



CSS

Creating a modal window with HTML5 & CSS3



By [Keenan Payne](#) • Posted Oct. 02, 2012 • Reading time: 6 minutes



Modal boxes are a frequently utilized tool in the web developer's arsenal. Used for, amongst many things, login/register forms; advertisements; or just notifications to the user.

However despite the fact that modal windows frequently contain mission critical information, they are routinely created with JavaScript, which does not sit

well with the best-practices of progressive enhancement or graceful degradation.

This doesn't need to be a problem, because HTML5 & CSS3 allow us to create modal windows with ease.

DEMO

What we are going to do is use CSS3's transition, opacity, pointer-event, and background gradient properties to create a very beautiful and functional modal box.

You can [view a demo here](#).

THE HTML

The first step to creating our modal box is this short but sweet markup:

```
<a href="#openModal">Open Modal</a>

<div id="openModal" class="modalDialog">

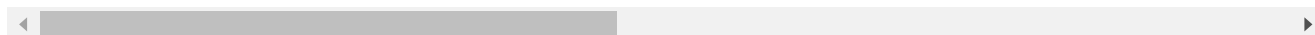
</div>
```

As you can see, we just have a simple link that says “Open Modal” and links to our *openModal* div that is placed right below it. We are doing all of our styling here with classes, so we use the ID just as a hook for opening our modal box, and we will style everything using the *modalDialog* class.

Next, we add a div tag that will hold all of our content in the modal box. Inside of this div we are going to have a link to close the box which we will style with our CSS. We will then put a simple heading with a few paragraphs of text beneath it. Your HTML markup should now look like this:

```
<a href="#openModal">Open Modal</a>

<div id="openModal" class="modalDialog">
  <div>
    <a href="#close" title="Close" class="close">X</a>
    <h2>Modal Box</h2>
    <p>This is a sample modal box that can be created using the r
    <p>You could do a lot of things here like have a pop-up ad th
  </div>
</div>
```



STARTING STYLING

Right now we just have a link with a div showing beneath it. We will begin styling our box and making it actually functional. Let's first create our *modalDialog* classes and start moving forward.

```
.modalDialog {  
    position: fixed;  
    font-family: Arial, Helvetica, sans-serif;  
    top: 0;  
    right: 0;  
    bottom: 0;  
    left: 0;  
    background: rgba(0,0,0,0.8);  
    z-index: 99999;  
    opacity:0;  
    -webkit-transition: opacity 400ms ease-in;  
    -moz-transition: opacity 400ms ease-in;  
    transition: opacity 400ms ease-in;  
    pointer-events: none;  
}
```

The code here is pretty simple. We style our dialog box by giving it a fixed position, meaning it will move down the page, when open, if you scroll. We also set our top, right, bottom, and left edges to 0 so that our dark background will span across the entire monitor.

Since we are going to want the background around the modal box to go dark when it's open, we set the background to black, and change the opacity slightly. We also make sure that our modal box is sitting on top of everything by setting our z-index property.

Lastly, we set a nice transition for our modal box to show up on the screen, and hide the box when it's not clicked on by setting the display to none.

You may not be completely familiar with what the pointer-events property, but it allows you to control when you do and don't want elements to be clickable. We set it for our *modalDialog* class because we don't want the link to be clickable until the ":target" pseudo class is fired.

