The Mozmonkey Blog

The life and thoughts of Jeremy Gillick

Good UX for Placeholders as Field Labels (i.e. Float Labels)

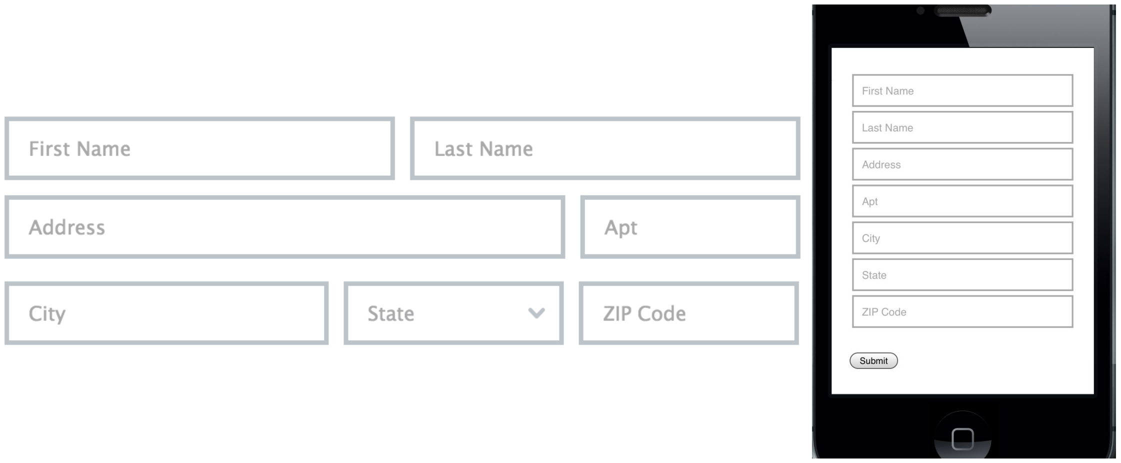
December 4, 2013 Code, Technical, Usability ux, webdev

Checkout my follow-up CSS only solution

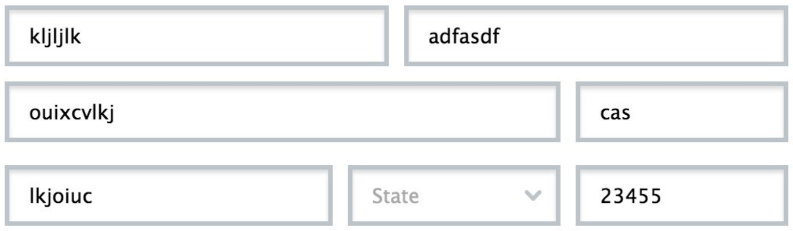
UPDATE: My floating labels were featured on CSS Tricks!

I know that it is well understood that using placeholder labels are bad practice. However, they continue to be used because they make forms appear tight, and responsive for mobile devices. Unfortunately, in practice they don't provide a very good user experience.

For those who are not familiar, this is what I'm talking about:



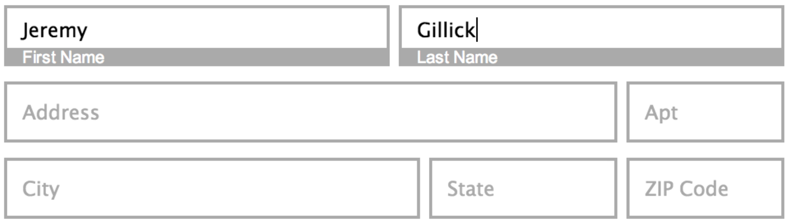
On the surface, the form looks good. It's especially sexy on mobile where screen real estate is limited. The problem comes when the user starts using the form:



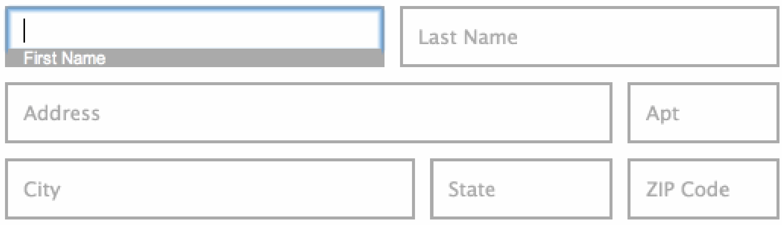
Now can you tell which field is for what value? (besides the state drop-down) If a user accidentally puts their city in the address field and the address in the city field, they will not know it. This is commonly the case when people are typing on autopilot. If you have extensive validation, you could catch this error, but then you punish the user for your poor design.

The Solution

I like the idea of using placeholders as labels, so to solve the inherent UX problems my solution was to “slide” the placeholder to the bottom of the field when the user focuses inside of it (this is also known as Float Labels).



You maintain the sharp appearance of your form while still supporting your user through it. See it in action and try the demo!



