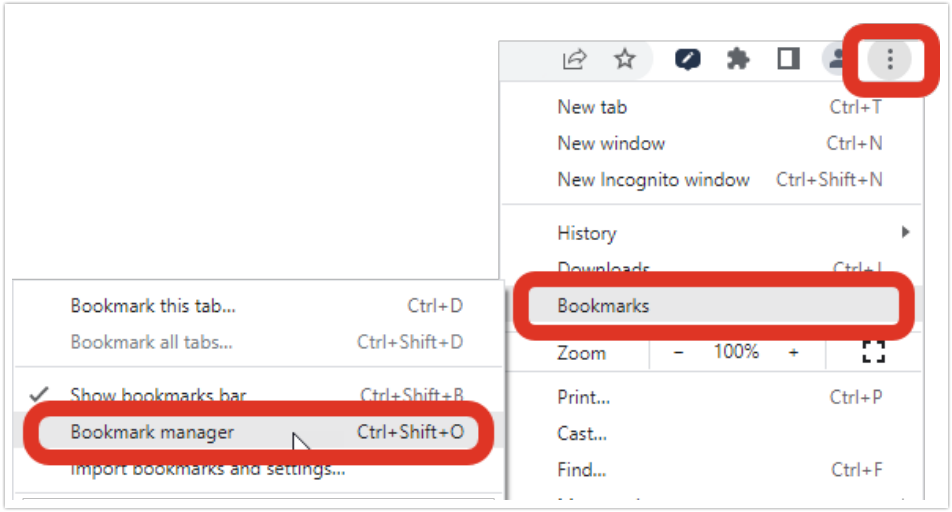
Use Autofill JavaScript to Save Time Taking and Testing Surveys

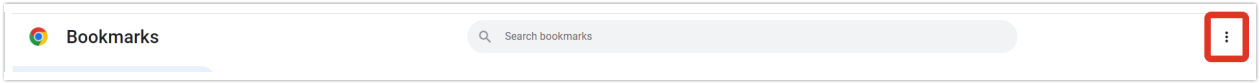
Ever get tired of clicking through your survey while testing? Adding some JavaScript to a bookmark can save you a ton of time! This tutorial will walk you through adding this bookmark to your browser.

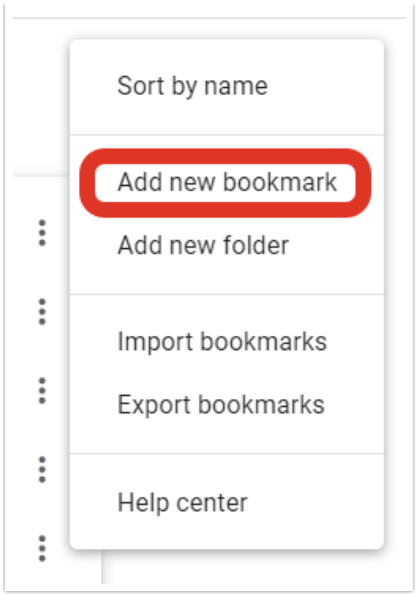
1. First, create a new bookmark in your browser. We are using Google Chrome but you can do this in any browser.

2. In Chrome click **Bookmarks** > **Bookmark manager**.



3. Next, click the three vertical dots in the upper righthand corner of the page (not the browser!), and then click **Add New Page**.





4. A new field will pop up. In our example, we are making the name *AutoFill*. After you name your new Bookmark, copy and paste the code below into the **URL** (Be sure to scroll all the way to the end!)

javascript:%2F* Alchemer v03%0A%0A Autofill Chrome Bookmarklet%0A%0A Works with question types%3A%0A checkbox (with optional min number of selections or optional write-in)%0A checkbox grid (with optional min number of selections)%0A conjoint%0A continuous sum (with optional required total)%0A essay%0A dropdown%0A dropdown menu grid%0A dropdown menu list%0A image multi select (but doesn%27t look at min number...yet)%0A image select%0A likert%0A max diff%0A nps%0A radio button (with optional write-in)%0A radio button grid%0A ranking grid%0A semantic diff%0A slider (optional min%2Fmax and step)%0A slider list (same as above)%0A star rating grid%0A textbox%0A -- email%0A -- date (all three formats but not min%2Fmax date)%0A -- numeric with min%2Fmax number or min%2Fmax number of characters%0A -- currency%0A -- percent%0A -- phone (based on the regex option)%0A textbox grid (same as above)%0A textbox list (same as above)%0A%0A Custom Groups containing any of the above%0A%0A Not working for%0A actions%0A audo sentiment%0A cascading dropdown%0A custom table%0A drag and drop ranking%0A file upload%0A grouping open %2F closed card sort%0A image heatmap%0A text highlighter%0A quick sort%0A signature%0A video feedback%0A video sentiment%0A%0A*%2F%0A%0A(function() %7B%0Aconst LOG %3D true%0Aconst BOOKMARKLET %3D true %2F%2F true when setting up as bookmarlet%2C false to run in the survey Style > HEADER%0A%0A%2F**%0A * Returns a random integer between min (inclusive) and max (inclusive).%0A * https%3A%2F%2Fstackoverflow.com%2Fquestions%2F1527803%2Fgenerating-random-whole-numbers-in-javascript-in-a-specific-range%0A *%2F%0Afunction getRandomInt(min%2C max) %7B%0A min %3D Math.ceil(min)%3B%0A max %3D Math.floor(max)%3B%0A return Math.floor(Math.random() * (max - min %2B 1)) %2B min%3B%0A%7D%0A%0A%2F***%0A * Shuffle array in place%0A * Knuth shuffle%3A https%3A%2F%2Fstackoverflow.com%2Fquestions%2F2450954%2Fhow-to-randomize-shuffle-a-javascript-array%0A *%0A * array (array) will be mutated%0A * return (array) the original array after being shuffled%0A *%2F%0Afunction shuffle(array) %7B%0A var currentIndex %3D array.length%2C temporaryValue%2C randomIndex%3B%0A%0A %2F%2F While there remain elements to shuffle...%0A while (0 !%3D%3D currentIndex) %7B%0A%0A %2F%2F Pick a remaining element...%0A randomIndex %3D Math.floor(Math.random() * currentIndex)%3B%0A currentIndex -%3D 1%3B%0A%0A %2F%2F And swap it with the current element.