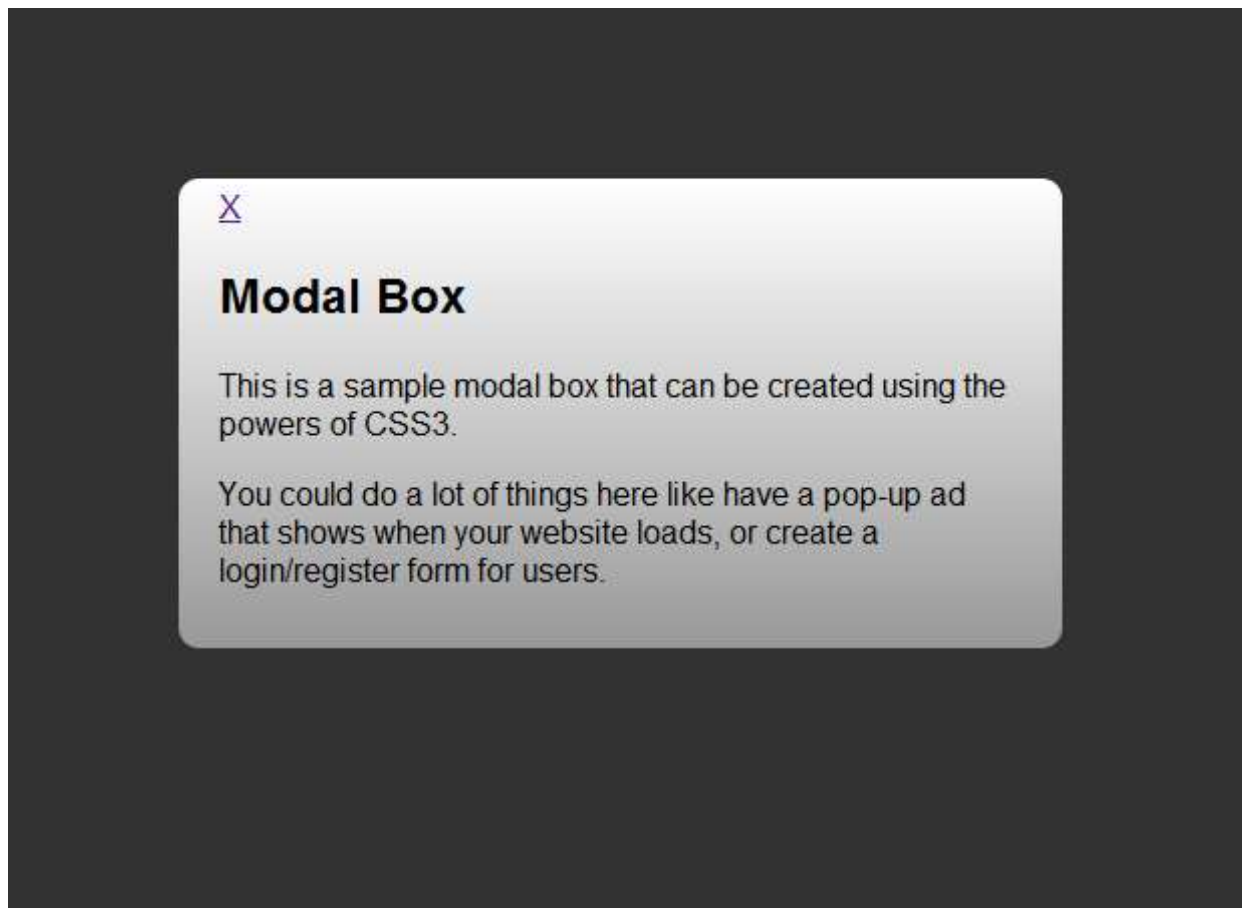
FUNCTIONALITY AND LOOKS

Now let's add our *:target* pseudo class as well as the styling for our modal box.

```
.modalDialog:target {  
    opacity:1;  
    pointer-events: auto;  
}  
  
.modalDialog > div {  
    width: 400px;  
    position: relative;  
    margin: 10% auto;  
    padding: 5px 20px 13px 20px;  
    border-radius: 10px;  
    background: #fff;  
    background: -moz-linear-gradient(#fff, #999);  
    background: -webkit-linear-gradient(#fff, #999);  
    background: -o-linear-gradient(#fff, #999);  
}
```

With our *target* pseudo class, we set our display to block, so that when the link is clicked our modal box will display. We also use our pointer-events property so that when the link is hovered over it's active.

We then style our div tag by setting the width, position, and using our margins to bump the modal box down from the top, and centering it on our page. We then add a little style by creating some padding, setting a nice border radius, and using a gradient of white to dark gray for our background.



