

****Published by: [SaaS Hero](#)****

JavaScript

<script>

```
// To get the following variables to work, you may need to add classes to
Unbounce elements.
// If you have a form headline that needs to change, add the class
'form-headline' to the text box
// If you have content that needs to hide/show based on step. You'll need to
add the class 'step-one' or 'step-two' etc.. to each steps content.
// If you have a form box add the class 'form-box' to it

// Set starting CTA copy
var startingCTACopy = 'name-of-your-CTA-to-move-users-to-step-2'

var apostrophe = '\u0027'

// If there's a back button, set to back button's text eg. var backButton =
'< Back', otherwise set it to false
var backButtonEnabled = 'Back'

// Set how many fields show on each step. Examples: [1,2,5] or [2,5,2,2,2]
var stepArray = [1,1,3]

// Set form headlines, if no change leave at ''
var formHeadlineTwo = 'What\u0027s your current monthly ad spend?*'
var formHeadlineThree = 'Contact details'
var formHeadlineFour = ''
var formHeadlineFive = ''

// Set CTA copy Changes, if no change leave at ''
var ctaCopyTwo = 'Next'
var ctaCopyThree = 'Audit my Account'
var ctaCopyFour = ''

// Set Box Height Changes, if no change leave at ''
var boxHieghtTwo = ''
var boxHieghtThree = ''
var boxHieghtFour = ''
var boxHieghtFive = ''

// Set Box Top Changes, if no change leave at ''
var boxTopTwo = ''
var boxTopThree = ''
```

```

var boxTopFour = ''
var boxTopFive = ''

// Set main CTA Top Changes, if no change leave at ''
var ctaTopTwo = ''
var ctaTopThree = ''
var ctaTopFour = ''
var ctaTopFive = ''

// Mobile Variables

// Set mobile Box Height Changes, if no change leave at ''
var mobileBoxHieghtTwo = ''
var mobileBoxHieghtThree = ''
var mobileBoxHieghtFour = ''
var mobileBoxHieghtFive = ''

// Set mobile main CTA Top Changes, if no change leave at ''
var mobileCtaTopTwo = ''
var mobileCtaTopThree = ''
var mobileCtaTopFour = ''
var mobileCtaTopFive = ''

// Set mobile Box Top Changes, if no change leave at ''
var mobileBoxTopTwo = ''
var mobileBoxTopThree = ''
var mobileBoxTopFour = ''
var mobileBoxTopFive = ''

/*****
    !!!WARNING!!! EDITING BELOW THIS POINT MAY BREAK THE SCRIPT
    *****/

var stepObject = []
stepArray.forEach(element => {
    stepObject.push({fields: element})
});

function UnbounceMultiStep(form, options) {
    // Validate arguments
    if (!form.is('.lp-pom-form form')) {

```

```

        return console.error('jQuery.unbounceMultiStep must be called on an
Unbounce form.');
```

```

    }

    if (typeof options !== 'object') {
        return console.error('No options were passed to
jQuery.unbounceMultiStep');
    }

    this.options = options;

    // Store DOM elements
    this.form = form;
    this.formContainer = this.form.closest('.lp-pom-form');
    this.fields = this.form.find('div.lp-pom-form-field');
    this.fieldsByStep = [];
    this.currentStep = 0;
    this.forwardButton = this.form.find('button[type=submit]').eq(0);

    // Verbiage
    this.text = {};
    this.text.next = this.options.nextButton;
    this.text.back = this.options.back;
    this.text.showSteps = this.options.showSteps;
    this.text.submit = this.forwardButton.text();

    this.init();
}

UnbounceMultiStep.prototype.init = function() {
    this.formContainer.addClass('multistep initial');
    this.form.attr('id', 'fields');

    // Add progress bar
    this.formContainer.prepend('<div id="progress-bar"></div>');
    this.progressBar = jQuery('#progress-bar');

    // Replicate Unbounce's field spacing
    var height = parseInt(this.fields.eq(0).css('height'), 10);
    var top = parseInt(this.fields.eq(1).css('top'), 10);
    this.fields.css('margin-bottom', top - height + 'px');
    this.progressBar.css('margin-bottom', top - height + 'px');

    // Set up fieldset elements for each step