%0A temporaryValue %3D array%5BcurrentIndex%5D%3B%0A array%5BcurrentIndex%5D %3D array%5BrandomIndex%5D%3B%0A array%5BrandomIndex%5D %3D temporaryValue%3B%0A %7D%0A%0A return array%3B%0A%7D%0A%0A%2F**%0A * Parse an Alchemer element %23ID in the form%3A%0A * sgE-5901811-28-305-box%0A * sgE-5901811-28-305-10997-element%0A * return (obj) Returns object of the constituent parts%0A *%2F%0Aconst parseSgId %3D (id) %3D> %7B%0A const regexID %3D %2FsgE-(%5Cd%2B)-(%5Cd%2B)-(%5Cd%2B)(-(%5Cd%2B))%3F-(%5Cw%2B)%2F%0A%0A const aParsed %3Did.match(regexID)%0A%0A if (!aParsed %7C%7C aParsed.length !%3D%3D 7) %7B%0A alert(%27Javascript error parsing ID %3D %27%2C id)%0A %7D%0A%0A return %7B%0A sid%3A aParsed%5B1%5D%2C %2F%2F ex%3A %275901811%27%0A pid%3A aParsed%5B2%5D%2C %2F%2F ex%3A %2728%27%0A qid%3A aParsed%5B3%5D%2C %2F%2F ex%3A %27305%27%0A oid%3A aParsed%5B5%5D%2C %2F%2F ex%3A %2710997%27 or undefined if this isn't an ID for an option%0A type%3A aParsed%5B6%5D %2F%2F ex%3A %27box%27 %2F %27element%27%0A %7D%0A%7D%0A%0A%2F***%0A * Get the survey%27s SGAPI variable%0A *%0A * return (obj) the SGAPI global vraible%0A *%2F%0Aconst getSGAPI %3D () %3D> %7B%0A %2F%2F Preview is in an iFrame%0A const iFrameElem %3D document.querySelector(%27iframe%23preview-the-page%27)%0A if (iFrameElem)%0A return iFrameElem.contentWindow.SGAPI%0A return sgapi %3D SGAPI%0A%7D%0A%0A%2F***%0A * Get the survey%27s document elemen

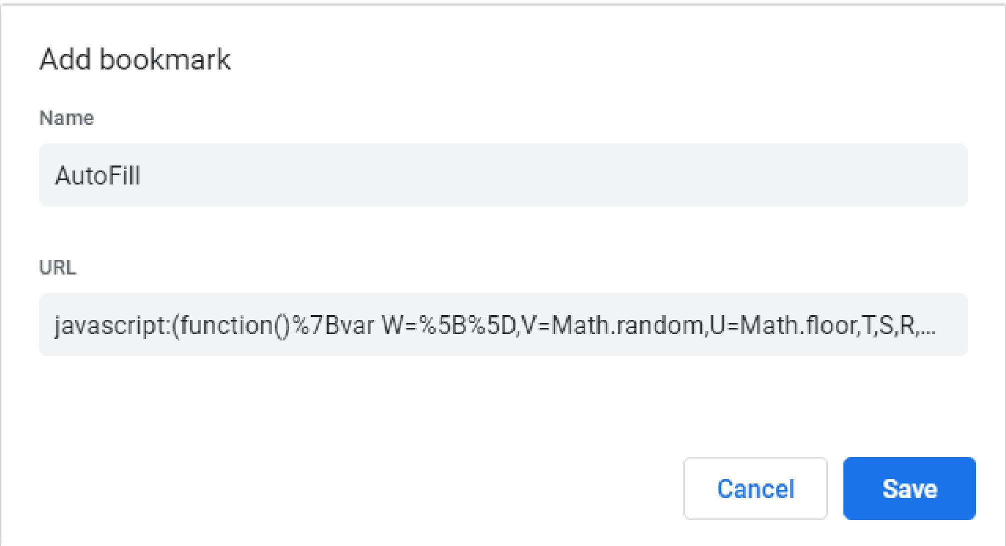
t%0A *%0A * return (elem)%0A *%2F%0Aconst getDocument %3D () %3D> %7B%0A %2F%2F Preview is in an iFrame
%0A const iFrameElem %3D document.querySelector(%27iframe%23preview-the-page%27)%0A if (iFrameElem)%0A
return iFrameElem.contentDocument %7C%7C iFrameElem.contentWindow.document%0A return document%0A%7D%0
A%0A%2F***%0A * Get a property from SGAPI for the qid%0A *%0A * qid (int) question ID%0A * propertyName (string)
name of property%0A * isInt (t%2Ff) convert return value to int if true%0A * return (int%2Fstring) the property or 0 %2F %
27%27 if property doesn%27t exist%0A *%2F%0Aconst getProperty %3D (qid%2C propertyName%2C isInt) %3D> %7B
%0A const val %3D getSGAPI().survey.surveyObject.questions%5Bqid%5D.properties%5BpropertyName%5D%0A if (isI
nt)%0A return parseInt(val) %7C%7C 0%0A return val %7C%7C %27%27%0A %7D%0A%0A%2F***%0A * autofill dr
opdowns%0A *%0A * questionElem (element)%0A * return (t%2Ff) true if auto-filled%2C false if there was already a valu
e%0A *%2F%0Aconst dropdowns %3D (questionElem) %3D> %7B%0A let autoFilled %3D false%0A const selectElems
%3D questionElem.querySelectorAll(%27select%27)%0A selectElems.forEach(selectElem %3D> %7B%0A if (LOG) cor
sole.log("selectElem.value %3D "%2C selectElem.value)%0A %2F%2F Don%27t change if dropdown already has a valu
e%0A %2F%2F %27NoAnswer%27 for dropdown and dropdown menu list%2C %27%27 for dropdown menu grid%0
A if (selectElem.value %3D%3D%3D %27NoAnswer%27 %7C%7C selectElem.value %3D%3D%3D %27%27) %7B%0
A autoFilled %3D true%0A const numOptions %3D selectElem.querySelectorAll(%27option%27).length%0A sel
ectElem.selectedIndex %3D getRandomInt(1%2C numOptions - 1) %2F%2F 1 to skip past "-- Please Select --"%2C and -
1 since this is 0-based%0A %7D%0A %7D)%0A return autoFilled%0A%7D%0A%0A%2F***%0A * autofill star rating
grid%0A *%0A * questionElem (element)%0A * return (t%2Ff) true if auto-filled%2C false if there was already a value%0
A *%2F%0Aconst starRatingGrid %3D (questionElem) %3D> %7B%0A let autoFilled %3D false%0A questionElem.quer
ySelectorAll(%27tbody td%27).forEach(tdElem %3D> %7B%0A if (!tdElem.querySelector(%27input%3Achecked%27))
%7B%0A autoFilled %3D true%0A const labelElems %3D tdElem.querySelectorAll(%27label%27)%0A const ra
ndom %3D getRandomInt(1%2C labelElems.length - 1) %2F%2F 1 to skip the initial X%2C -1 since it%27s zero-based%0
A if (LOG) console.log("-- selecting stars %3D "%2C random)%0A for (let i %3D 1%3B i <%3D random%3B i%2B%
