

Validate Textbox Answer Formats

When using [Textbox](#) questions, often you will want to validate the data your respondents enter. It makes analyzing, reporting, and then acting on your data easier when it all conforms to certain formatting.

Basic Validation

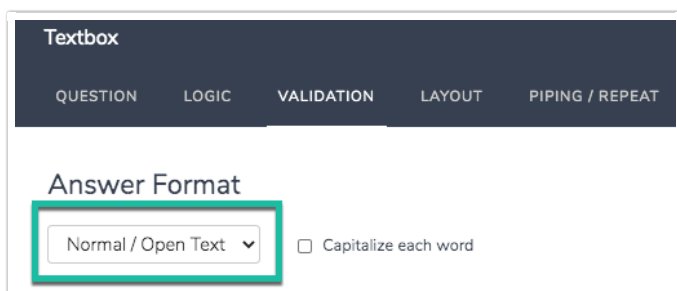
Email addresses, phone numbers, even IP addresses are easy to collect with the proper answer formatting that is built into Alchemer. With a little extra effort, you can create much more complicated validation patterns on your own!

In addition to a standard Textbox question, you are given the option to choose from the following pre-formatted question types:

- [Number](#)
- [Email](#)
- [Percent](#)
- [Date](#) (either MM/DD/YYYY or DD/MM/YYYY)

These questions will automatically apply answer formatting that can be adjusted from the **Validation** tab.

You can also apply formatting validation to a Textbox question after the fact on the **Validation** tab. If you select a *numeric validation*, you can choose to **Force Whole/Positive Numbers** and/or **Allow Values Between** a minimum and maximum range.



The screenshot shows the configuration interface for a Textbox question. At the top, there are five tabs: QUESTION, LOGIC, VALIDATION, LAYOUT, and PIPING / REPEAT. The VALIDATION tab is currently selected. Below the tabs, the 'Answer Format' section is visible. It contains a dropdown menu set to 'Normal / Open Text' and a checkbox labeled 'Capitalize each word' which is currently unchecked. A red rectangular box highlights the dropdown menu.

(Optional) When using validation we recommend displaying an example format to your survey respondents. This is particularly useful if you are using complex validation patterns. A good place for this is the additional instructions section of your question title.

To add example formatting, edit your question and click on the **Layout** tab. Add your example format in the **Additional Instructional Text** field and click **Save Question**.

Textbox
QUESTION LOGIC VALIDATION **LAYOUT** PIPING / REPEAT

Additional Instructional Text

abc@def.com

B *I* Size - **A** - Merge Codes

Show instructions above the question title

RegEx Patterns for Complex Validation

A Regular Expression pattern or RegEx pattern is a sequence of characters that is used to validate specific string formats in user input. These patterns are commonly used to ensure the data collected is properly formatted.

For example, if you were collecting telephone numbers and it was essential they were valid US telephone numbers, you could apply a RegEx pattern to the question. In doing so, all input for the telephone number field would have to match an (xxx) xxx-xxxx pattern. If the input did not match that pattern, an error message would display indicating that it must match the appropriate format.

Preloaded RegEx Patterns in Alchemer

Below are the commonly used RegEx patterns readily available inside Alchemer:

Numbers & Currency:

- A number 1-4 digits in length: `^\d{1,4}$`
- A whole number percentage with '%' sign: `^\d{1,3}\%$`
- A whole number or up to two decimal places: `^\d+(\.\d{1,2})?$`
- Currency (Dollars and Cents with commas): `^\$?([1-9]{1}[0-9]{0,2}(\,[0-9]{3})*(\.[0-9]{0,2})?)|[1-9]{1}[0-9]{0,}(\.[0-9]{0,2})?|0(\.[0-9]{0,2})?|(\.[0-9]{1,2})?$`
- Currency (Euros with periods and comma): `^\e?([1-9]{1}[0-9]{0,2}(\.[0-9]{3})*(\,[0-9]{0,2})?)|[1-9]{1}[0-9]{0,}(\,[0-9]{0,2})?|0(\,[0-9]{0,2})?|(\,[0-9]{1,2})?$`
- Currency (French with spaces and comma): `^\e?([1-9]{1}[0-9]{0,2}(\s[0-9]{3})*(\,[0-9]{0,2})?)|[1-9]{1}[0-9]{0,}(\,[0-9]{0,2})?|0(\,[0-9]{0,2})?|(\,[0-9]{1,2})?$`
- Range of number 0..999: `^([0-9]|[1-9][0-9]|[1-9][0-9][0-9])$`