```

```

for (var i = 0; i < this.options.steps.length; i++) {
    console.log('Adding new fieldset.');
```

`this.form.find('.fields').append('<fieldset></fieldset>');`

```

}
this.steps = this.form.find('fieldset');
this.steps.addClass('step');

// Sort fields into new steps
var currentField = 0;
for (currentStep = 0; currentStep < this.options.steps.length;
currentStep++) {
    this.fieldsByStep[currentStep] = [];
    for (i = 0; i < this.options.steps[currentStep].fields; i++) {
        console.log('Field ' + currentField + ' -> step ' + currentStep);
        this.fields.eq(currentField).appendTo(this.steps.eq(currentStep));
        this.fieldsByStep[currentStep].push(this.fields.eq(currentField));
        currentField++;
    }
}

// Add any remaining fields to last step
if (currentField < this.fields.length) {
    currentStep--;
    for (i = currentField; i < this.fields.length; i++) {
        console.log('Field ' + currentField + ' -> step ' + currentStep);
        this.fields.eq(currentField).appendTo(this.steps.last());
        this.fieldsByStep[currentStep].push(this.fields.eq(currentField));
        currentField++;
    }
}

this.progressBarItems = jQuery('#progress-bar .step');
```

```

console.log(this)
if (backButtonEnabled) {
    this.backButton =
this.forwardButton.clone().insertBefore(this.forwardButton);
    this.backButton.addClass('back-button');
    this.backButton.children('span').html(this.text.back);
    this.backButton.bind('click.unbounceMultiStep',
this.backHandler.bind(this));
    this.forwardButton.addClass('next-button');
}

```

```

        this.forwardButton.bind('click.unbounceMultiStep',
this.forwardHandler.bind(this));

        this.fields.find(':input').bind('invalid', function(event) {
            // Prevent browser from trying to focus invalid inputs on non-visible
steps
            if (jQuery(event.currentTarget).closest('fieldset.active').length === 0)
{
                event.preventDefault();
            }
        });

        // Show first step
        this.goToStep(0);
    };

    UnbounceMultiStep.prototype.goToStep = function(newStep) {
        // Make sure we aren't going to a step that doesn't exist
        if (newStep < 0 || newStep >= this.steps.length) {
            return false;
        }

        this.steps
            .eq(this.currentStep)
            .removeClass('active')
            .hide();

        this.steps
            .eq(newStep)
            .addClass('active')
            .show();

        if (newStep != 0) {
            $(".active input").first().focus();
        }

        /*
            *
            *
            *****
            * MAIN STEP FUNCTION
            *****
            *
            *
        */
    };

```

```

        */
// This is where you would change what shows and appears each step
// Please note that this script starts counting at 0
// So step 2 would be equal to 1 here
// Step Two
if (newStep == 0) {
    $('#lp-pom-form-150').show()
    $('.step-one').show();
    $('.step-two').hide();
    $('.form_mainbutton').css('left', '208px');
    if (jQuery(window).width() <= 600) {
        $('.form_mainbutton').css('left', '88px');
    }
}
if (newStep == 1) {
    $('.step-one, .step-three').hide();
    $('.step-two').show();
    $('.form_mainbutton').css('left', '275px');
    $('.form_mainbutton').width(90);

    if (formHeadlineTwo !== '') {
        $('.form-headline').children().text(formHeadlineTwo);
    }
    if (boxHieghtTwo !== '') {
        $('.form-box').css('height', boxHieghtTwo);
    }
    if (ctaTopTwo !== '') {
        $('button[type="submit"]').css('top', ctaTopTwo);
    }
    if (boxTopTwo !== '') {
        $('.form-box').css('top', boxTopTwo);
    }
    // Mobile code
    if (jQuery(window).width() <= 600) {
        $('.form_mainbutton').css('left', '145px');
        if (mobileBoxHieghtTwo !== '') {
            $('.form-box').css('height', mobileBoxHieghtTwo);
        }
        if (mobileCtaTopTwo !== '') {
            $('button[type="submit"]').css('top', mobileCtaTopTwo);
        }
        if (mobileBoxTopTwo !== '') {
            $('.form-box').css('top', mobileBoxTopTwo);
        }
    }
}