2B)%0A labelElems%5Bi%5D.classList.add(%27sg-star-on%27)%0A labelElems%5Brandom%5D.querySelector(
%27input%27).checked %3D true%0A %7D%0A %7D)%0A return autoFilled%0A%7D%0A%0A%2F***%0A * autofill
continuous sum%0A *%0A * questionElem (element)%0A * return (t%2Ff) true if auto-filled%2C false if there was alread
y a value%0A *%2F%0Aconst continuousSum %3D (questionElem) %3D> %7B%0A%0A const inputElems %3D questio
nElem.querySelectorAll(%27tbody input%5Btype%3Dtext%5D%27)%0A%0A for (let i %3D 0%3B i < inputElems.length
%3B i%2B%2B) %7B%0A if (inputElems%5Bi%5D.value !%3D%3D %27%27)%0A return false%0A %7D%0A%0A
const qid %3D parseSgId(questionElem.id).qid%0A %2F%2F const maxTotal %3D parseInt(SGAPI.survey.surveyObject.q
uestions%5Bqid%5D.properties.max_total) %7C%7C 0%0A const maxTotal %3D getProperty(qid%2C %27max_total%2
7%2C true)%0A if (maxTotal) %7B%0A const val %3D Math.floor(maxTotal %2F inputElems.length)%0A for (let i %3
D 0%3B i < inputElems.length - 1%3B i%2B%2B) %7B%0A inputElems%5Bi%5D.value %3D val%0A %7D%0A in
putElems%5BinputElems.length - 1%5D.value %3D maxTotal - (val * (inputElems.length - 1))%0A %7D%0A else %7B
%0A inputElems.forEach(inputElem %3D> inputElem.value %3D getRandomInt(0%2C10))%0A %7D%0A%0A %2F%2
F update the total%0A inputElems%5B0%5D.focus()%0A inputElems%5B0%5D.blur()%0A%0A return true%0A%7D%
0A%0A%2F***%0A * autofill slider%0A *%0A * questionElem (element) can be a question or a sliderRowElem for a slider
list%0A * return (t%2Ff) true if auto-filled%2C false if there was already a value%0A *%2F%0Aconst slider %3D (questio
nElem) %3D> %7B%0A%0A %2F***%0A * Get a random value and percent%0A *%0A * return (obj of int) %7B rand
omValue%2C randomPercent %7D%0A *%2F%0A const getRandomValueAndPercent %3D () %3D> %7B%0A%0A c
onst setupObj %3D JSON.parse(questionElem.querySelector(%27.slider-setup%27).value)%0A if (LOG) console.log("set
upObj %3D "%2C setupObj)%0A const steps %3D Math.floor((setupObj.max - setupObj.min) %2F setupObj.stepval)%0
A const randomValue %3D setupObj.min %2B getRandomInt(0%2C steps) * setupObj.stepval%0A const randomPerce
nt %3D Math.floor(((randomValue - setupObj.min) %2F (setupObj.max - setupObj.min)) * 100)%0A if (LOG) console.log("
random val %2F percent %3D "%2C randomValue%2C %27 %2F %27%2C randomPercent%2C %27%25%27)%0A%0A
return %7B randomValue%2C randomPercent %7D%0A %7D%0A%0A %2F***%0A * main()%0A *%2F%0A%0A
%2F%2F already set%2C don%27t change%0A if (questionElem.querySelector(%27input.sg-input%27).value)%0A retu
rn false%0A%0A const %7B randomValue%2C%0A randomPercent %7D %3D getRandomValueAndPercent()%0A
%0A %2F%2F set slider value%0A questionElem.querySelector(%27input.sg-input%27).value %3D randomValue%0A%
0A %2F%2F set the slider handle%2C this must be on a timer b%2Fc of how the slider functions%0A const sliderHandle
Elem %3D questionElem.querySelector(%27.ui-slider-handle%27)%0A setTimeout(function () %7B sliderHandleElem.styl
.left %3D %60%24%7BrandomPercent%7D%25%60 %7D%2C 400)%0A setTimeout(function () %7B sliderHandleElem.s
tyle.left %3D %60%24%7BrandomPercent%7D%25%60 %7D%2C 1000) %2F%2F ensure it worked!%0A return true%0
A%7D%0A%0A%2F***%0A * autofill sliderList%0A *%0A * questionElem (element)%0A * return (t%2Ff) true if auto-fill
ed%2C false if there was already a value%0A *%2F%0Aconst sliderList %3D (questionElem) %3D> %7B%0A let autoFill
ed %3D false%0A questionElem.querySelectorAll(%27.sg-slider-row%27).forEach(sliderRowElem %3D>%0A autoFille
d %3D slider(sliderRowElem) %7C%7C autoFilled)%0A return autoFilled%0A%7D%0A%0A%2F***%0A * autofill textbo