Contact Info & Demographics

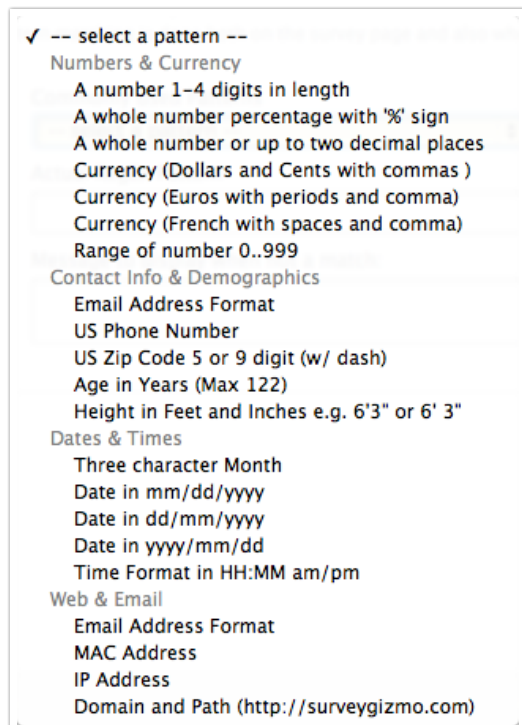
- Email Address Format: `^([a-zA-Z0-9_-\.\+])*(\+[a-z0-9-])?@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)(\|[a-zA-Z0-9\-\+]\.))([a-zA-Z]{2,4}|[0-9]{1,3})(\)?)$`
- US Phone Number: `^((\(\d{3}\)\)|(\d{3}[-\s]))?\d{3}[-\s]\d{4}$`
- US Zip Code 5 or 9 digit (w/ dash): `^\d{5}([\-]\d{4}){0,1}$`
- Age in Years (Max 122): `^([0-9]|[1-9][0-9]|[1-9][0-1][0-9]|[1-9]2[0-2])$`
- Height in Feet and Inches e.g. 6'3" or 6' 3": `^([1-8]')?\s?([1-9]|1[01])`

Dates & Times

- **Three character Month:** `^(Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sept|Oct|Nov|Dec)$`
- **Date in mm/dd/yyyy:** `^(0[1-9]|1[012])[\/] (0[1-9]|12)[0-9]|3[01])[\/] (19|20)\d\d$`
- **Date in mm/dd/yyyy or m/d/yyyy:** `^(0?[1-9]|1[012])[\/] (0?[1-9]|12)[0-9]|3[01])[\/] (19|20)\d\d$`
- **Date in dd/mm/yyyy:** `^(0?[1-9]|12)[0-9]|3[01])[\/] (0?[1-9]|1[012])[\/] (19|20)\d\d$`
- **Date in yyyy/mm/dd:** `^(19|20)\d\d[\/] (0?[1-9]|1[012])[\/] (0?[1-9]|12)[0-9]|3[01])$`
- **Time Format in HH:MM am/pm:** `^([1-9]|1[012]):(0[0-9]|1[1-5][0-9])\s?(am|AM|pm|PM)$`

Web & Email

- **Email Address Format:** `^([a-zA-Z0-9_-\.\+])*(\+[a-z0-9-])?@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. |([a-zA-Z0-9-]+\.)+)([a-zA-Z]{2,4}|[0-9]{1,3})(\?)?$`
- **MAC Address:** `^([0-9a-fA-F][0-9a-fA-F]:){5}([0-9a-fA-F][0-9a-fA-F])$`
- **IP Address:** `^(((0-9)|1-9)[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}((0-9)|1-9)[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$`
- **Domain and Path (http://alchemer.com):** `^(http|https):\/\/[a-z0-9]+([\-\.\.]{1}[a-z0-9]+)*\.[a-z]{2,5}(:[0-9]{1,5})?(\/.*)?$`



Set Up RegEx

To set up a RegEx pattern on a specific question, select **Edit > Validation** and click the **Answer Format** dropdown menu. Choose **RegEx**.