Click for the live demo

The Nitty Gritty

HTML

Starting with some basic HTML.

```
<span class="field">
   <label for="fname">First Name</label>
   <input type="text" id="fname" name="fname" placeholder="First Name" />
</span>
```

The `field` classed element will be the wrapper around the input and label elements. We want to remove the border from the input element and apply it to the field wrapper element, making it appear as the text field (this is important for how we position the label later):

CSS

```
.field {
  display: block;
  position: relative;
  height: 36px;
  border: 2px solid #aaa;
}
.field input {
   border: none;
   width: 100%;
   height: 36px;
   box-sizing: border-box;
}
```

Position the label to the bottom of the wrapper element and hide it by setting height to zero:

```
.field label {
  height: 0;
  overflow: hidden;
  position: absolute;
  right: 0; bottom: 0; left: 0;
  font-size: 11px;
  color: #fff;
  background: #aaa;
}
```

Now define the `.show-label` class to show the label and hide the placeholder text. This class will be added and remove by JavaScript later.

```
.field.show-label {
  height: 38px;
  border-bottom-width: 0;
}
.field.show-label input {
  height: 28px
}
.field.show-label label {
  height: auto;
}
```

JavaScript

Last, but not least, the JS to toggle the `.show-label` class.

```
$('.field').each(function(){
  var parent = $(this),
      field = parent.find('input, select');

  // Focus: Show label
  field.focus(function(){
    parent.addClass('show-label');
  });

  // Blur: Hide label if no value was entered (go back to placeholder)
  field.blur(function(){
    if (field.val() === '') {
      parent.removeClass('show-label');
    }
  });
});
```

CSS Transitions

So far so good, but what we have so far is a bit jarring out of the box. Let's add some CSS transitions to smooth things out:

```
.field {
  transition-property: height, border-width;
  transition-duration: 0.3s;
  transition-timing-function: ease-in;
}
.field input,
.field select {
  transition: height 0.3s ease-in;
}
.field label {
  transition: max-height 0.3s ease-in;
}
.field input[placeholder]::-webkit-input-placeholder {
  transition: opacity 0.3s ease-in;
}
```

That's it! Well almost…

Safari

By default, Safari keeps the placeholder text visible until the user start typing. Our new program shows the label as soon as the field is focused, so we'll want to hide the placeholder. Luckily we can do that with this line of CSS:

```
.field.show-label input[placeholder]::-webkit-input-placeholder {
  opacity: 0;
}
```

We're hiding with it by opacity so we can fade it out later with CSS3 transitions.

Internet Explorer

You knew you couldn't get through this without IE complaining. Internet Explorer 9 and below does not support the placeholder attribute. You can easily get around this by using the jQuery placeholder plugin.

```
<script type="text/javascript" src="jquery.placeholder.js"></script>
<script type="text/javascript">
$(document).ready(function(){
  $('input, textarea').placeholder();
});
</script>
```

Auto-fill

To support browser auto-fill and other programs that might update your field values, you'll want to add a change event handler t your JavaScript `each()` block, like this:

```
// ... inside the $(field).each() block...

// Handles change without focus/blur action (i.e. form auto-fill)
field.change(function(){
  if (field.val() !== '') {
    parent.addClass('show-label');
  } else {
    parent.removeClass('show-label');
  }
});
```

Safari Note: Safari does not call the onchange event when it auto-fills fields. The common solution to this is to run a setInterval timer to check if any of the field values have been changed.

Full Demo

To see it all together, view the full demo.

What's next

Checkout my follow-up post showing a CSS only solution.

← Booze Bookshelf with LEDs and built-in dance party

Serializing Embedded Relationships with Ember Data 1.0.0 beta →

Comments

Floated Labels Still Suck | Web Axe

October 30, 2014 at 4:06 PM

[…] attempt is to, upon focus of the input, reduce the text size of the floated label and move it to the bottom of the input field (rather than it disappearing entirely). This is not a feasible solution either as this […]

Search …

Meta

- Log in
- Entries RSS
- Comments RSS
- WordPress.org

Facebook Activity

Recent Posts

- Ember: Getting the index in #each loops
- CSS Only Placeholders Field Labels (i.e. Float Labels)
- Loading JSON with embedded records into Ember Data 1.0.0 beta
- Serializing Embedded Relationships with Ember Data 1.0.0 beta
- Good UX for Placeholders as Field Labels (i.e. Float Labels)

Archives

- March 2014
- February 2014
- December 2013
- October 2013
- September 2013
- February 2011
- November 2009
- January 2009
- August 2008
- July 2008
- April 2008
- March 2008
- February 2008
- January 2008
- December 2007
- July 2007
- May 2007
- April 2007
- March 2007
- February 2007
- January 2007
- November 2006
- October 2006

Categories

- Accessibility
- Arduino
- Code
- Conferences
- Design
- Electronics
- Events
- Extensions
- Funny
- Insipiring
- Life
- Miscellaneous
- Software
- Technical
- Travel
- Twickr
- Usability
- Websites

May 2006

April 2006

March 2006