Randomize Slider Question Left/Right Labels

> ### 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

Randomize the left/right labels of Slider Questions while maintaining proper values for reporting.

Effort: ✔ ✔ ✔

Solution

Add the code found below to a new Javascript Action on the page containing the slider questions that should randomize their left/right labels.

> Special Considerations
>
> - Do <u>not</u> select **Layout** > **Sliders** > **Flip the min and max values** on the Slider questions
> - When clicking on the slider handle the number displayed in the bubble above the handle will be initially incorrect but will update to the correct value as the slider handle is moved.

```
/* Alchemer v01

  Randomize slider left/right prompts.

  Documentation and updates:  https://help.alchemer.com/help/randomize-slider-leftright-labels
*/

// *** NO CHANGES NEEDED BELOW ***

// Turn on logging, only displays properly for one slider on page
// because of timer required to wait for setStyleLeft() to settle down
const LOG = true

/* ----------------------------------
  Helper: Get slider handle element
  ---------------------------------- */
function getHandleElem(slider) { return slider.querySelector(".ui-slider-handle") }

/* ----------------------------------
  Helper: Get slider-setup::json as object
  ---------------------------------- */
function getSliderSetupJSON(slider)      { return JSON.parse(slider.querySelector(".slider-setup").value) }
function setSliderSetupJSON(slider, json) { slider.querySelector(".slider-setup").value = JSON.stringify(json) }

/* -----------------------------------
   Get Min/Max/flipped values of slider, as setup in builder,
   slider-setup::json[min/max]
   ----------------------------------- */
function getMin(slider)     { return getSliderSetupJSON(slider)["min"] }
function getMax(slider)     { return getSliderSetupJSON(slider)["max"] }
function getFlipped(slider) { return getSliderSetupJSON(slider)["flipped"] }

/* -----------------------------------
   Does slider have a value set?  IE, are we showing a slider
   with a value set or did respondent set a value?
   This has nothing to do with a default/starting value.
   ----------------------------------- */
function isSet(slider) {
  return (   slider.querySelector(".ui-slider-horizontal-blank") === null
          && slider.querySelector(".ui-slider-vertical-blank") === null)
}

/* -----------------------------------
   Get/Set left/right prompts
   ----------------------------------- */
function getLeftPrompt(slider)  { return slider.querySelector(".js-label-slider-left").innerHTML }
function getRightPrompt(slider) { return slider.querySelector(".js-label-slider-right").innerHTML }
function setLeftPrompt(slider, newValue)  { slider.querySelector(".js-label-slider-left").innerHTML = newValue }
function setRightPrompt(slider, newValue) { slider.querySelector(".js-label-slider-right").innerHTML = newValue }
function swapPrompts(slider) {
  let leftPrompt = getLeftPrompt(slider)
  setLeftPrompt(slider, getRightPrompt(slider))
  setRightPrompt(slider, leftPrompt)
}

/* -----------------------------------
   Get/Set .ui-slider-handle::aria-valuenow
   ----------------------------------- */
function getAriaValuenow(slider) { return getHandleElem(slider).getAttribute("aria-valuenow") }
function setAriaValuenow(slider, newValue) { return getHandleElem(slider).setAttribute("aria-valuenow", newValue) }
function swapAriaValuenow(slider) {
  let newValue = getMax(slider) - getAriaValuenow(slider) + getMin(slider)
  setAriaValuenow(slider, newValue)
}

/* -----------------------------------
   Get/Set slider::input.value
   ----------------------------------- */
function getValue(slider) { return slider.querySelector(".sg-slider-hidden-value input").value }
function setValue(slider, newValue) { slider.querySelector(".sg-slider-hidden-value input").value = newValue }
function swapValue(slider) {
  // Test using the the arai value to fix the single click issue
  // let newValue = getMax(slider) - inputElem.value
  let newValue = getMax(slider) - getAriaValuenow(slider) + getMin(slider)
  setValue(slider, newValue)
}

/* -----------------------------------
   Get/Set slider-setup.value::json("sliderval")
   ----------------------------------- */
function getSlidervalue(slider) {
  getSliderSetupJSON(slider)["sliderval"]
}
function setSlidervalue(slider, newValue) {
  let json = getSliderSetupJSON(slider)
  json["sliderval"] = newValue
  setSliderSetupJSON(slider, json)
}
function swapSlidervalue(slider) {
  let newValue = getMax(slider) - getSlidervalue(slider) + getMin(slider)
  setSlidervalue(slider, newValue)
}

/* -----------------------------------
   Get/Set slider-setup.value::json("startval")
   ----------------------------------- */
function getStartval(slider) {
  return getSliderSetupJSON(slider)["startval"]
}

/* -----------------------------------
   Get/Set handle.style.left/botton

   swap() uses a setTimeout() because the SG app's code dynamically moves
   the selector using its own timer over 350ms.  We need to wait
   for this to complete before setting the style.

   The style value includes a % sign, example: style="left: 20%;"
   But the get/set are numeric only, example: 20
   Note:  horizontal slider is "left:20%", vertical is "bottom:20%"
   ----------------------------------- */
function getStyleLeft(slider) {
  let value = null
  if (getHandleElem(slider).getAttribute("style").startsWith("left"))
    value = getHandleElem(slider).style.left
  else
    value = getHandleElem(slider).style.bottom