xes%0A *%0A * questionElem (element)%0A * return (t%2Ff) true if auto-filled%2C false if there was already a value%0A
*%2F%0Aconst textboxes %3D (questionElem) %3D> %7B%0A%0A const qid %3D parseSgId(questionElem.id).qid%0
A%0A %2F***%0A * Get the validation parameters for min%2Fmax charachter count. or 0 if not set%0A *%0A * retur
n (obj of ints) %7B minCharacters%2C maxCharacters %7D%0A *%2F%0A const getMinMaxCharacters %3D () %3D>
%7B%0A %2F%2Fconst qid %3D parseSgId(questionElem.id).qid%0A const minCharacters %3D getProperty(qid%2C
%27min_characters%27%2C true)%0A const maxCharacters %3D getProperty(qid%2C %27max_characters%27%2C tr
ue)%0A%0A if (LOG) console.log("textboxes()%2C qid %3D "%2C qid)%0A if (LOG) console.log("- min%2Fmax chars
%3D "%2C minCharacters%2C %27 %2F %27%2C maxCharacters)%0A return %7B minCharacters%2C maxCharacters

%7D%0A %7D%0A%0A %2F***%0A * Get the validation parameters for min%2Fmax number%2C or 0 if not set%0A *%0A * return (obj of ints) %7B minNumber%2C maxNumber %7D%0A *%2F%0A const getMinMaxNumber %3D () %3D> %7B%0A %2F%2Fconst qid %3D parseSgId(questionElem.id).qid%0A const minNumber %3D getProperty(qid %2C %27min_number%27%2C true)%0A const maxNumber %3D getProperty(qid%2C %27max_number%27%2C true)%0A%0A if (LOG) console.log("textboxes()%2C qid %3D "%2C qid)%0A if (LOG) console.log("- min%2Fmax number %3D "%2C minNumber%2C %27 %2F %27%2C maxNumber)%0A return %7B minNumber%2C maxNumber %7D%0A %7D%0A%0A %2F***%0A * Get the input mask (regex validation)%0A *%0A * reutrn (string) the question property i nputmask.MASK or empty string%0A *%2F%0A const getInputMask %3D () %3D> %7B%0A const inputMask %3D ge tProperty(qid%2C %27inputmask%27%2C false)%0A const retval %3D (inputMask) %3F inputMask.MASK %3A %27% 27 %0A if (LOG) console.log("getInputMask() %3D "%2C retval)%0A return retval%0A %7D%0A%0A %2F***%0A * main()%0A *%2F%0A%0A let autoFilled %3D false%0A%0A const %7B minCharacters%2C%0A maxCharacter s %7D %3D getMinMaxCharacters()%0A%0A let %7B minNumber%2C%0A maxNumber %7D %3D getMinMaxNum ber()%0A minNumber %3D Math.ceil(minNumber)%0A maxNumber %3D Math.floor(maxNumber)%0A%0A const inputE lems %3D questionElem.querySelectorAll(%27input%5Btype%3Dtext%5D%27)%0A inputElems.forEach(inputElem %3D > %7B%0A%0A %2F%2F only fill if there%27s no value%0A if (!inputElem.value) %7B%0A%0A autoFilled %3D tr ue%0A%0A const classList %3D inputElem.classList%0A%0A %2F%2F EMAIL%0A if (classList.contains(%27sg-validation-email%27))%0A inputElem.value %3D %27test%40test.com%27%0A%0A %2F%2F DATE%0A else if (classList.contains(%27sg-validation-date%27)) %7B%0A if (classList.contains(%27sg-validation-date-yyyy%27)) %0A inputElem.value %3D %272025%2F01%2F01%27%0A else%0A inputElem.value %3D %2701%2F0 1%2F2025%27%0A %7D%0A%0A %2F%2F NUMERIC%0A else if (classList.contains(%27sg-validation-nume ric%27) %0A %7C%7C classList.contains(%27sg-validation-percent%27)%0A %7C%7C classList.contain s(%27sg-validation-currency%27)) %7B%0A if (minNumber %26%26 maxNumber)%0A inputElem.value %3D getRandomInt(minNumber%2C maxNumber)%0A else if (minNumber)%0A inputElem.value %3D getRandomInt(minNumber%2C minNumber %2B 20)%0A else if (maxNumber)%0A inputElem.value %3D getRandomInt(0%2C maxNumber)%0A else if (minCharacters)%0A inputElem.value %3D %271%27.repeat(minCharacters)%0A el se%0A inputElem.value %3D %27123%27.slice(0%2C maxCharacters %7C 3)%0A %7D%0A%0A %2F%2F U S PHONE (from the Alchemer regex for a US Phone%2C note%3A the backslashes are escaped so they appear doubled)% 0A else if (getInputMask() %3D%3D%3D "%5E((%5C%5C(%5C%5Cd%7B3%7D%5C%5C) %3F)%7C(%5C%5Cd%7B 