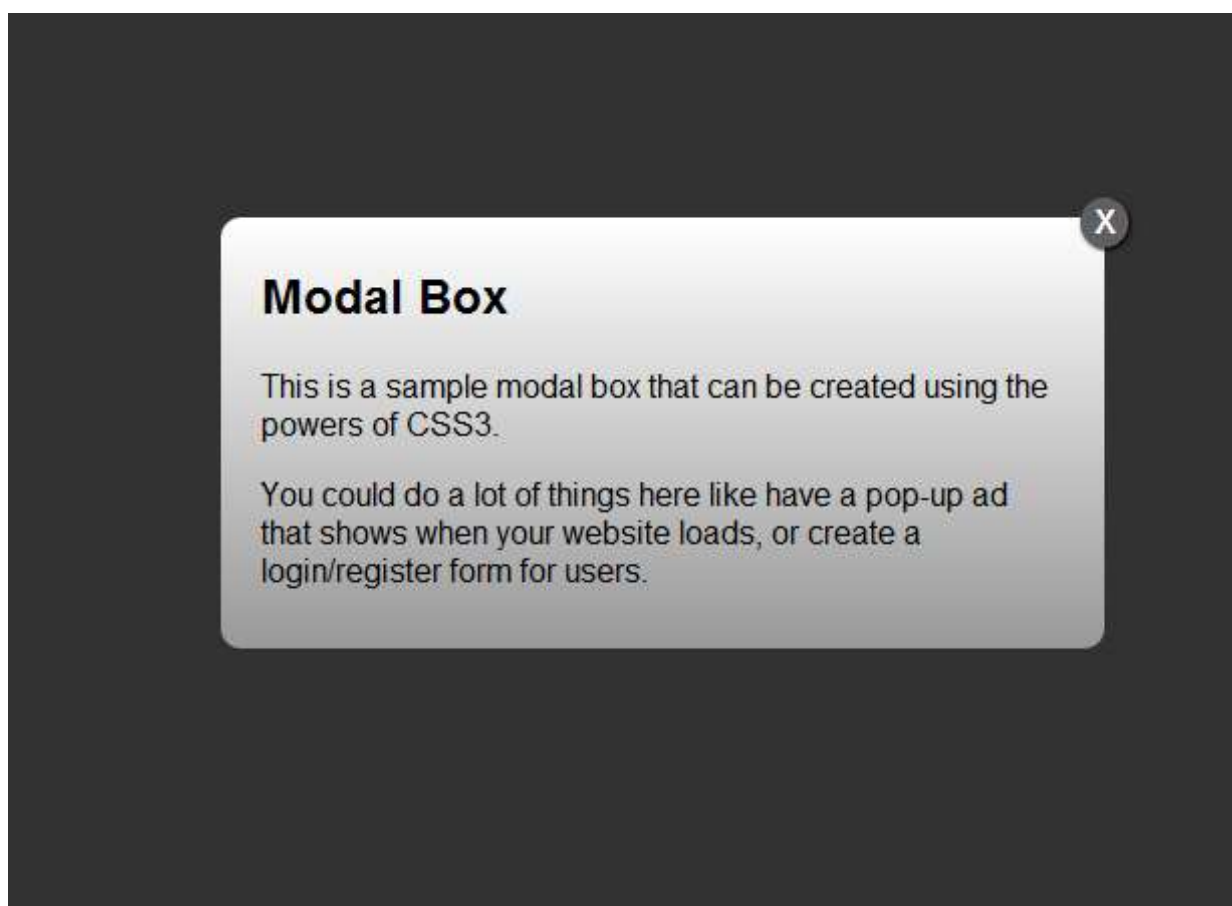
CLOSING IT UP

Now that we have styled the modal box and made it functional, the last thing we need to do is get our close button looking good. Using CSS3 and HTML5 can create buttons that look like they are images, but really aren't. Here's the CSS we use for this:

```
.close {  
    background: #606061;  
    color: #FFFFFF;  
    line-height: 25px;  
    position: absolute;  
    right: -12px;  
    text-align: center;  
    top: -10px;  
    width: 24px;  
    text-decoration: none;  
    font-weight: bold;  
    -webkit-border-radius: 12px;  
    -moz-border-radius: 12px;
```

```
border-radius: 12px;  
-moz-box-shadow: 1px 1px 3px #000;  
-webkit-box-shadow: 1px 1px 3px #000;  
box-shadow: 1px 1px 3px #000;  
}  
  
.close:hover { background: #00d9ff; }
```

