

Show Random Pages or Questions, Limit by Checkbox question

Scripting Solutions

Additional scripting solutions will be added in the future. Please reach out to Alchemer with comments and suggestions on solutions you'd like to see via the link [here](#).

Goal

Show a limited number of random follow up pages (or questions) based on selections made in a Checkbox question.

A **Prioritization Option** ensures the respondent is *a/ways* asked about a specific selection if checked (for example: the Brand you're researching).

See also: [Show Random Pages or Questions](#)

Effort: ✓✓✓

Solution

This solution fills a **Hidden Value Action** with a list of random Reporting Values from a Checkbox question. Later pages or questions use this list to determine if they should be shown or hidden, based on whether it contains a specific value or not.

Step 1: Add checkbox question

Add a **Checkbox question** with options to ask about in follow up pages (questions). These use the Reporting Values of the checkbox later in the Display Logic of the pages (questions) to be randomly shown.

OPTION	REPORTING VALUE
AeroMexico	AMX
Asiana Airlines	AAR
Kenya Airways	KQA
Lufthansa	DLH
United	UAL

Step 2: Add a Hidden Value Action

Add a **Hidden Value Action** to the same page with the title **random-pages-to-show**. The script sets this to a list of randomly selected Reporting Values from the Checkbox question added in Step 1.

? Hidden Value Action

random-pages-to-show

Your hidden value has not been set yet...

Step 3: Add Javascript Action

1. Add a **Javascript Action** to the same page with the code at the end of this article.
2. Set the highlighted values.

Step 4: Add follow up pages (or questions)

Add follow up pages (or questions).

Set the Display Logic for each to only show if the **random-pages-to-show Hidden Value Action** added in Step 2 *contains* one of the Reporting Values from the Checkbox added in Step 1. The text must match exactly, including case.

Example: the follow up page below only displays for Kenya Airways (which has a Reporting Value of **KQA**):

Edit Page

PAGE LOGIC LAYOUT REPEAT ♥ Need Help?

Logic Rule

Remove All Logic

random-pages-to-show contains KQA

+ Add Condition + Add Group

Prioritization Option To prioritize options so they are *always* presented if chosen in the Checkbox question, add an **asterisk** to the end of the Reporting Value, for example **DLH*** below.

Note: do not include the asterisk when setting up the page (or question) display logic in Step 4.

Checkboxes

QUESTION LOGIC VALIDATION LAYOUT PIPING / REPEAT ♥ Need Help?

Question Type

Checkboxes

What question do you want to ask?

What airlines have you flown with

Require this question

Multiple Choice Options

OPTION	REPORTING VALUE
AeroMéxico	AMX
Asiana Airlines	AAR
Kenya Airways	KQA
Lufthansa	DLH*
United	UAL