3%7D%5B-%5C%5Cs%5D))%3F%5C%5Cd%7B3%7D%5B-%5C%5Cs%5D%5C%5Cd%7B4%7D%24") %7B%0A i putElem.value %3D %27123-456-7890%27%0A %7D%0A%0A %2F%2F OTHERWISE%2C normal text%0A el se %7B%0A if (LOG) console.log("otherwise%2C normal text")%0A if (minCharacters) %7B%0A if (minChara cters %3D%3D%3D 5) %2F%2F special case for zip%0A inputElem.value %3D %2712345%27%0A else%0A inputElem.value %3D %27x%27.repeat(minCharacters)%0A %7D%0A else %7B%0A inputElem.valu e %3D %27test%27.slice(0%2C maxCharacters %7C 4)%0A %7D%0A %7D%0A%0A %2F%2F fire display logi c on later questions%0A inputElem.focus()%0A inputElem.blur()%0A %7D%0A %7D)%0A return autoFilled%0A %7D%0A%0A%2F***%0A * autofill essay%0A *%0A * questionElem (element)%0A * return (t%2Ff) true if auto-filled%2 C false if there was already a value%0A *%2F%0Aconst essay %3D (questionElem) %3D> %7B%0A%0A const textareaE lem %3D questionElem.querySelector(%27textarea%27)%0A%0A %2F%2F if already has a value%2C do nothing%0A if (textareaElem.value)%0A return false%0A%0A textareaElem.value %3D %27test%27%0A%0A %2F%2F fire display l ogic on later questions%0A textareaElem.focus()%0A textareaElem.blur()%0A return true%0A%7D%0A%0A%2F***%0 A * autofill radio button%0A *%0A * questionElem (element) question or other elem type for radio button grid or conjoint% 0A * return (t%2Ff) true if auto-filled%2C false if there was already a value%0A *%2F%0Aconst radioButton %3D (questi onElem) %3D> %7B%0A if (LOG) console.log("radioButton %3D "%2C questionElem)%0A %2F%2F if already selected %2C do nothing%0A if (questionElem.querySelectorAll(%27input%5Btype%3Dradio%5D%3Achecked%27).length)%0A return false%0A%0A const radioElems %3D questionElem.querySelectorAll(%27input%5Btype%3Dradio%5D%27)%0A %0A const radioElem %3D radioElems%5BgetRandomInt(0%2C radioElems.length - 1)%5D%0A if (LOG) console.log("cli cking "%2C radioElem)%0A%0A %2F%2F fire display logic on later questions%0A radioElem.click()%0A%0A %2F%2F check for Other Write In%0A if (radioElem.parentElement.classList.contains(%27sg-other-li%27))%0A radioElem.parent Element.querySelector(%27input%5Btype%3Dtext%5D%27).value %3D %27write-in%27%0A%0A return true%0A%7 D%0A%0A%2F***%0A * autofill radio button grid%0A *%0A * questionElem (element)%0A * return (t%2Ff) true if auto-f illed%2C false if there was already a value%0A *%2F%0Aconst radioButtonGrid %3D (questionElem) %3D> %7B%0A let autoFilled %3D false%0A const trElems %3D questionElem.querySelectorAll(%27tbody tr%27)%0A trElems.forEach(trE lem %3D>%0A autoFilled %3D radioButton(trElem) %7C%7C autoFilled)%0A return autoFilled%0A%7D%0A%0A%2F ***%0A * autofill image select and image multi select%0A *%0A * questionElem (element)%0A * return (t%2Ff) true if aut o-filled%2C false if there was already a value%0A *%2F%0Aconst imageSelect %3D (questionElem) %3D> %7B%0A%0A if (questionElem.querySelector(%27.sg-image-selected%27))%0A return false%0A%0A const imageSelectElems %3D questionElem.querySelectorAll(%27.sg-image-box label%27)%0A const imageSelectElem %3D imageSelectElems%5Bge tRandomInt(0%2C imageSelectElems.length - 1)%5D%0A if (LOG) console.log("clicking "%2C imageSelectElem)%0A%0A %2F%2F fire display logic on later questions%0A imageSelectElem.click()%0A%0A return true%0A%7D%0A%0A%2F* **%0A * autofill conjoint -- all pages%0A *%0A * questionElem (element)%0A * return (t%2Ff) true if auto-filled%2C fals e if there was already a value%0A *%2F%0Aconst conjoint %3D (questionElem) %3D> %7B%0A%0A if (questionElem.q uerySelector(%27input%5Btype%3Dradio%5D%3Achecked%27))%0A return false%0A%0A const conjointSetElems % 3D questionElem.querySelectorAll(%27.sg-conjoint-set%27)%0A for (let i %3D 0%3B i < conjointSetElems.length%3B i %2B%2B) %7B%0A if (LOG) console.log("%5Cnconjoint set %3D "%2C conjointSetElems%5Bi%5D)%0A radioButton(conjointSetElems%5Bi%5D)%0A%0A if (i !%3D%3D conjointSetElems.length - 1)%0A document.querySelector(%27.