For our close button we set the background button and position the text using our text-align and line-height. Then we position the button using position absolute, and set our top and right properties. How do we make it a circle and add some depth? We set the border radius to 12 and create a slight drop shadow. Then to add a little user response we change the background to a light blue when you hover over it (other ideas for hovering include adding a linear transition to fade in, changing font text, or expanding the box shadow slightly).



WHY OUR MODAL BOX IS BETTER

As you've probably noticed, the big selling point with this technique is creating a modal box in HTML5 and CSS3. Why is that such a big deal though? Modal boxes in

JavaScript are something a beginner could create, there are hundreds of examples and downloads ready to be used. So why do we want to scrap JavaScript in favor of HTML5 and CSS3?

It would be naïve of me to say that one selling point is making sure people with JavaScript can use them; statistics show that only 2% of people worldwide browse without JavaScript, so if that is not an issue, what is? Well we took advantage of CSS3 transitions when creating our modal box. The complete code was:

```
-webkit-transition: opacity 400ms ease-in;  
-moz-transition: opacity 400ms ease-in;  
transition: opacity 400ms ease-in;
```

We only used three lines of code to create our animation. If you compare that to any JavaScript animation library you will be shocked to see how much our code is minimized. This leads into another reason, that is, we have cleaner code. We know what div that the animation is applied to, and that it's only 3 lines long. This makes it easy for us to modify it or change the div as we see fit, and not have to worry about changing a div in JavaScript as well as the CSS and HTML.

Lastly, HTML5 and CSS3 are the future. Everybody is working to implement them into their designs and projects, and using them helps perpetuate its adoption and ensures you don't get left behind. You get cleaner code, you don't have to worry about JavaScript libraries, and you have a whole slew of web designers and developers that are ready to help with any issues you might have because they are excited to learn more about the language as well. HTML5 and CSS3 aren't going anywhere, so there's no reason not to use them.

