color: #999999;
28. }

```





### Step 10 Panel Buttons

- First we center the button using the auto margin technique. Then we position it and add a background of the button. We also add all of that styling information to accommodate for all of the different browser's preferences. And make the button seem clickable by making the cursor a pointer, when you hover over it. This just improves the usability.
- We're also going to wrap the text in a link, to provide an on hover effect, as well as positioning.

## Step 11 Button HTML

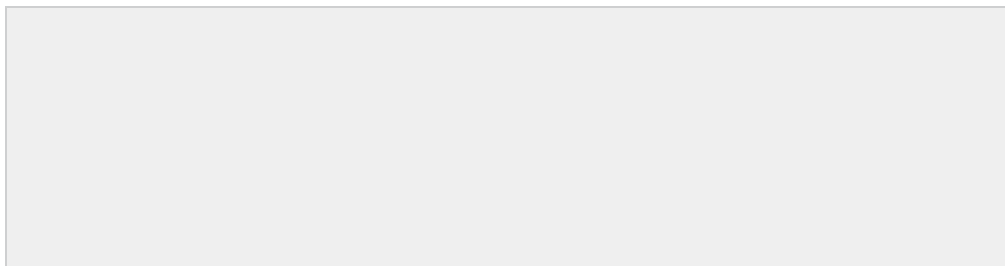
Now, in preparation for the JQuery, we need to set up the buttons, with their HTML. First off we're going to add an image to each button, and position it with CSS, you'll see the HTML in a second:

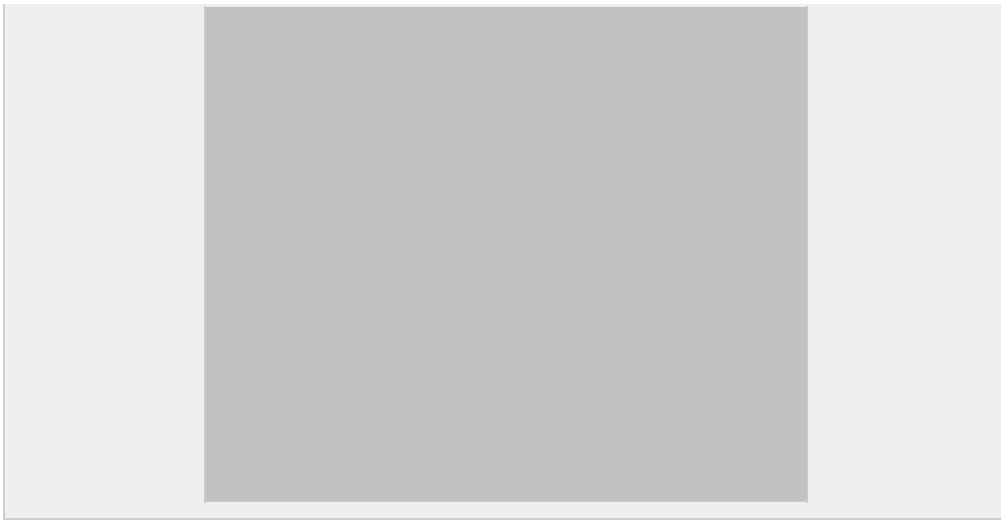
```
view plain copy to clipboard print ?
1. .panel_button img{
2. position: relative;
3. top: 10px;
4. border: none;
5. }
```

Now, we also need to hide the Hide button for now. As much as I hate, inline styling, I think it is just easier to add this CSS inline, so here is the new HTML code for the buttons, with the images:

```
view plain copy to clipboard print ?
1. <div class="panel_button" style="display: visible;">Login Here</a>
3.
4. </div>
5. <div class="panel_button" id="hide_button" style="display: none;">Hide</a>
8. </div>
```

Ok, so notice, right now, the hide button is hidden with inline styling. This will be toggled later with jQuery. Notice, I also added an ID to the second button, so we can target it later easily.





[Step 11](#) Panel Button

## Step 12 Adding the Content

This is a quick, but necessary step, adding content. I wrote one sentence and added one paragraph of dummy text. I centered it using the auto margin technique, and colored it a gray color:

```
view plain copy to clipboard print ?  
1. #content {  
2.   margin-left: auto;  
3.   margin-right: auto;  
4.   width: 600px;  
5.   position: relative;  
6.   top: 90px;  
7.   text-align: left;  
8.   color: #545454;  
9.   font-family: Arial, Helvetica, sans-serif;  
10.  font-size: 12px;  
11.  padding-bottom: 30px;  
12. }
```



See the text behind the panel in [Step 12](#).

## Step 13 JQuery Time!

Ok, now for the final part of the tutorial, JQuery! You can grab the latest JQuery at [jquery.com](http://jquery.com). If you're just beginning

with it, be sure to check out this other Nettuts tutorial by Jeffrey Way, for great JQuery resources. I've already grabbed a copy from JQuery.com, and have already linked it to the page in Step 4.

## Step 14 Think about what we need

Lets first think about what we need the JQuery to do, before we write the code.

- We want to activate the animation on 'div.panel\_button' click.
- We then want to animate the height of the panel to 400px.
- We then want to toggle the button.
- Then we want to activate the reverse animation on 'div#hide\_button' click.
- Then we want to animate the height back to 0px

## Step 15 Write the Code

So first we start out by getting the script ready with the following JQuery:

```
view plain copy to clipboard print ?
1. $(document).ready(function() {
2. });
3.
4. Now we write the code that goes inside there:
5.
6. $(document).ready(function() {
7.     $("div.panel_button").click(
8. <wbr>
9.         function(){ $("div#panel").animate({ height: "400px" }); $("div.panel_button"
10.     ).toggle(<wbr>;
11.     }); $("div#hide_button").click(<wbr>function(){
12.         $("div#panel").animate({
13.             height: "0px"
14.         });
15.     });
16. });
```



Panel in motion in [Step 15](#).

At first, when you examine the previous code, some of you might wonder why I only have toggle in the first action. Well, you need to remember that the hide button also has a class of panel\_button. Therefore, when you click the 'Hide' button, you are actually applying both actions.

## Step 16 Making the animation look 'real'

So now it's looking pretty good, but we can still do more; like making the animation look better. When animating, it's usually important to try to imitate real life as much as possible. In this case, I think of a real life example, like a pull-down projection screen. Remember when you pull those, you pull it further down then it will be, then it goes back up a little. In the same way, when you want to put it back up, you pull it down a little before it goes up very fast.

Let's try to imitate that:

```
view plain copy to clipboard print ?  
1. $(document).ready(function() {  
2.     $("div.panel_button").click(  
3.     <wbr>  
4.     function(){ $("div#panel").animate({ height: "500px" }) .animate({  
5.     height: "400px" }, "fast"); $("div.panel_button").toggle()<wbr>;  
6.     }); $("div#hide_button").click(<wbr>function(){  
7.     $("div#panel").animate({  
8.     height: "0px"  
9.     <wbr>  
10.     }, "fast");  
11.     });  
12. });
```

Notice that we animate the panel to a height of 500 before going to 400. We also added a difference of speed like in real life, by making certain parts slower. If you look at the demo you will see that when you hide the panel, it still goes to 500 first. Again, this is because both buttons have the same class. So really when you hide the panel, it goes through this process:

- Animate to 500 and toggle
- Animate back to 400 fast
- Animate back to 0 fast

Now we have a working Sliding Panel using jQuery. Hope you found this tutorial to be useful! If so, please submit it to Digg, StumbleUpon, DZone, etc!

## Related Posts

Check out some more great tutorials and articles that you might like

[How to Build a Super Duper News Scroller](#)

Screencasts

[Create a Photo Admin Site Using PHP and jQuery](#)

Screencasts

[How to Mimic the iGoogle Interface](#)

JavaScript & AJAX

## Enjoy this Post?

Your vote will help us grow this site and provide even more awesomeness

A row of social sharing buttons. From left to right: a '4 tweets' button with a 'retweet' button below it; a '2 diggs' button with a 'digg it' button below it; a 'SUBMIT DZone' button; a 'submit to reddit' button; and a StumbleUpon logo button.

**Plus Members**  
Source Files, Bonus Tutorials and More for \$9 a month for all TUTS+ sites in one subscription.  
[JOIN NOW](#)