

# Basic Math

## 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](#).

You can perform basic math using either [Javascript](#) or [Lua](#) script in Alchemer.

### Javascript Alternative

A **Javascript Action** allows users to perform basic math and easily save the results to a Textbox question or Hidden Value Action. For example:

The screenshot displays a survey editor interface with two pages. Page 1 (ID: 4) contains a question '1. Number of days' with radio button options 1, 2, 3, 4, and 5. Page 2 (ID: 12) contains a question '2. Cost per day' with a text input field. Below the input field is a 'Hidden Value Action' section with a warning icon and the text 'calculated-total-cost Your hidden value has not been set yet...'. At the bottom is a 'JavaScript Action' section with the text 'Perform calculation'. A red box highlights the text 'Save Number of Days \* Cost per day' within the JavaScript Action section. Three red arrows point from this box to the radio button options on Page 1, the text input field on Page 2, and the warning message of the Hidden Value Action.

Only the colored rows of the **Javascript Action** below need to be changed to perform math in your survey. This example shows how to get a value from both an earlier page and the current page.

`// get the numeric reporting value for a question on an earlier page using a merge code`

```
const numberOfDays = parseInt(` [question("value"), id="35"] `) || 0
```

The above uses a merge code to get an integer value from question ID 35 on an *earlier* page. If the question isn't answered, then numberOfDays will be set to zero. In this example the previous question was a Radio Button with numeric reporting values. The question could have been a Textbox, Hidden Value Action, or anything that contains a single, numeric value.

parseInt() is used to convert the string to an integer.

```
// get the numeric value for a question on this page based on its question ID  
const costPerDay = parseFloat(getElemByQid(39).value) || 0
```

Use the function `getElemByQid(##).value` to get a value for question ID 39 on the *current* page. If the question isn't answered, costPerDay will be set to zero. This question must be a Textbox question or Hidden Value Action.

parseFloat() is used to convert the string to a floating point number assuming the respondent will enter a currency with a decimal. If they will only enter an integer use `parseInt()` as above.

```
// perform some math!  
let totalCost = numberOfDays * costPerDay
```

Perform math on the values.

```
// save the value rounded to two decimal digits (important since floating point math in  
Javascript is base-2)  
getElemByQid(40).value = totalCost.toFixed(2)
```

Use the function `getElemByQid(##).value` to set a value on question ID 40 on the *current* page. The `.toFixed(2)` function is used to ensure the math calculation is rounded to decimal places. Because floating point math is performed in base-2 on computers and converted to base-10, some calculations produce repeating decimal approximations like 3.20000000001 instead of 3.20.

If the calculation is performed using only integer values, you do not need `.toFixed(2)`.

The complete Javascript Action:

```

/* Alchemer example of performing basic math in a survey

Documentation and updates: https://help.alchemer.com/help/basic-math
*/

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

  /**
   * 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
  }

  /**
   * main()
   */

  // wait for the Back / Next button to be clicked to submit the page
  document.forms[0].addEventListener("submit", function() {

    // get the numeric reporting value for a question on an earlier page using a merge code
    const numberOfDays = parseInt(` [question("value", id="35")] `) || 0

    // get the numeric value for a question on this page based on its question ID
    const costPerDay = parseFloat(getElemByQid(39).value) || 0

    // perform some math
    const totalCost = numberOfDays * costPerDay

    // save the value rounded to two decimal digits (important since floating point math in Javascript is base-2)
    getElemByQid(40).value = totalCost.toFixed(2)

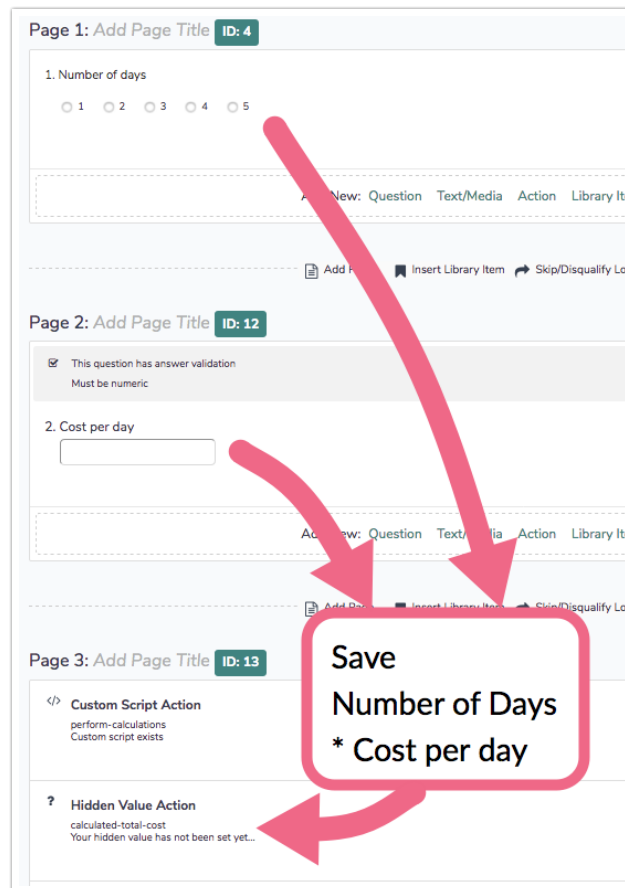
  })
})

```

## Lua Script Alternative

A Lua Script Action allows you to perform basic math and easily save the results to a Textbox

question or Hidden Value Action. For example:



The **Lua Script Action** doesn't require as much setup code so it's more concise than the **Javascript Action**. The `getvalue()` and `setvalue()` functions are used get and set values from any page of the survey. The test for `nil` is used in case a question wasn't answered.

The biggest difference from the Javascript example is that the **Lua Script Action** and the **Hidden Value Action** that holds the calculated value must be on a later page than the two values used in the calculation.

```
numberOfDays = getvalue(35)
if (numberOfDays == nil) then
  numberOfDays = 0
end

costPerDay = getvalue(39)
if (costPerDay == nil) then
  costPerDay = 0
end

setvalue(40, numberOfDays * costPerDay)
```

Related Articles