**Some commonly used regex formula examples:**

*Examples related to use of numbers (for all text questions that use numbers, do not forget to type* ***numbers*** *under the appearance column)*

* Restrict mobile number to ten digits : **regex(., '^[0-9]{10}$')** or **regex(., '^\d{10}$')**
* Restrict an input to **1234.56.78** format: **regex(., '^[0-9]{4}.[0-9]{2}.[0-9]{2}$')** or **regex(., '^\d{4}\.\d{2}\.\d{2}$')**
* Restrict an input between **01** to **99** digits where input format of a single number (like 1 or 2) is not allowed: **regex(., '^[01-99]{2}$') and (. >= 01)**
* Restrict an input to only two numbers i.e. either to **12** or **345**: **regex(., '^(12|345)$')**
* Restrict an input of nine digits where the first number can't be **0**: **regex(., '^1-9][0-9]{8}$') or regex(., '^[^0$')**
* Restrict an input to one digit in between **0** to **9**: **regex(., '^\d$')**
* Restrict an input to five digits in between **0** to **9**: **regex(., '^\d{5}$')**
* Restrict an input to two digits and three decimals (e.g. **12.345**): **regex(., '\d{2}\.\d{3}$')**
* Restrict an input to two digits and three decimals (while the decimals are optional) (e.g. **12** or **12.345**): **regex(., '^\d{2}(\.\d{3})?$')**

*Examples related to use of alphabets letters*

* Restrict an input of any characters (up to 6 characters long) consisting lowercase letters: **regex(., '^[a-z]{1,6}$')**
* Restrict an input of any characters (up to 10 characters long) consisting uppercase letters: **regex(., '^[A-Z]{1,10}$')**
* Restrict an input to only one of the three fruits outlined i.e. either to Apple or Orange or Banana: **regex(., '^(Apple|Orange|Banana)$')**
* Restrict an input to only two words i.e. pear or pair: **regex(., '^p(ea|ai)r$')**
* Restrict an input to only two words i.e. sun or son: **regex(., '^s[ou]n$')**
* Restrict an input with a valid email address: **regex(., '^[A-Za-z0-9.\_%+-]+@[A-Za-z0-9-]+[.][A-Za-z]{2,}$')**
* Restrict an input of the beneficiaries name where the initials of the first name and last name are uppercase e.g. Kobe Bryant: **regex(., '^[A-Z]{1}[a-z]{1,}[ ]{1}[A-Z]{1}[a-z]{1,}$')**
* Restrict an input of the beneficiaries name with first name, middle name (if any) and last name e.g. Kobe Bean Bryant: **regex(., '^\w{1,}\s(\w{1,})?(\s)?\w{1,}$')**
* Restrict an input of the beneficiaries' full name where the initials of the names are in uppercase and the name are quite long (often greater than 3 words) e.g. Samayamantri Venkata Rama Naga Butchi Anjaneya Satya Krishna Vijay *(this is an example of south Indian names)*: **regex(., '^([A-Z]{1}[a-z]{1,}\s)([A-Z]{1}[a-z]{1,}\s?)+$')**
* Restrict an input of the beneficiaries' first name (so that you are able to capture the exact spelling) where the enumerators are forced to enter the beneficiaries first name twice e.g. Kobe Bryant Kobe. *(This could be helpful when you are trying to document beneficiaries details where a typo error could cost you heavy)*: **regex(., '^(\D+)\s(\D+)\s?\1$')**
* Restrict an input of the beneficiaries' last name (so that you are able to capture the exact spelling) where the enumerators are forced to enter the beneficiaries last name twice e.g. Kobe Bryant Bryant. *(This could be helpful when you are trying to document beneficiaries details where a typo error could cost you heavy)*: **regex(., '^(\D+)\s(\D+)\s?\2$')**
* Restrict a character within a word by using the **?** (quantifier) e.g. allow either *color* or *colour* as an input: **regex(., '^colou?r$')**
* Restrict a character within a word by using the **\*** (quantifier) e.g. allow either ***a!*** or ***ah!*** or ***ahh!*** or ***ahhh!*** and so on as an input: **regex(., '^ah\*!$')**
* Restrict a character within a word by using the **+** (quantifier) e.g. allow either ***ah!*** or ***ahh!*** or ***ahhh!*** and so on as an input: **regex(., '^ah+!$')**
* Restrict an input to a non-digit character (e.g. **a** or **c** or **!** or **#** or **%** etc.): **regex(., '^\D$')**
* Restrict an input to five non-digit character (e.g. **aZcB!#%** etc.): **regex(., '^\D{5 }$')**

*Examples related to use of a combination of alphabets letters and numbers*

* Restrict one character which matches between a to z or A to Z or 0 to 9 or \_ (i.e. match one character from **[a-zA-Z0-9\_]**): **regex(., '^\w$')**
* Restrict three character which matches between a to z or A to Z or 0 to 9 or \_ (i.e. match one character from **[a-zA-Z0-9\_]**): **regex(., '^\w{3}$')**
* Restrict your beneficiary ID to a specific format e.g. **CAR\_PRC\_2020\_0048**: **regex(., '^[A-Z]{3}[\_][A-Z]{3}[\_][0-9]{4}[\_][0-9]{4}$')**
* Restrict your beneficiary ID to a specific format e.g. **CAR-PRC-2020-0048** *(where the enumerators should enter an exact match from* ***CAR*** *to - i.e.* **CAR-PRC-2020-** and can enter any 4 digit serial number): **regex(., '^CAR-PRC-2020-[0-9]{4}$')**
* Restrict a currency input of three digits with a currency sign (either dollar or pound) in front (e.g. **$999** or **£500**): **regex(., '^[\$|\£]\d{3}$')**
* Restrict an exact input of number of words (e.g. to restrict exactly 3 words *"I love you."*): **regex(., '^\W\*(\w+\b\W\*){3}$')**
* Restrict an input of number of words (e.g. to restrict a range of words say 3 to 5): **regex(., '^\W\*(\w+\b\W\*){3,5}$')**

**Considerations when using regex**

* If you wish to use a regex constraint on a number, make sure you ALWAYS type numbers under the appearance. This restricts the display of alphabets, making only numbers visible for inputs.
* Be careful, with the mode of data collection (i.e. KoBoCollect android app or Enketo web forms) you wish, once you have completed designing and deploying your survey forms (using regex). This is because, KoBoCollect android app and Enketo web forms behaves a bit differently with regex. KoBoCollect android app behaves as if you have used the anchors (^ $) around the regex (even if you have not used them), while Enketo requires the anchors as mandatory for an exact match.