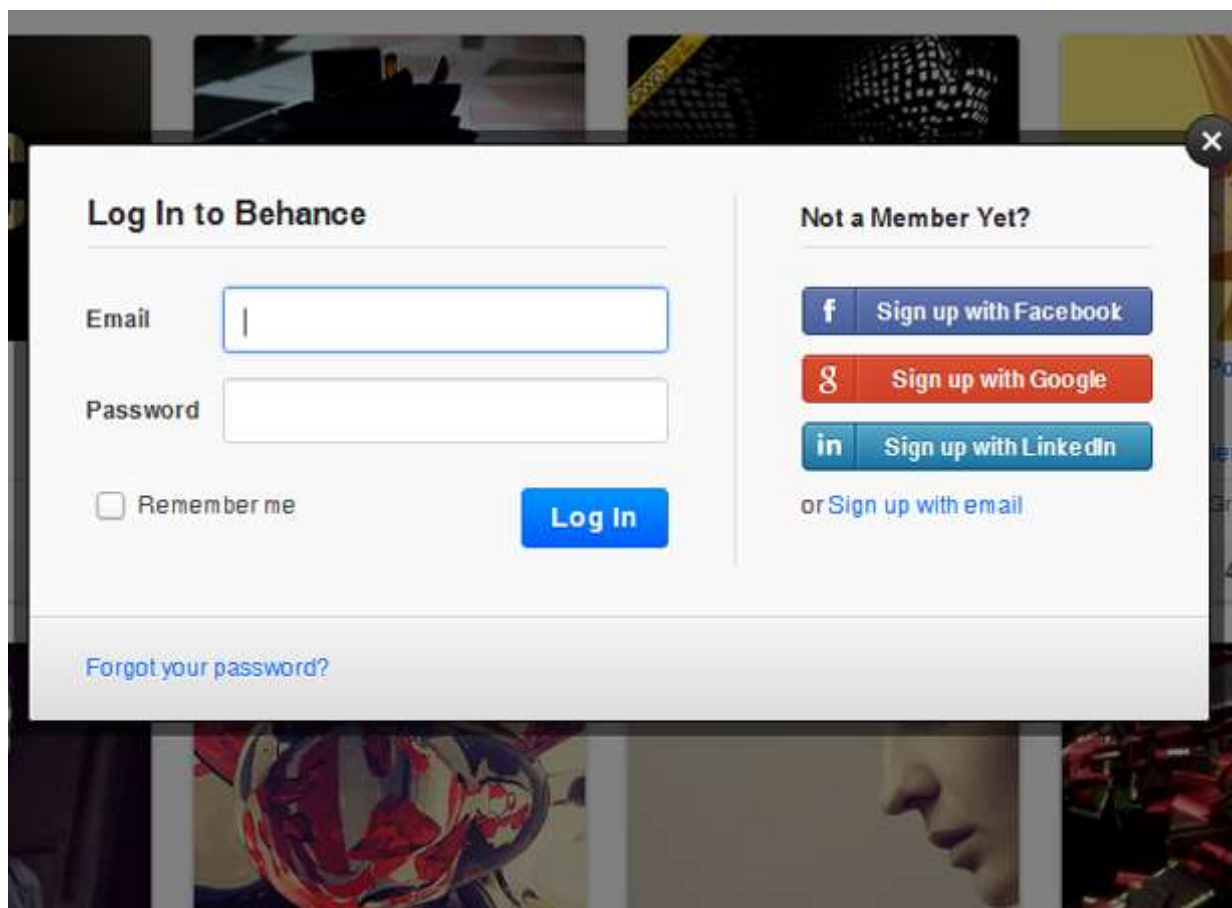
WHEN TO USE MODAL BOXES

So now that we've covered how to make the modal boxes and why you should make them with HTML5 and CSS3, what about when you should use modal boxes?