  return parseFloat(value, 10).toFixed(1)
}
function setStyleLeft(slider, newValue) {
  if (getHandleElem(slider).getAttribute("style").startsWith("left"))
    getHandleElem(slider).style.left = newValue + "%"
  else
    getHandleElem(slider).style.bottom = newValue + "%"
}
function swapStyleLeft(slider) {
  let newValue = 100 - getStyleLeft(slider)
  setTimeout(function() { setStyleLeft(slider, newValue) }, 400)
  setTimeout(function() { setStyleLeft(slider, newValue) }, 1000) // ensure it worked!
}

/* -----------------------------------
   Get/set Bubble::div::innerText
   ----------------------------------- */
function getBubble(slider) {
  let elem = slider.querySelector(".sg-slider-bubble-horizontal div")
  return (elem) ? parseInt(elem.innerText, 10) : null
}
function setBubble(slider, newValue) {
  slider.querySelector(".sg-slider-bubble-horizontal div").innerText = newValue
}
function hasBubble(slider) {
  return (slider.querySelector(".sg-slider-bubble-horizontal")) ? true : false
}
function swapBubble(slider) {
  let newValue = getMax(slider) - getBubble(slider) + getMin(slider)
  setBubble(slider, newValue)
}

/* -----------------------------------
   logCurrStatus
   ----------------------------------- */
```

```
function logCurrStatus(slider, heading, delay) {

  function logIt() {
    console.log("--- Status: ", heading, " ---")
    console.log("isSet     = ", isSet(slider))
    console.log("prompts   = ", getLeftPrompt(slider), " - ", getRightPrompt(slider))
    console.log("max       = ", getMax(slider))
    console.log("startval  = ", getStartval(slider))
    console.log("value     = ", getValue(slider))
    console.log("styleLeft = ", getStyleLeft(slider))
    console.log("valuenow  = ", getAriaValuenow(slider))
    console.log("sliderval = ", getSlidervalue(slider))
    console.log("bubble    = ", getBubble(slider))
  }

  if (LOG) {
    if (delay) {
      setTimeout( function() { logIt() }, 1000)
    }
    else {
      logIt()
    }
  }
}

/* -----------------------------------
   Swap a slider display
   ----------------------------------- */
function swapSliderDisplay(slider) {
  if (LOG) {
    console.log("-------------------")
  }
  logCurrStatus(slider, "initial", false)

  swapPrompts(slider)

  // Init slider that's not set but has a default value.
  // These values will be flipped below.
  if (getStartval(slider) && !isSet(slider)) {
    let startval = getStartval(slider)
    setValue(slider, startval)
    setStyleLeft(slider, (startval-getMin(slider)) / (getMax(slider)-getMin(slider)) * 100) // StyleLeft wants a percentage
    setAriaValuenow(slider, startval)
    // setBubble(slider, startval) CAN'T SET BUBBLE, IT DOESN'T EXIST YET
    setSlidervalue(slider, startval)
  }

  // Flip slider that's set or has a default value
  if (isSet(slider) || getStartval(slider)) {
    swapValue(slider)
    swapStyleLeft(slider)
    swapAriaValuenow(slider)
    if (hasBubble(slider))  // false when there is a Startval but no selection has been made
      swapBubble(slider)
    swapSlidervalue(slider)
  }
  // init slider that's not set and doesn't have a default value
  else {
    setValue(slider, 0)
    setStyleLeft(slider, 0)
    setAriaValuenow(slider, 0)
    // setBubble(slider, 0) CAN'T SET BUBBLE, IT DOESN'T EXIST YET
    setSlidervalue(slider, 0)
  }
  logCurrStatus(slider, "swapped", true)
}

/* ====================================
   DOM loaded
   ==================================== */
document.addEventListener("DOMContentLoaded", function() {

  let isSwapped = [] // boolean array of swap status for all sliders on page
  let sliders = document.getElementsByClassName("sg-type-slider")


  /* -----------------------------------
     randomly swap slider displays
     ----------------------------------- */
  function swapRandomSliders() {


    // go through all sliders
    for (let sliderIdx = 0; sliderIdx < sliders.length; sliderIdx++) {


      // fail is slider is setup with flipped scales
      if (getFlipped(sliders[sliderIdx])) {
        console.log("JAVASCRIPT / PAGE SETUP ERROR, attempt to swap slider with flipped scales: ")
        console.log(sliders[sliderIdx])
        alert("JAVASCRIPT / PAGE SETUP ERROR, attempt to swap slider with flipped scales, see console")
        return
      }


      // randomly decide if slider should be swapped
      let randomBool = Math.floor( Math.random() * 2 );
      isSwapped.push(randomBool)


      // swap it
      if (randomBool)
        swapSliderDisplay(sliders[sliderIdx])
    }
  }

  /* -----------------------------------
     save swapped sliders
     ----------------------------------- */
  function saveSwappedSliders() {


    // go through all sliders
    for (let sliderIdx = 0; sliderIdx < sliders.length; sliderIdx++) {


      let slider = sliders[sliderIdx]


      // if swapped, swap the value being saved or clear it
      if (isSwapped[sliderIdx]) {
        logCurrStatus(slider, "Saving start", false)
        if (isSet(slider))
          swapValue(slider)
        else
          setValue(slider, null)  // otherwise we'll save a value the user didn't set
        logCurrStatus(slider, "Saving end", false)
        if (LOG)
          console.log("** SAVED value = ", slider.querySelector(".sg-slider-hidden-value input").value)
      }
    }
    if (LOG)
      console.log(" ")
  }


  /* ==================================
     DOM loaded actions
     ================================== */

  swapRandomSliders()

  document.forms[0].onsubmit = function() {
    saveSwappedSliders()
  }
})
```

Related Articles