

Show a Random Number of Rows in a Radio Button Grid or Checkbox Grid 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](#).

Scripting and Other Custom Solutions

We're always happy to help you debug any documented script that is used as is. That said, we do not have the resources to write scripts on demand or to debug a customized script.

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 needs. Beyond this, check out our [Professional Services](#); these folks have the scripting chops to help you to achieve what you are looking for!

Goal

Show a Random Number of Rows in a Radio Button Grid or Checkbox Grid question.

Effort: ✓✓✓

Solution

Step 1 Add a Radio Button Grid or Checkbox Grid question to the page.

Step 2 Add a **Hidden Value Action** to the same page with the title **save-randomly-shown-rows**. Leave the hidden value blank. This is used to save the list of rows that are shown so the same list will be shown if the respondent returns to this page.

Step 3 Set Style > Layout > Mobile Interaction to **Standard**.

Step 4 Add a new **Javascript Action** with the code below and set the highlighted values.

Optional Step 5: Require the question Set **Require this question** and set if answered, minimum rows required to the number of rows you set to show in the script.

Optional Step 6: Hide the page until rows are hidden When the page is

displayed to the user it will briefly show all options in the grid question until the Javascript code determines the random rows to show and hide. To hide the page until the Javascript completes:

1. Set the Page > Layout > CSS Class Name to `hide-page`
2. Add the following to Style tab > HTML/CSS Editor (lower right of page) > Custom CSS:

```
.hide-page {  
  visibility: hidden;  
}
```

Javascript Action code:

```
/* Alchemer v01  
  
  Show random X radio button / checkbox grid rows  
  
  Documentation and updates: https://help.alchemer.com/help/show-random-x-grid-rows  
*/  
  
const NUM_TO_SHOW = 2 // how many options to show  
const GRID_QID = 12 // question ID of the grid question  
const SAVE_SHOWN_QID = 9 // question ID of the Hidden Value Action used to save which rows are shown  
  
// * * * * *  
// * no changes needed below *  
// * * * * *  
  
const LOG = false  
  
document.addEventListener("DOMContentLoaded", function() {  
  
  /**  
   * Test boolean value and alert() and throw Error if it's false  
   *  
   * bool {t/f} value to test  
   * msg {string} message to throw in new Error  
   */  
  const assert = (bool, msg) => {  
    if (!bool) {  
      alert(msg)  
      console.error(msg)  
      throw new Error(msg)  
    }  
  }  
}  
  
  /**  
   * Get an element based on it's 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 surveyInfo = SGAPI.surveyData[Object.keys(SGAPI.surveyData)[0]]  
    const id = "sgE-" + surveyInfo.id + "-" + surveyInfo.currentpage + "-" + qid + "-" + section  
    const elem = document.getElementById(id)
```

```

    assert(elem, "Javascript error: can't find element with id = " + id)
    return elem
  }

  /**
   * Save sorted list of QIDs of elements shown on page to Hidden Value Action
   *
   * saveToQID {int} QID to save the list of row numbers
   * rowsShown {array of int} array of row numbers that were shown
   */
  const saveShown = (saveToQID, rowsShown) => {
    const sortedRows = rowsShown.sort((a, b) => (a - b))
    getElemByQID(saveToQID).value = sortedRows.join(',')
  }

  /**
   * Get row numbers saved from last time page was shown
   *
   * saveToQID {int} QID where list is saved
   * return {array of int / undefined} array of row numbers or undefined if the QID is empty
   */
  const getSavedRows = (saveToQID) => {
    const savedRows = getElemByQID(saveToQID).value
    if (!savedRows || savedRows === 'not-set')
      return undefined
    return savedRows.split(',').map(s => parseInt(s))
  }

  /**
   * Show specific rows and hide all others
   *
   * gridQID {int} the grid question id
   * rowsToShow {array of int} 0-based row numbers to show
   */
  const showRows = (gridQID, rowsToShow) => {
    if (LOG) console.log("showRows() QID = ", gridQID, " rowsToShow = ", rowsToShow)

    const gridElem = getElemByQID(gridQID, 'box')
    const rowElems = gridElem.querySelectorAll('tbody tr')
    for (let i = 0; i < rowElems.length; i++) {
      if (rowsToShow.includes(i))
        rowElems[i].classList.remove('sg-hide')
      else
        rowElems[i].classList.add('sg-hide')
    }
  }

  /**
   * Shuffle array in place
   */
  const shuffle = (a) => {
    for (let i = a.length - 1; i > 0; i--) {
      const j = Math.floor(Math.random() * (i + 1));
      if (LOG) console.log("i,j = ", i, ", ", j);
      [a[i], a[j]] = [a[j], a[i]];
    }
    return a;
  }

  /**
   * Get random rows numbers from grid
   *
   * gridQID {int} the grid QID
   * numToShow {int} how many rows to show

```

```

numToShow [int] how many rows to show
*/
return {array of int} numToShow length array of 0-based row numbers
*/
const getRandomRows = (gridQID, numToShow) => {

  const gridElem = getElemByQID(gridQID, 'box')
  const trElems = gridElem.querySelectorAll('tbody tr')

  let aRowNums = []
  for (let i = 0; i < trElems.length; i++)
    aRowNums.push(i)

  if (LOG) console.log("aRowNums = ", aRowNums)

  return shuffle(aRowNums).slice(0, numToShow)
}

/**
 * Fixes zebra striping for all tables on the page
 */
const fixZebraStriping = () => {

  document.querySelectorAll("table").forEach( table => {
    let oddRow = true
    table.querySelectorAll("tbody tr:not(.sg-hide)").forEach( row => {
      if (LOG) console.log("fixing zebra stripe for row: ", row)

      row.classList.remove("sg-odd-row")
      row.classList.remove("sg-even-row")

      // apply correct sg-odd-row or sg-even-row if not hidden
      if (!row.classList.contains("sg-hide")) {
        row.classList.add(oddRow ? "sg-odd-row" : "sg-even-row")
        oddRow = !oddRow
      }
    })
  })
}

/**
 * main()
 */

const previouslyShownRows = getSavedRows(SAVE_SHOWN_QID)
if (LOG) console.log("previouslyShownRows = ", previouslyShownRows)

if (previouslyShownRows) {
  showRows(GRID_QID, previouslyShownRows)
} else {
  let rowsToShow = getRandomRows(GRID_QID, NUM_TO_SHOW)
  showRows(GRID_QID, rowsToShow)
  saveShown(SAVE_SHOWN_QID, rowsToShow)
}
fixZebraStriping()

// unhide the page
document.querySelector('body').classList.remove('hide-page')
})

```

Related Articles