LOGIN/REGISTER FORM

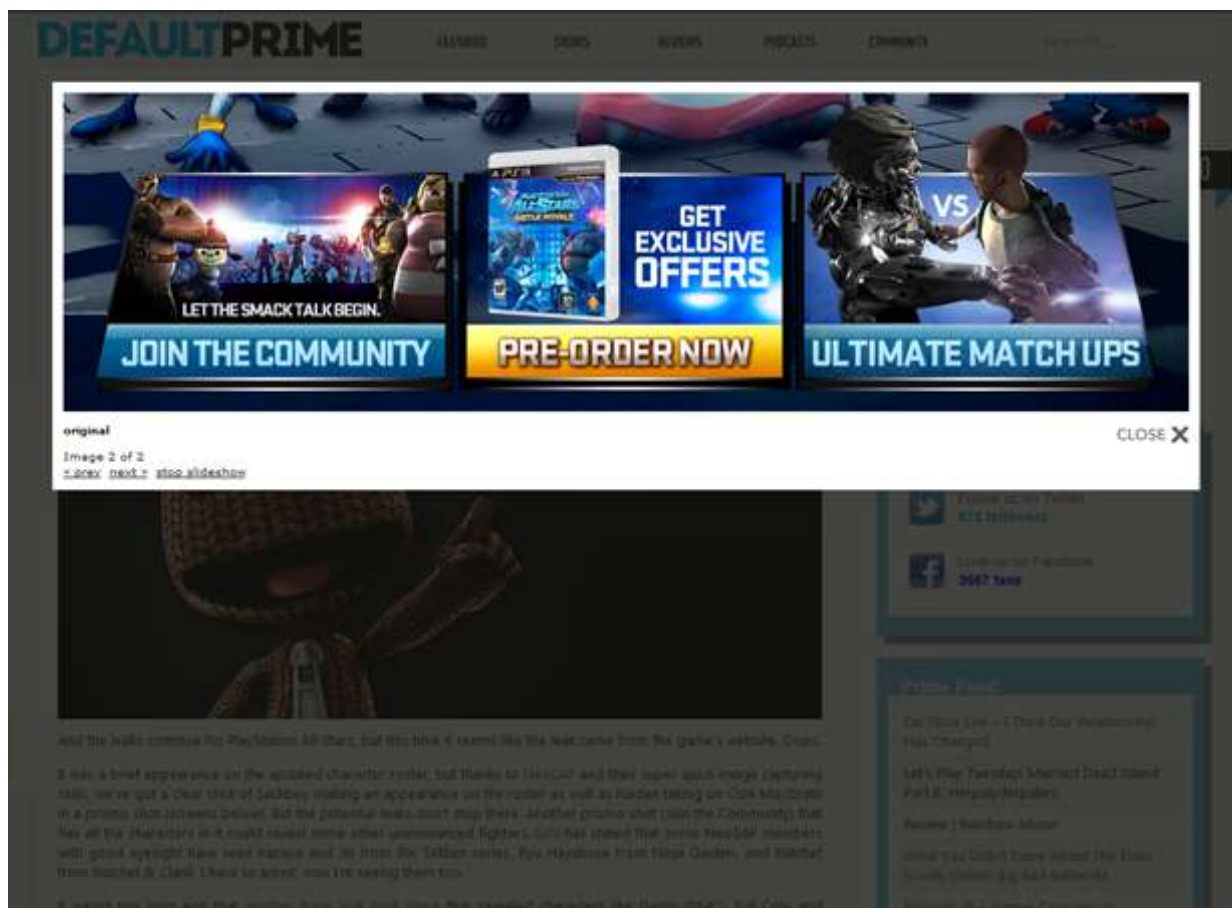
I think that modal boxes are particularly useful when used for login and registration forms. It creates such a streamlined experience for the user, and can really make an

impression on your visitors.



DISPLAY IMAGES/VIDEOS

Another excellent use of modal boxes is to display images or videos (commonly known as lightboxes). It allows for the user to view the content of your website without leaving the webpage, thus keeping them further invested in your web experience.



