ZURB Playground

- **ZURB University**

-

  - Courses
  - Lessons
  - Library

    - **« Back**

      - Pattern Tap
      - Word
      - Responsive
      - Playground
      - Quips
      - Friday 15
      - Triggers
      - Building Blocks
  - Community
  - Job Board

Library

Playground

- Triggers
- Quips
- Responsive
- Pattern Tap
- Word
- Playground
- Friday15
- Building Blocks

- Home
- About
- Graduates

Image Uploads with 100% Less Suck. Guaranteed.

Using a little bit of jQuery we can upload images without the suck. *Note: This demo will rock your socks in all browsers.*
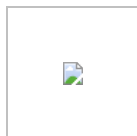
Old and Busted

Upload a picture of yourself

[Choose File] proxy.jpg   Upload

Select a file and click upload. Simple right? Except once I select my image I can no longer see what was selected. The name of the file is at the end of the input, and if the input is short, or the file path is deep, I'm not going to see anything useful. Assuming that I took the time to give my file a descriptive name, which I probably didn't because hey, it's an image, it should be self descriptive.

New Hotness

Upload a picture of yourself

[Choose File] No file chosen   Save

Now try this variation. We ditch the upload button in favor of a save button and fire the AJAX upload event as soon as a file is selected. The image is processed server side and a thumbnail is loaded onto the existing page. Doesn't that feel so much better?

We now have a visual representation (imagine that) of the image we selected. This is particularly useful in larger forms when many fields will be submitted with a single action. It allows the user to review the form before pressing save and see what image (or images) they selected.

The Code

You're going to need jQuery and the AJAX Upload jQuery plugin. Link them up, make sure jQuery is loaded first.

```
<script src="/js/jquery.min.js" type="text/javascript"></script>
<script src="/js/ajaxupload.js" type="text/javascript"></script>
```

Here is the JavaScript we're going to add in its entirety. You can also see it if you view the source of this page.

```
$(document).ready(function(){

  var thumb = $('img#thumb');

  new AjaxUpload('imageUpload', {
    action: $('form#newHotnessForm').attr('action'),
    name: 'image',
    onSubmit: function(file, extension) {
      $('div.preview').addClass('loading');
    },
    onComplete: function(file, response) {
      thumb.load(function(){
        $('div.preview').removeClass('loading');
        thumb.unbind();
      });
      thumb.attr('src', response);
    }
  });

});
```

Now let's break it down.

First we attach the the AjaxUpload behavior to our `file` form element.

```
new AjaxUpload('imageUpload', {
```

Next we specify where we want to post the AJAX upload to. We like to keep all our urls in our html document so we pass this url using the `action` attribute of our `form` element.

```
action: $('form#newHotnessForm').attr('action'),
```

Set the name of the `file` form element that will be posted to your server.

```
name: 'image',
```

Add a `class` to your preview div to indicate that the image is uploading. In our case we are applying a background image to the preview div. We also need to `display: none;` the `img` tag in the preview div so that the loading background image is visible, also for a more subtle reason explained below.

```
onSubmit: function(file, extension) {
  $('div.preview').addClass('loading');
},
```

When the image has been uploaded we need to do two things. First we need to set the `src` attribute of our preview `img` tag to the new thumb. Secondly we need to remove the loading class. If we simply execute those things in that order then we will see an annoying flicker of the old image when the loading class has been removed but the new image has not yet loaded.