+ Click to Add Option Add "Other," "N/A," etc Paste in Options

Prioritized

Javascript code:

```
/* Alchemer v01
```

```
Choose a random X brands/concepts to ask follow up questions
based on the brands/concepts the respondent is aware of.
```

```
Documentation and updates: https://help.alchemer.com/help/show-random-x-pages-or-questions-limit-to-aware-re-0f-conceptsbrands
```

```
*/
```

```
document.addEventListener("DOMContentLoaded", function() {
```

```

const AWARENESS_CHECKBOX_QID = 21 // checkbox of brands/concepts respondent is aware of
const SAVE_CONCEPTS_TO_SHOW_QID = 22 // hidden value action to save list of brands to show
const NUMBER_TO_SHOW = 2 // how many concepts/brands to show

// *****
// * no changes needed below *
// *****

const LOG = true

/**
 * Test boolean value, alert() and throw Error if it's false
 *
 * bool {t/f} value to test
 * msg {string} message to alert and throw in new Error
 */
const assert = (bool, msg) => {
  if (!bool) {
    console.error(msg)
    alert(msg)
    throw new Error(msg)
  }
}

/**
 * Get an element based on its Question ID
 *
 * qid {integer/string} question ID
 * section = "element" {string} the final section of the element id
 * return {element} looks for id's in the form: "sgE-1234567-12-123-element"
 */
const getElemByQid = (qid, section = "element") => {
  const id = "sgE-" + SGAPI.survey.surveyObject.id + "-" + SGAPI.survey.pageld + "-" + qid + "-" + section
  const elem = document.getElementById(id)
  assert(elem, "Javascript: can't find element with id = " + id + ", section = " + section)
  return elem
}

/**
 * Get the reporting values for the CHECKED options of
 * a radio button or checkbox QID.
 *
 * qid {int / string} question ID
 * return {array of string} array of reporting values
 */
const getCheckedByQid = (qid) => {

  const checkedElems = getElemByQid(qid, "box").querySelectorAll('.sg-question-options input:checked')
  if (LOG) console.log("checkedElems = ", checkedElems)

  // VARIATION FOR CUSTOMER WHO WANTS TITLES RATHER THAN REPORTING VALUES
  // get the selected checkbox ariaLabel's
  // const checkedTitles = [...checkedElems].map(elem => elem.ariaLabel)
  // if (LOG) console.log("checkedTitles = \n", checkedTitles)
  // return checkedTitles

  const checkedOptionIDs = [...checkedElems].map(elem => elem.value)
  if (LOG) console.log("checkedOptionIDs = ", checkedOptionIDs)

  // get object that maps optionID to reporting value, ex: { "10014": "reporting value1", "10015": "value2"}
  const optionsObj = SGAPI.survey.surveyObject.questions[qid].options

  const checkedReportingValues = checkedOptionIDs.map(optionId => optionsObj[optionId].value )
  if (LOG) console.log("checkedReportingValues = \n", checkedReportingValues)

```

```

    return checkedReportingValues
  }

  /**
   * Shuffle / randomize an array in place
   *
   * array {array} - array is mutated
   * return {array} - shuffled array
   */
  function shuffle(array) {

    // Knuth shuffle (https://stackoverflow.com/questions/2450954/how-to-randomize-shuffle-a-javascript-array)

    var currentIndex = array.length, temporaryValue, randomIndex;

    // While there remain elements to shuffle...
    while (0 !== currentIndex) {

      // Pick a remaining element...
      randomIndex = Math.floor(Math.random() * currentIndex);
      currentIndex -= 1;

      // And swap it with the current element.
      temporaryValue = array[currentIndex];
      array[currentIndex] = array[randomIndex];
      array[randomIndex] = temporaryValue;
    }

    return array;
  }

  /**
   * Sort strings ending with an '*' first, otherwise leave the order the same
   */
  const prioritizeAsteriskFn = (a, b) => {
    if (a.slice(-1) === '*' && b.slice(-1) === '*')
      return 0
    if (a.slice(-1) === '*')
      return -1
    if (b.slice(-1) === '*')
      return 1
    return 0
  }

  /**
   * remove trailing asterisk
   */
  const removeTrailingAsteriskFn = (s) => {
    if (s.slice(-1) === '*')
      return s.slice(0, -1)
    return s
  }

  /**
   * Sort two strings in alpha order
   */
  const alphaSortFn = (a, b) => a.localeCompare(b)

  /**
   * main()
   */
  document.forms[0].addEventListener("submit", function(evt) {

    const LOG = true

```

```
let reportingVals = getCheckedByQid(AWARENESS_CHECKBOX_QID)

reportingVals = shuffle(reportingVals)
if (LOG) console.log("\nshuffled = \n", reportingVals)

reportingVals = reportingVals.sort(prioritizeAsteriskFn)
if (LOG) console.log("\npriority sorted = \n", reportingVals)

reportingVals = reportingVals.map(removeTrailingAsteriskFn)
if (LOG) console.log("\nremove trailing asterisk = \n", reportingVals)

reportingVals = reportingVals.slice(0, NUMBER_TO_SHOW)
if (LOG) console.log("\nnumber to show = \n", reportingVals)

reportingVals = reportingVals.sort(alphaSortFn)
if (LOG) console.log("\nsort by value = \n", reportingVals)

getElemByQid(SAVE_CONCEPTS_TO_SHOW_QID).value = reportingVals.join("")
})
})
```

Related Articles