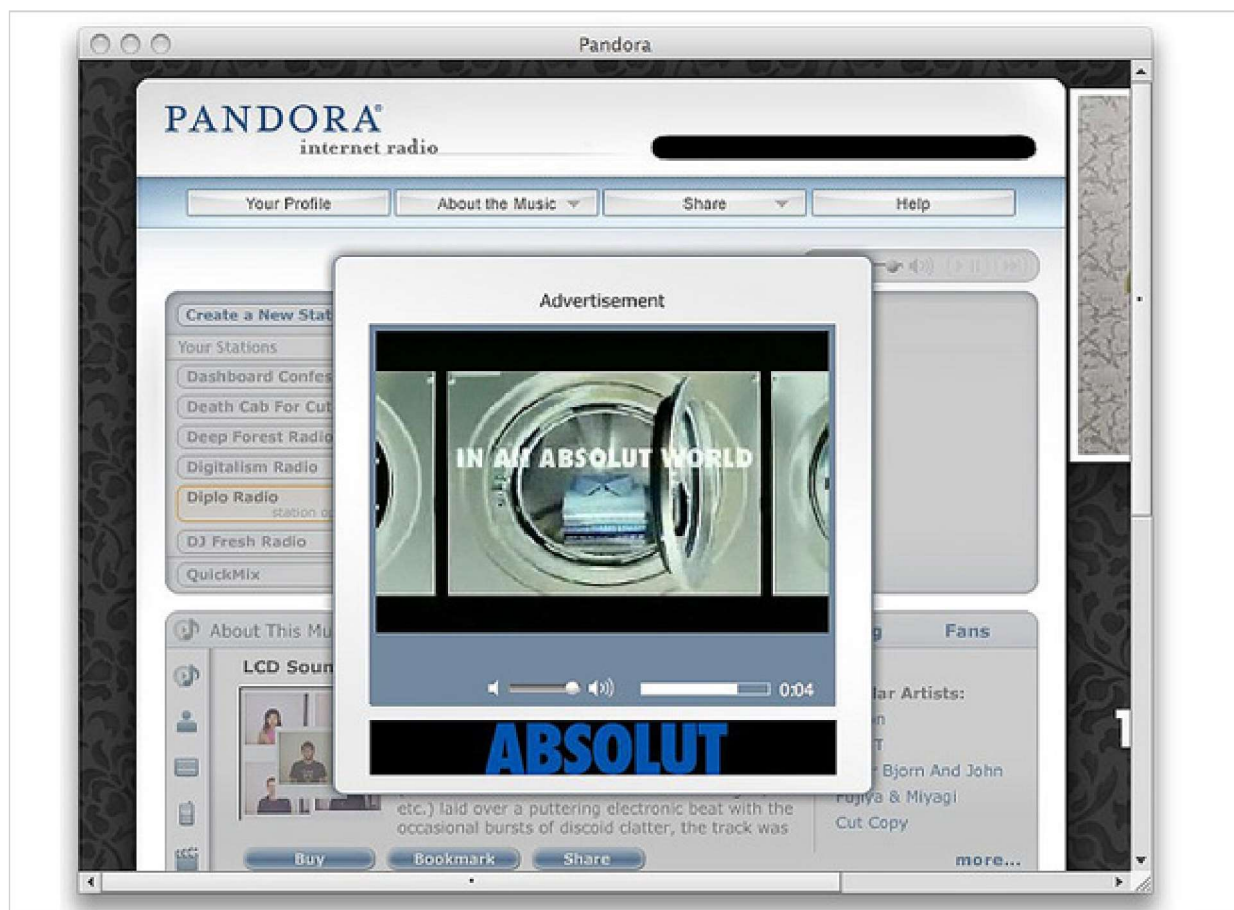
USER FEEDBACK

When a user is interacting on your webpage and you want to communicate or alert them about something, what better way to do it than a modal box? Facebook takes great advantage of this for when you want to change your language on their website, or for short blurbs like the “Why do I need to provide my birthday” link when you’re signing up. If it doesn’t have enough text or need to be a whole page, why not use a modal box?



ADVERTISEMENTS

A great example of advertising using modal boxes is the music radio service Pandora. If you are not a member of their premium service, every once in a while you will get an advertisement on your screen. Sometimes these are audio advertisements that look like you're just listening to a song, and sometimes it's a popup box for a video. This is a great tactic for people looking to inject advertising into their webpages without always having them on the page.



CONCLUSION

There you have it, you can now create a simple HTML5 and CSS3 modal box, and you can use it for things like user login/registration forms, advertisements, and much more. You've also learned why we should use HTML5 and CSS3 instead of JavaScript, and seen some examples of how sites use modal boxes and how you can implement them into your own designs.

Are you excited by the possibilities of HTML5 & CSS3? Will JavaScript eventually go the way of Flash? Lend us your views in the comments below.

Thumbnail image [via Shutterstock](#).



BY KEENAN PAYNE

Keenan Payne is a full time student and a freelance web designer in his spare time. Besides writing and web design, he loves to skateboard and snowboard, and has been doing both for 11 and 6 years respectively, although some would say it's getting progressively harder for him to step away from his work and wander outside. [More articles](#) by Keenan Payne

Comments for this thread are now closed.



20 Comments Webdesigner Depot

Login ▾

Recommend 22

Share

Sort by Oldest ▾

**Controlz 05** • 4 years ago

How to you properly code this for a video? For example I did [Open Modal](#) but the video display the entire width of the page complete with toolbar, etc. The user (after watching the video) then has to use the back button to get back to the page where the link to the video appeared.