Now you will see a field for **Regex Validation**. This is where you can select the RegEx pattern that you wish to implement or write your own. You can also specify a message to display if the respondent's input does not match the given pattern.

Answer Format

RegEx ⌵

RegEx Validation

`^([1-8])?\s?([1-9])1[01]`

Common RegEx Patterns

Height in Feet and Inches e.g. 6'3" or 6' 3"

Message to display when answer does not pass RegEx

Sorry about that, we need your height in feet AND inches (6'1" or 5' 5")!

Other Useful RegEx Patterns

If the RegEx pattern you are looking for is not available in the above list, here are a few additional RegEx patterns that are helpful.

- **Match only email addresses from Gmail:** `[^]*@gmail\.com$`
- **Match only serial numbers with 4 numbers, dash, 4 numbers:** `^[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]$`
- **Match only first names that start with R or r and have a last name:** `^[Rr][^] [a-zA-Z]+$`
- **Match the numbers 30, 31 or 32:** `^30$|^31$|^32$`
- **Match any dates in July in the format MM/DD/YYYY:** `^07/[0-9][0-9]/[0-9][0-9][0-9][0-9]$`
- **Match only pure alpha strings:** `^[a-zA-Z]+$`
- **Match only letters, numbers and spaces (Removes special characters):** `^[a-zA-Z0-9]+$`
- **Match only letters, numbers and underscores:** `^[a-zA-Z0-9_]*$`
- **Character Limit, min=1 max=10:** `^(.\|s){1,10}$`
- **Number between 1-10:** `^([0-9]|1[0])$`
- **UK ZIP codes:** `^([A-PR-UWYZ0-9][A-HK-Y0-9][AEHMNPRTVXY0-9]?[ABEHMNPRTVWX0-9]? {1,2}[0-9][ABD-HJLN-UW-Z]{2}|GIR 0AA)$`
- **Allow apostrophes in email addresses:** `^([0-9a-zA-Z](['-.\w]*[0-9a-zA-Z])*)*@[([0-9a-zA-Z]+([-'\w']*[0-9a-zA-Z])*\.)+[a-zA-Z]{2,9})$`
- **Smart telephone number (allows 1234567891, (123) 456-7891, and 123-456-7891):** `[()?[2-9][0-9][0-9][0-9]?[-|]?[0-9][0-9][0-9][0-9]?[-|]?[0-9][0-9][0-9][0-9]`

- **UK phone number:** `^\s*\(?((\+0?44)?\?)?[\ -]?(\\(0\\))|0)((20[7,8]{1}\\)\?)?[\ -]?[1-9]{1}[0-9]{2}[\ -]?[0-9]{4})|([1-8]{1}[0-9]{3}\\)\?)?[\ -]?[1-9]{1}[0-9]{2}[\ -]?[0-9]{3})\s*$`
- **US ZIP, US ZIP +4, Canada postal code:** `^\d{5}-\d{4}|\d{5}|[A-Z]\d[A-Z] \d[A-Z]\d\d$`
- **Time in 24 hour format:** `^([0-1]?[0-9]|2[0-4]):([0-5][0-9])(:([0-5][0-9]))?$`
- **Match English characters only:** `^[A-Za-z0-9'\.,&@:;?!()$#'"]+$`

If the RegEx pattern you are looking for is not listed here, you can always write your own RegEx pattern in the **Regex Validation** field.

Other Useful RegEx Resources

Below are a few helpful resources for RegEx Validation:

- Test your regular expressions: [RegEx Tester](#)
- A RegEx library to search other commonly used patterns: [RegExLib](#)
- Reg Ex Cheat Sheet: [Cheat Sheet](#)

Logic and RegEx

Let's say you asked for respondents' email addresses in a survey. Let's also assume that you wanted to ask only respondents who have Gmail accounts a follow-up question. Alchemer offers two operators: **matches regex pattern** and **does not match pattern** for setting up a logic condition like this.