sg-next-button.btn-conjoint%27).click()%0A %7D%0A return true%0A%7D%0A%0A%2F***%0A * autofill max diff -- al
l pages%0A *%0A * questionElem (element)%0A * return (t%2Ff) true if auto-filled%2C false if there was already a value
%0A *%2F%0Aconst maxDiff %3D (questionElem) %3D> %7B%0A%0A if (questionElem.querySelectorAll(%27input%5
Btype%3Dradio%5D%3Achecked%27).length)%0A   return false%0A%0A const maxDiffSetElems %3D questionElem.q
uerySelectorAll(%27.sg-maxdiff-set%27)%0A for (let i %3D 0%3B i < maxDiffSetElems.length%3B i%2B%2B) %7B%0A
  if (LOG) console.log("%5CmaxDiff set %3D "%2C maxDiffSetElems%5Bi%5D)%0A%0A   const trElems %3D shuffle(%
5B...maxDiffSetElems%5Bi%5D.querySelectorAll(%27tbody tr%27)%5D)%0A   trElems%5B0%5D.querySelectorAll(%27
nput%5Btype%3Dradio%5D%27)%5B0%5D.click()%0A   trElems%5B1%5D.querySelectorAll(%27input%5Btype%3Drad
o%5D%27)%5B1%5D.click()%0A%0A   if (i !%3D%3D maxDiffSetElems.length - 1)%0A   document.querySelector(%27
.sg-next-button%27).click()%0A %7D%0A return true%0A%7D%0A%0A%2F***%0A * autofill ranking grid%0A *%0A *
 questionElem (element)%0A * return (t%2Ff) true if auto-filled%2C false if there was already a value%0A *%2F%0Aconst
 rankingGrid %3D (questionElem) %3D> %7B%0A%0A if (questionElem.querySelectorAll(%27tbody tr input%5Btype%3
Dradio%5D%3Achecked%27).length)%0A   return false%0A%0A const trElems %3D questionElem.querySelectorAll(%2
7tbody tr%27)%0A%0A %2F%2F get a randomized array of ints %5B0..trElems.length-1%5D%0A let aRanking %3D %
5B%5D%0A for (let i %3D 0%3B i < trElems.length%3B i%2B%2B)%0A   aRanking.push(i)%0A aRanking %3D shuffle(
aRanking)%0A%0A trElems.forEach((trElem%2C idx) %3D> %7B%0A   const inputElems %3D trElem.querySelectorAll(
%27input%5Btype%3Dradio%5D%27)%0A   inputElems%5BaRanking%5Bidx%5D%5D.click()%0A %7D)%0A%0A retu
rn true%0A%7D%0A%0A%2F***%0A * autofill checkbox%0A *%0A * checkboxElem (element) a question for a checkbox
 OR a TR for a checkbox grid row%0A * return (t%2Ff) true if auto-filled%2C false if there was already a value%0A *%2F
%0Aconst checkbox %3D (questionElem%2C isCheckboxGridRow %3D false) %3D> %7B%0A%0A console.log("checkbo
x() questionElem %3D "%2C questionElem)%0A%0A %2F%2F if already checked%2C do nothing%0A if (questionElem.q
uerySelectorAll(%27input%5Btype%3Dcheckbox%5D%3Achecked%27).length)%0A   return false%0A%0A %2F%2F ch
eckboxes%0A const checkElems %3D questionElem.querySelectorAll(%27input%5Btype%3Dcheckbox%5D%27)%0A%0
A %2F%2F the minimum number of checks based on the Validation for the checkbox question or checkbox grid%0A let mi
nChecks %3D undefined%0A if (isCheckboxGridRow) %7B%0A   %2F%2F checkbox grid TRs have a class name in the fo
rm %27row-12%27%2C where 12 is the QID%0A   %2F%2Fconst row_qid %3D %5B...questionElem.classList%5D.find(s
 %3D> s.slice(0%2C 4) %3D%3D%3D %27row-%27)%0A   const row_qid %3D %5B...questionElem.classList%5D.find(s
%3D> s.startsWith(%27row-%27))%0A   const qid %3D parseInt(row_qid.slice(4))%0A   minChecks %3D getProperty(qi
d%2C %27min_answers_per_row%27%2C true)%0A %7D%0A else %7B%0A   const qid %3D parseSgId(questionElem
.id).qid%0A   minChecks %3D getProperty(qid%2C %27minimum_response%27%2C true)%0A %7D%0A minChecks %
3D Math.min(minChecks%2C checkElems.length) %7C%7C 1%0A console.log("minChecks %3D "%2C minChecks)%0A%