2 ^ | v • Share ›

**Walter** → Controlz 05 • 4 years ago

That really depends on how you're embedding your video in the page. This code just hides and shows a <div>, it won't affect the contents.

8 ^ | v • Share ›

**Guest** • 4 years ago

This is a very useful guide, thanks for sharing!

5 ^ | v • Share ›

**Thor** • 4 years ago

Is it just me - or isn't it animated at all?

5 ^ | v • Share ›

**aleguarita** → Thor • 4 years ago

Yeah... it didn't :(

1 ^ | v • Share ›

**Walter** → Thor • 4 years ago

Thanks for pointing that out, we've adjusted the code above and the demo code to correct that.

8 ^ | v • Share ›

**iCodeGuy** • 4 years ago

Great little tutorial. I have to admit to being lazy however and using the modal js and css from Bootstrap every time I need to implement a Modal window.

5 ^ | v • Share ›

**Jenn** • 4 years ago

What's the browser compatibility of this modal window?

^ | v • Share ›

**Walter** → Jenn • 4 years ago

It will essentially work in anything, including all mobile browsers, except IE 7 and IE 8. (It's fine in IE9.)

19 ^ | v • Share ›



Pete Nawara • 4 years ago

I don't believe that 'display: none' to 'display: block' will animate in chrome or ff, unfortunately. Currently toying with finding a way around this.

^ | v • Share ›



Pete Nawara • 4 years ago

So after toying with this for a minute (which is awesome by the way) i came up with a dirty method for allowing for transitions (tested in chrome). I am going to continue to work on making it nice nice. <http://codepen.io/petebot/pen/...>

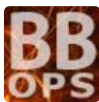
2 ^ | v • Share ›



Walter ➔ Pete Nawara • 4 years ago

Good job, looks great!

^ | v • Share ›



Black Book Operations • 4 years ago

Thanks for this little guide here, useful indeed!

^ | v • Share ›



Adam Taylor • 4 years ago

You can't transition 'display' properties.

I've done the same trick by using a negative z-index instead, so the hidden dialog is 'below' the level of the page, and swapping it out for z-index:100 or whatever on :target.

^ | v • Share ›



Walter ➔ Adam Taylor • 4 years ago

That's right, the reason you can't animate display properties is that there's no degree of display:none, you can either see it or you can't.

If you need to animate something like this you'll need to use opacity.

1 ^ | v • Share ›



Morné • 4 years ago

It's a good tutorial, the only problem with this is compatibility with older browser like IE7 and IE8. It's great though that you can do this with just html and css.

HTML5 and CSS3 will in no way kill off javascript :)

Keep up the good work

3 ^ | v • Share ›



Walter ➔ Morné • 4 years ago

That's very true, if you need to support browsers that don't work with CSS3 (most will work with HTML5) then you might want to consider a javascript backup plan.

**Kishore** • 4 years ago

Thanks for your tutorial, It is helps me to resolve my issue.

1 ^ | v • Share ›

**pramod bisht** • 4 years ago

nice collection



13 ^ | v • Share ›

**cvolz** • 4 years ago

this is great. I am new to this and I was wondering if there is a way I can have this pop up on my page without clicking the tab. Thanks for sharing!! and thanks for the help

36 ^ | v • Share ›

ALSO ON WEBDESIGNER DEPOT

**Popular design news of the week:
November 28, 2016 - December 4, 2016**

4 comments • a month ago•

**Albert Greens** — This is interesting! I will figure it out something before I use all the information here.**3 basic web design errors**

5 comments • 3 days ago•

**CyncialOne** — Well written article that seems to be missing a big point. QA lot of**Why you should be using pattern libraries**

9 comments • a month ago•

**gojquery.com** — I was looking for more pattern library examples.**Comics of the week #372**

1 comment • 10 days ago•

**Tag Team Design** — Great stuff - Always enjoy your posts!

these flaws aren't in websites by mistakes

 [Subscribe](#)  [Add Disqus to your site](#) [Add Disqus](#) [Add](#)  [Privacy](#)