```

```

    }
  }
  if (typeof ga === 'function') {
    ga('send', 'event', 'CTA click', 'Step 1 completed');
  }
  if (typeof dataLayer !== 'undefined') {
    dataLayer.push({'event': 'Multi-Step', 'eventAction': 'Step ' +
newStep + ' completed'});
  }
}
// Step Three
else if (newStep == 2) {
  $('.step-two, .step-four').hide();
  $('.step-three').show();
  $('.form_mainbutton').width(166);
  $('.form_mainbutton').css('left', '215px');
  // $('.back-button').css('left', '110px!important');
  $(".back-button").attr("style", "left: 110px!important");

  if (formHeadlineThree !== '') {
    $('.form-headline').children().text(formHeadlineThree);
  }
  if (boxHieghtThree !== '') {
    $('.form-box').css('height', boxHieghtThree);
  }
  if (ctaTopThree !== '') {
    $('button[type="submit"]').css('top', ctaTopThree);
  }
  if (boxTopThree !== '') {
    $('.form-box').css('top', boxTopThree);
  }
  // Mobile code
  if (jQuery(window).width() <= 600) {
    $('.form_mainbutton').css('left', '105px');
    $(".back-button").attr("style", "left: 1px !important");

    if (mobileBoxHieghtThree !== '') {
      $('.form-box').css('height', mobileBoxHieghtThree);
    }
    if (mobileCtaTopThree !== '') {
      $('button[type="submit"]').css('top', mobileCtaTopThree);
    }
    if (mobileBoxTopThree !== '') {
      $('.form-box').css('top', mobileBoxTopThree);
    }
  }
}

```

```

    }
  }
  if (typeof ga === 'function') {
    ga('send', 'event', 'CTA click', 'Step 2 completed');
  }
  if (typeof dataLayer !== 'undefined') {
    dataLayer.push({'event': 'Multi-Step', 'eventAction': 'Step ' +
newStep + ' completed'});
  }
}
// Step Four
else if (newStep == 3) {
  $('.step-three, .step-five').hide();
  $('.step-four').show();
  if (formHeadlineFour !== '') {
    $('.form-headline').children().text(formHeadlineFour);
  }
  if (boxHieghtFour !== '') {
    $('.form-box').css('height', boxHieghtFour);
  }
  if (ctaTopFour !== '') {
    $('.button[type="submit"]').css('top', ctaTopFour);
  }
  if (boxTopFour !== '') {
    $('.form-box').css('top', boxTopFour);
  }
  // Mobile code
  if (jQuery(window).width() <= 600) {
    if (mobileBoxHieghtFour !== '') {
      $('.form-box').css('height', mobileBoxHieghtFour);
    }
    if (mobileCtaTopFour !== '') {
      $('.button[type="submit"]').css('top', mobileCtaTopFour);
    }
    if (mobileBoxTopFour !== '') {
      $('.form-box').css('top', mobileBoxTopFour);
    }
  }
  if (typeof ga === 'function') {
    ga('send', 'event', 'CTA click', 'Step 3 completed');
  }
  if (typeof dataLayer !== 'undefined') {
    dataLayer.push({'event': 'Multi-Step', 'eventAction': 'Step ' +
newStep + ' completed'});
  }
}