0A %2F%2F check the min number of checkboxes and fire display logic on later questions%0A let checked %3D 0%0A
while (checked < minChecks) %7B%0A   const random %3D getRandomInt(0%2C checkElems.length - 1)%0A   if (!checkE
lems%5Brandom%5D.checked) %7B%0A   const checkElem %3D checkElems%5Brandom%5D%0A   checkElem.click(
)%0A   checked%2B%2B%0A%0A   %2F%2F check for Other Write In%0A   if (checkElem.parentElement.classList.c
ontains(%27sg-other-li%27))%0A   checkElem.parentElement.querySelector(%27input%5Btype%3Dtext%5D%27).valu
e %3D %27write-in%27%0A   %7D%0A %7D%0A return true%0A%7D%0A%0A%2F***%0A * autofill checkbox grid
%0A *%0A * questionElem (element)%0A * return (t%2Ff) true if auto-filled%2C false if there was already a value%0A *
%2F%0Aconst checkboxGrid %3D (questionElem) %3D> %7B%0A let autoFilled %3D false%0A const trElems %3D que
stionElem.querySelectorAll(%27tbody tr%27)%0A trElems.forEach(trElem %3D>%0A   autoFilled %3D checkbox(trElem
%2C true) %7C%7C autoFilled)%0A return autoFilled%0A%7D%0A%0A%2F***%0A * autofill the page%0A *%0A * Thi
s function uses a Timeout to recurse. The Timeout allows the survey%27s%0A * display logic engine to run and we go thr
ough the questions again to%0A * fill any new ones that were displayed.%0A *%0A * questionElems (arr of elems) all que
stion on the page including hidden%0A *%2F%0Aconst autofill %3D (questionElems) %3D> %7B%0A%0A let autoFilled
AnyQuestion %3D false%0A%0A questionElems.forEach(questionElem %3D> %7B%0A%0A   if (LOG) console.log("%5
Cn----------%5CnquestionElem %3D "%2C questionElem)%0A%0A   let autoFilled %3D false%0A%0A   if (!q
uestionElem.classList.contains(%27sg-hide%27)) %7B%0A   if (LOG) console.log("-- autopopulating%3A "%2C question
Elem.id)%0A%0A   %2F%2F CHECKBOX%0A   if (questionElem.classList.contains(%27sg-type-checkbox%27))%0A
  autoFilled %3D checkbox(questionElem)%0A   %2F%2F CHECKBOX GRID%0A   else if (questionElem.classList.cont
ains(%27sg-type-table-checkbox%27))%0A   autoFilled %3D checkboxGrid(questionElem)%0A%0A   %2F%2F CONJ
OINT%0A   else if (questionElem.classList.contains(%27sg-type-conjoint_new%27))%0A   autoFilled %3D conjoint(qu
estionElem)%0A%0A   %2F%2F CONTINUOUS SUM%0A   else if (questionElem.classList.contains(%27sg-type-contin
uous-sum%27))%0A   autoFilled %3D continuousSum(questionElem)%0A%0A   %2F%2F DROPDOWN%0A   %2F
%2F DROPDOWN MENU LIST%0A   %2F%2F DROPDOWN MENU GRID%0A   else if (questionElem.classList.contai
ns(%27sg-type-menu%27)%0A   %7C%7C questionElem.classList.contains(%27sg-type-multimenu%27)%0A
  %7C%7C questionElem.classList.contains(%27sg-type-table-menu-matrix%27))%0A   autoFilled %3D dropdowns(
questionElem)%0A%0A   %2F%2F ESSAY%0A   else if (questionElem.classList.contains(%27sg-type-essay%27))%0A
  autoFilled %3D essay(questionElem)%0A%0A   %2F%2F IMAGE SELECT%0A   %2F%2F IMAGE MULTI SELECT
%0A   else if (questionElem.classList.contains(%27sg-type-imageselect%27))%0A   autoFilled %3D imageSelect(ques
tionElem)%0A%0A   %2F%2F MAX DIFF%0A   else if (questionElem.classList.contains(%27sg-type-maxdiff%27))%0
A   autoFilled %3D maxDiff(questionElem)%0A%0A   %2F%2F RADIO BUTTON%0A   else if (questionElem.classLis
t.contains(%27sg-type-radio%27))%0A   autoFilled %3D radioButton(questionElem)%0A   %2F%2F RADIO BUTTON
 GRID%0A   else if (questionElem.classList.contains(%27sg-type-table-radio%27))%0A   autoFilled %3D radioButton
Grid(questionElem)%0A%0A   %2F%2F RANKING GRID%0A   else if (questionElem.classList.contains(%27sg-type-ran
k-table%27))%0A   autoFilled %3D rankingGrid(questionElem)%0A%0A   %2F%2F SEMANTIC DIFF%0A   else if (q

```
uestionElem.classList.contains(%27sg-type-table-semantic%27))%0A       autoFilled %3D radioButtonGrid(questionElem)
