

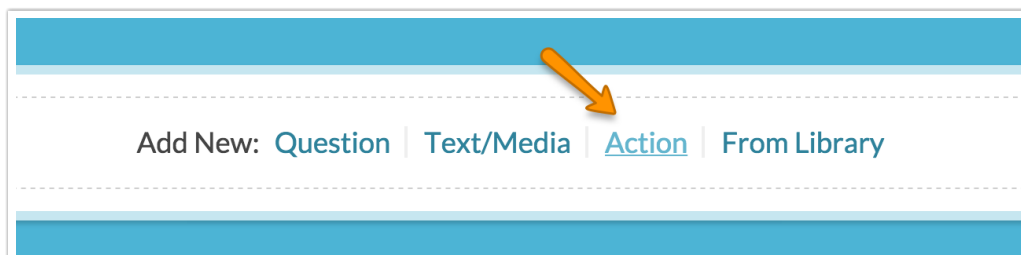
Getting Started with Custom Scripting

Important Update to Custom Scripting

The CustomScript Action now supports the LUA programming language. Visit our [NEW Lua Scripting Resources!](#)

1. New accounts (created after October 29, 2018) will only have the option to use Lua in scripts.
2. As of October 29, 2018 Custom Scripting Actions will default to Lua as the scripting type in the Custom Scripting Action for accounts created before this date. You will be able to switch to the Legacy Custom Scripting; though we highly encourage using Lua.
3. In the long term, Legacy Custom Scripting Actions will be switched to read-only. The exact date on this is to be determined; we will send notifications well ahead of time.

To add a custom script action to your survey, click the **Action** button at the bottom of any page in your survey.



Give your action a name. It's a good idea to describe what the action is doing in the name, for example, "Disable after answering." Select **Custom Script** from the menu and click **Save Action and Edit**.

Next, write or paste in the script you'd like to run.

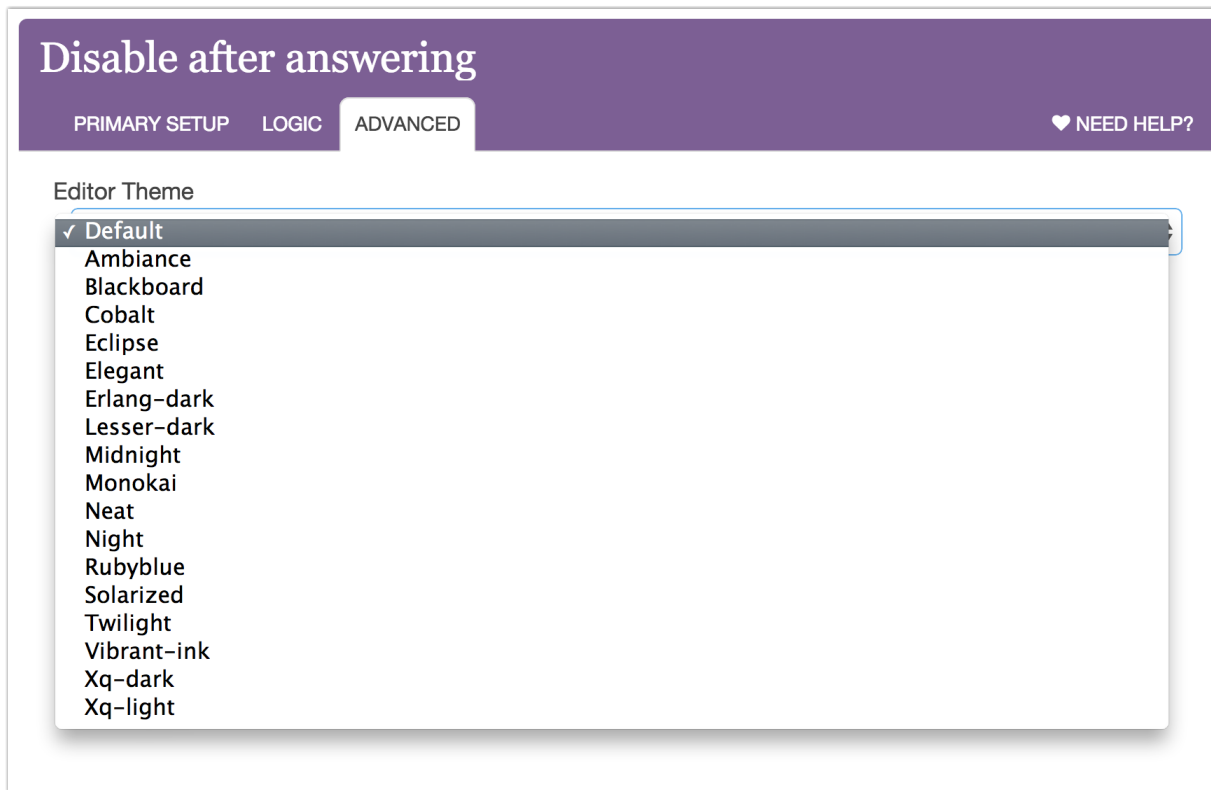
Using Logic in Custom Script Actions

On the **Logic** tab you can set up conditions for when you want your script action to run.

Change the Script Editor Look & Feel

If you want to change the look of the script editor, you can choose a different color scheme on the

Advanced tab of your action.



Important Info to Get Started Scripting

- Custom Scripting actions are processed in order (with questions) from the top of the page to the bottom when the page is rendered. To affect a question on the same page as the script it will need to precede the question on the page.
- All variables begin with %% rather than \$.
- Instead of echo, we use %%output to print to the page. To use this, just store anything you want to print in this variable. When the script runs, all the contents of %%output will be written to the page.
- The following control structures are available: for, foreach, if, elseif, else, while, and switch.
- You cannot define classes and any functions you create must start with 'sgapi.'
- Array elements in Custom Scripts *must* be accessed using curly brace notation if you are using a *variable* as the key. String or numeric keys can be accessed with either square bracket or curly brace notation as normal.

Example:

```
%%myArray = array("foo" => "bar", "this" => "that");
%%arrayKey = "foo";

%%output .= sgapiPrint_R( %%myArray[%%arrayKey] ); // DO NOT USE! prints the whole array
%%output .= sgapiPrint_R( %%myArray{%%arrayKey} ); // print "bar"
```

What Can I Do With Custom Scripting?

The possibilities are vast. Choose your own adventure! Check out our [Working Examples of Custom Scripting](#).

Scripting and Other Out-of-the-Box Customizations

We're always happy to help you debug any documented script. That said, we do not have the resources to write scripts on demand.

If you have customization ideas that you haven't figured out how to tackle, we're happy to be a sounding board for Alchemer features and functionality ideas that might meet your customization. Beyond this, you might want to consult with someone on our [Programming Services Team](#); these folks might have the scripting chops to help you to achieve what you are looking for!

-
-
-

Author: Bri Hillmer

Share This Article:

Last updated: 10/11/2020 2:16 pm EDT