We avoid this problem by waiting to remove the loading class until after the preview image's `load` event fires. We also `unbind` our listener after it had fired since we only wanted to capture this event once.

```
onComplete: function(file, response) {
  thumb.load(function(){
    $('div.preview').removeClass('loading');
    thumb.unbind();
  });
```

Lastly we set the source of preview image to the thumbnail our server just created. In this example the response from the AJAX call is just the thumbnails url as text. You could also return it in whatever fancy format you like.

```
  thumb.attr('src', response);
}
```

Score an awesome product engineering or design job

- Designer on the Senior Path ZURB in Campbell, CA
- Application Developer - Android OS Booking.com in Amsterdam
- Application Developer - iPhone/iPad OS Booking.com in Amsterdam, the Netherlands
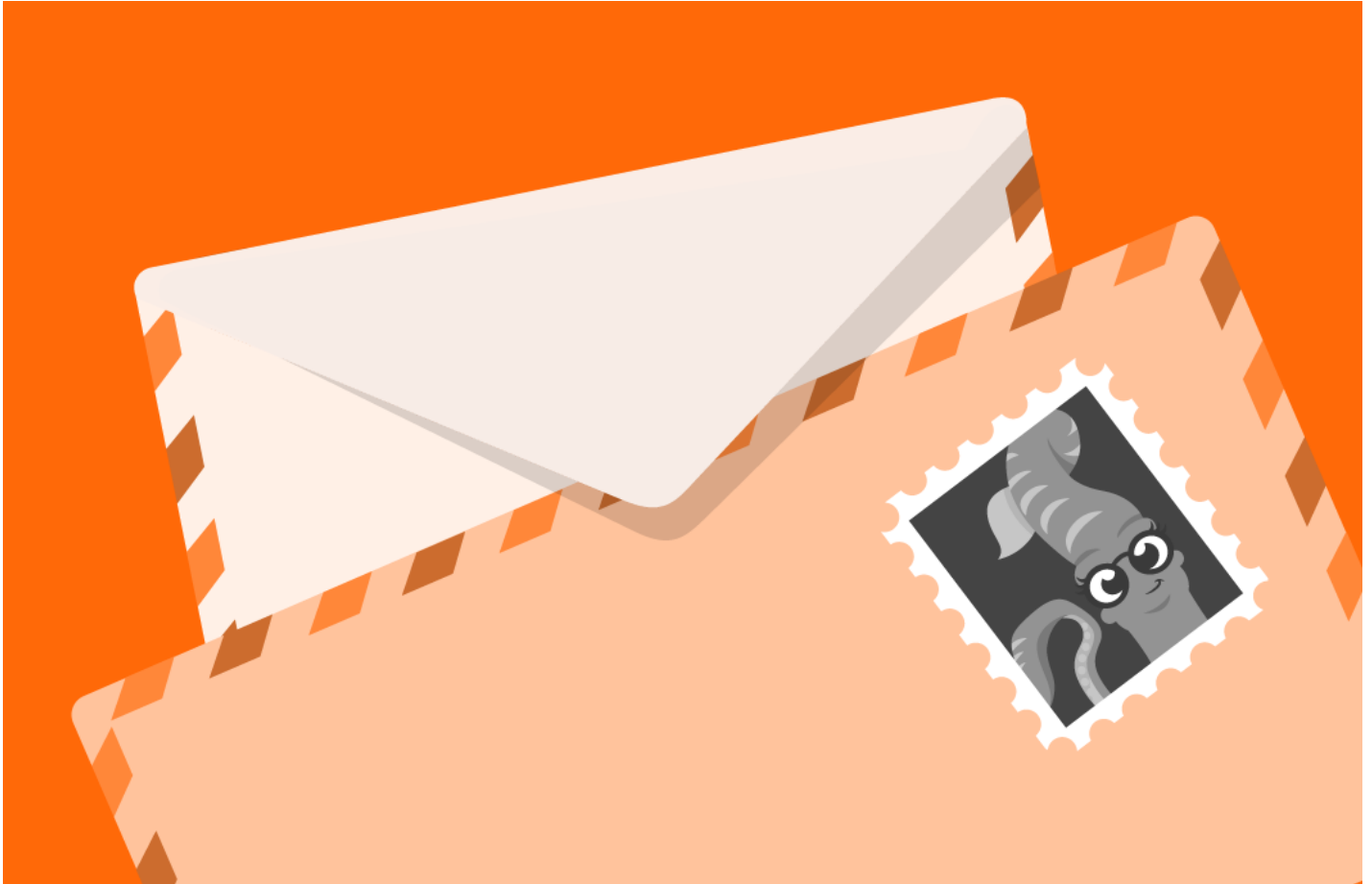
via Job Board from ZURB

GitHub

Follow us on GitHub to be informed about the latest code updates to our projects as they happen.