%0A%0A     %2F%2F SLIDER%0A     else if (questionElem.classList.contains(%27sg-type-slider%27))%0A       autoFille
d %3D slider(questionElem)%0A     %2F%2F SLIDER LIST%0A     else if (questionElem.classList.contains(%27sg-type-m
ulti-slider%27))%0A       autoFilled %3D sliderList(questionElem)%0A%0A     %2F%2F STAR RATING GRID%0A     else
if (questionElem.classList.contains(%27sg-type-table-stars%27))%0A       autoFilled %3D starRatingGrid(questionElem)%
0A%0A     %2F%2F TEXTBOX %2F TEXTBOX LIST %2F TEXTBOX GRID%0A     else if (   questionElem.classList.contains
(%27sg-type-textbox%27)%0A           %7C%7C questionElem.classList.contains(%27sg-type-multitext%27)%0A
   %7C%7C questionElem.classList.contains(%27sg-type-table-textbox%27))%0A       autoFilled %3D textboxes(questio
nElem)%0A%0A     %2F%2F UNKOWN%2C IGNORE%0A     else %7B%0A       console.log("-- question type not recogni
zed%2C ignored")%0A     %7D%0A     if (LOG %26%26 !autoFilled) console.log("-- Already set")%0A     autoFilledAnyQ
uestion %3D autoFilled %7C%7C autoFilledAnyQuestion%0A     questionElem.scrollIntoView()%0A   %7D%0A   else %
7B%0A     console.log("-- question hidden%2C igore it")%0A   %7D%0A  %7D)%0A  if (autoFilledAnyQuestion) %7B%0
A   if (LOG) console.log("%5Cn--------------------%5Cn>>>> Looping to see if display logic popped up anything else")%0
A   setTimeout(function() %7B%0A     autofill(questionElems)%0A   %7D%2C 500) %2F%2F wait to allow the survey en
gine%27s display logic to fire%0A  %7D%0A  else %7B%0A   if (LOG) console.log("%5CnDONE!")%0A  %7D%0A%7D%
0A%0A%2F***%0A * Get array of questions on the page (pulling up questions in a Custom Group)%0A *%2F%0Aconst g
etQuestionElems %3D () %3D> %7B%0A%0A  let docElem %3D getDocument()%0A%0A  if (LOG) console.log("docElem
%3D "%2C docElem)%0A%0A  const a %3D %5B...docElem.querySelector(%27.sg-question-set%27).children%5D%0A%
0A  const questionElems %3D %5B%5D%0A%0A  a.forEach(elem %3D> %7B%0A%0A   %2F%2F if elem is a Custom G
roup%2C look for the question elements in it%0A   if (elem.classList.contains(%27sg-type-group%27)) %7B%0A     %2F
%2F the questions in a Custom Group are .sg-queston%27s under .sg-group-item elements%0A     const groupItemElems
%3D elem.querySelectorAll(%27.sg-group-item%27)%0A     groupItemElems.forEach(groupItemElem %3D>%0A       qu
estionElems.push(groupItemElem.querySelector(%27.sg-question%27)))%0A   %7D%0A   %2F%2F else elem is a questi
on itself%0A   else %7B%0A     questionElems.push(elem)%0A   %7D%0A  %7D)%0A%0A  return questionElems%0A
 %7D%0A%0A%2F***%0A * main()%0A *%2F%0A%0Aif (BOOKMARKLET) %7B%0A  const questionElems %3D getQu
estionElems()%0A  console.log("%5Cn--------------------%5CnquestionElems %3D "%2C questionElems%2C %27%5C
n%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5Cn%27)%0A a
utofill(questionElems)%0A%7D%0Aelse %7B%0A  document.addEventListener("DOMContentLoaded"%2C function() %7
B%0A   const questionElems %3D getQuestionElems()%0A   console.log("%5Cn--------------------%5CnquestionElems
 %3D "%2C questionElems%2C %27%5Cn%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5E%5
E%5E%5E%5E%5E%5E%5Cn%27)%0A   autofill(questionElems)%0A  %7D)%0A%7D%0A%7D)()%0A%0A
```

5. Your new bookmark will look something like this. Once completed click **Save**.



Now it's time to test!

Pull up a survey with questions. On any page that requires input, hit your new Autofill button in the Bookmark Toolbar. This should auto populate all questions on that page.

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!

Related Articles