```



```

    }
  }
  // Step Five
  else if (newStep == 4) {
    $('.step-four').hide();
    $('.step-five').show();
    if (formHeadlineFive !== '') {
      $('.form-headline').children().text(formHeadlineFive);
    }
    if (boxHieghtFive !== '') {
      $('.form-box').css('height', boxHieghtFive);
    }
    if (ctaTopFive !== '') {
      $('button[type="submit"]').css('top', ctaTopFive);
    }
    if (boxTopFive !== '') {
      $('.form-box').css('top', boxTopFive);
    }
    // Mobile code
    if (jQuery(window).width() <= 600) {
      if (mobileBoxHieghtFive !== '') {
        $('.form-box').css('height', mobileBoxHieghtFive);
      }
      if (mobileCtaTopFive !== '') {
        $('button[type="submit"]').css('top', mobileCtaTopFive);
      }
      if (mobileBoxTopFive !== '') {
        $('.form-box').css('top', mobileBoxTopFive);
      }
    }
    if (typeof ga === 'function') {
      ga('send', 'event', 'CTA click', 'Step 4 completed');
    }
    if (typeof dataLayer !== 'undefined') {
      dataLayer.push({'event': 'Multi-Step', 'eventAction': 'Step ' +
newStep + ' completed'});
    }
  }
}

this.progressBarItems.eq(this.currentStep).removeClass('active');
this.progressBarItems.eq(newStep).addClass('active');

this.formContainer.toggleClass('initial', newStep === 0);

```

```

// Update the label of the forward button
var current = parseInt(newStep) + 2;
var total = this.steps.length;
var nextText = this.text.showSteps
? this.text.next + ' (Step ' + current + ' of ' + total + ' )'
: this.text.next;
var submitText = this.text.submit;

var forwardButtonLabel = newStep === this.steps.length - 1 ? submitText :
nextText;

this.forwardButton.children('span').html(forwardButtonLabel);

this.currentStep = newStep;

/*
    *
    *
    *****
    * CTA COPY STEP FUNCTION
    *****
    *
    *
    */
// This is where you can change the cta copy of each step
// The last step is controlled by the editor NOT this script
// The FIRST STEP is controlled at the bottom of this script
// Again step 1 is would equal 0 here
// Add "else if" gate if you want the copy to change throught the steps
if (newStep == 1) {
    if (ctaCopyTwo !== '') {
        this.forwardButton.children('span').html(ctaCopyTwo);
    }
} else if (newStep == 2) {
    if (ctaCopyThree !== '') {
        this.forwardButton.children('span').html(ctaCopyThree);
    }
} else if (newStep == 3) {
    if (ctaCopyFour !== '') {
        this.forwardButton.children('span').html(ctaCopyFour);
    }
}
};

```

```

UnbounceMultiStep.prototype.isValid = function() {
    return
    this.steps.eq(this.currentStep)[0].querySelectorAll(':invalid').length === 0;
};

UnbounceMultiStep.prototype.forwardHandler = function(event) {
    if (!this.isValid()) {
        // If one or more fields on the current step is invalid, prevent going to
        next step. Allow the
        // default click action, which will display the validation errors.
        return;
    }

    if (this.currentStep === this.steps.length - 1) {
        // If we are on the last step, go back to the first step, and allow the
        default click action,
        // which will submit the form.
        //this.goToStep(0);
    } else {
        event.preventDefault();
        this.goToStep(this.currentStep + 1);
    }
};

UnbounceMultiStep.prototype.backHandler = function(event) {
    event.preventDefault();
    this.goToStep(this.currentStep - 1);
};

jQuery.fn.unbounceMultiStep = function(options) {
    this.map(function(index, element) {
        new UnbounceMultiStep(jQuery(element), options);
    });
    return this;
};

jQuery('.lp-pom-form form').unbounceMultiStep({
    steps: stepObject,
    nextButton: startingCTACopy,
    back: backButtonEnabled
});

jQuery('#progress-bar').hide();

```

```
jQuery(window).keydown(function(event) {  
  if (event.keyCode === 13) {  
    event.preventDefault();  
    return false;  
  }  
});
```

```
</script>
```