1. Click **Edit** on the follow-up question that you want to show to respondents who answered that they have Gmail accounts, then click on the **Logic** tab.
2. On the **Logic** tab, you should see an option to **Only show this question based on answers to previous questions or other logic conditions**. Check this option.
3. Select the source question from the first dropdown menu (the question where the email is collected). Now specify if you want the RegEx pattern to be matched or not for the logic to run.
4. Finally, input the following RegEx code into the third text field to complete the logic:

```
[^ ]*@gmail\.com$.
```

This will show our follow-up question only if the email collected ends in "@gmail.com".

The screenshot shows the 'Logic Rule' configuration screen in Alchemer. The 'LOGIC' tab is active. A checkbox labeled 'Only show this question based on answers to previous questions or other logic conditions' is checked. Below this, there is a 'Remove All Logic' link. A logic rule is defined with three components: a dropdown menu showing 'Email Address', a dropdown menu showing 'matches regex', and a text input field containing the RegEx pattern '[^]*@gmail.com\$'. At the bottom, there are buttons for '+ Add Condition' and '+ Add Group'.

HTML 5

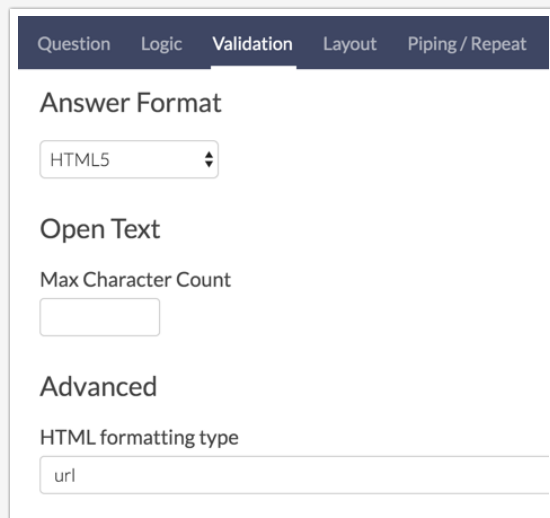
The HTML5 formatting type allows you to use HTML5 input types to validate Textbox questions. These input types are simple, one or two-word commands that function in similar ways to RegEx but allow for more flexibility. In the example below, you can see the input type **URL** inside this Textbox question, validating the response to ensure a URL is entered.

HTML5 Supported Browsers

Not all browsers (or versions) support HTML5. Google Chrome is the most compatible with HTML5, but Firefox is a close second. If you would like to see what your particular browser can support, please visit [HTML5 Test](#). If you visit the Forms section, you will see what input types are supported.

Set Up HTML5 Input Validation

Click **Edit > Validation** and click the **Answer Format** dropdown menu. Choose **HTML5**. Enter the input type you would like to use in the **HTML formatting type** field and then click **Save Question** to see the input type take effect!



The screenshot shows a form editor interface with a dark navigation bar at the top containing the tabs: Question, Logic, Validation (selected), Layout, and Piping / Repeat. Below the navigation bar, the 'Validation' section is expanded to show the 'Answer Format' dropdown menu set to 'HTML5'. Underneath, the 'Open Text' section includes a 'Max Character Count' input field. The 'Advanced' section contains the 'HTML formatting type' field, which has 'url' entered.

Available HTML5 Input Types

We recommend checking out [html5 doctor](http://html5doctor.com/html5-forms-input-types/) for all the input types that are available: <http://html5doctor.com/html5-forms-input-types/>. The following are the ones we find most useful for Alchemer customers.

If you'd like to use the URL input type, all you have to do is type **URL** into the HTML Formatting Type field.

If you want a nice looking date picker, use **date**.

3. Please enter the date of your arrival.

October 2014

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

4. P

If you want to mask the input of a password, use `password`. *The password input type is not available on Free or Trial accounts.*

4. Please enter your password.

We've found that `search`, `datetime`, `email`, `tel` are not as useful as their RegEx or built-in counterparts and `color` is very temperamental. If an input type is not supported by the browser viewing that question, it will behave as a regular Textbox.

Related Articles