Product Design Training from the Experts at ZURB

Introduction to Responsive Web Design

Introduction to Responsive Email

Introduction to Foundation
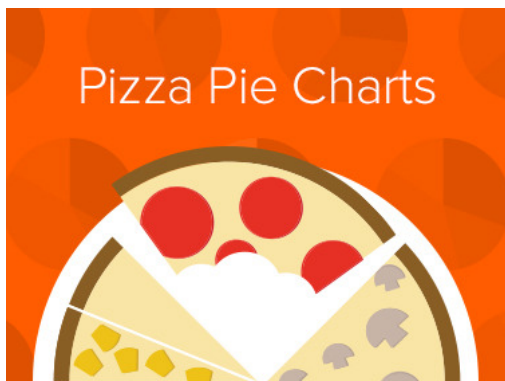
Master Design Feedback

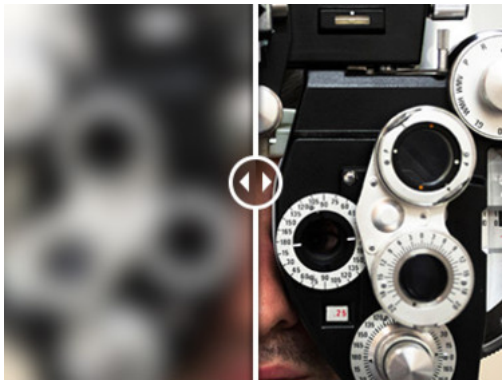## Similar Playground experiments

Pizza Amore!

Pizza Pie Charts

Wrangle jQuery Plugin

[jQuery TwentyTwenty Plugin](#)

- [Home](#)
- [About](#)
- [Graduates](#)

[Playground](#)

Hundreds of thousands of people come through ZURB University, where they get the training they need to be awesome Product Designers. ZURB's courses, workshops, library of reference materials, and design community get you the Product Design skills you need to design better products faster.

Want more?

- [Courses](#)
- [Lessons](#)
- [Custom Training](#)
- [Pattern Tap](#)

Talk to us

Got an interesting experiment to share? Email us at university@zurb.com

Stay in touch

Get monthly updates from ZURB in your inbox. Read our [latest news & buzz.](#)

[Get ZURBnews!](#)
[Studios Helping more than 200 startups succeed since 1998.](#)
[Foundation The most advanced front-end framework in the world.](#)
[Design Apps Prototype, iterate and collect feedback on your products.](#)
[University Ideas, thoughts and design resources shared with you.](#)

-
-
-

- [About](#)
- [Blog](#)
- [News & Events](#)
- [Contact](#)
- [Sitemap](#)

© 1998–2014 ZURB, Inc. All rights reserved.

Reveal Modal Goodness

This is a default modal in all its glory, but any of the styles here can easily be changed in the CSS.

This is just a simple modal with the default styles, but any type of content can live in here. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi quis sem vel enim eleifend tristique. Etiam tincidunt faucibus pharetra.

×

Playground

- [Home](#)
- [About](#)
- [Graduates](#)

More ZURB Goodness

- [Foundation Training](#)
- [Product Design Courses](#)
- [Design Apps](#)

- ZURB
- [About](#)
- [Blog](#)
- [Contact](#)

- [Studios Helping startups win since '98.](#)
- [Foundation World's most advanced responsive framework.](#)
- [Design Apps Tools to rapidly prototype and iterate.](#)
- [University Online training for smarter product design.](#)