VBA Regex Search

I like to start with as simple a search/code I can then build so here is a simple working RegEx VBA Sub that searches for 2 Letters then 8 Digits un-case-sensitive.

Note: Many guides say to enable a reference library, but IMO that makes the script non-portable so this does not require that

What you'll want to do is put an example string in cell A1, then test your strPattern. Once that's working adjust the range as desired.

```
Public Sub RegExSearch()
    Dim regexp As Object
    Dim rng As Range, rcell As Range
    Dim strInput As String, strPattern As String

    Set regexp = CreateObject("vbscript.regexp")
    Set rng = ActiveSheet.Range("A1:A1")

    For Each rcell In rng.Cells

        strPattern = "([a-z]{2})([0-9]{8})"

        With regexp
                .Global = False
                .MultiLine = False
                .ignoreCase = True
                .Pattern = strPattern
        End With

        If strPattern <> "" Then
            strInput = rcell.Value

            If regexp.test(strInput) Then
                MsgBox rcell & " Matched in Cell " & rcell.Address
            Else
                MsgBox "No Matches!"
            End If
        End If
    Next
End Sub
```

Metacharacter	Description
.	"Wildcard." The unescaped period matches any character, except a new line.
^	"Beginning of a string or line" or "negation." The significance of the caret in search patterns is context-dependent. 1) If the VBA `MultiLine` parameter is set to True, the caret means "beginning of line," rather than "beginning of string." 2) If the caret is placed within a character class [^], it acts as a **negation**, i.e. "match all characters NOT matched by the other parameters of the character class."
$	"End of a string." The significance of the dollar sign in search patterns is context-dependent. If the VBA `MultiLine` parameter is set to True, the dollar sign means "end of line," rather than "end of string."
\	"Escape." The backslash in front of a metacharacter turns it into a literal character.
\b	"Word boundary" or "backspace character." Outside character classes, \b matches a position before or after a word within the text source. Within character classes, \b denotes the backspace character.
\B	"Not a word boundary." \B is the negation of \b, but has no alternate meaning within character classes.
\d	"Digit." Matches any digit from 0-9.
\D	"Not digit." Matches any character that's not a digit.
\s	"Whitespace." Matches a space, newline or tab character.
\S	"Not whitespace." Matches a character that's not a space, newline or tab.

Operator	Meta-character	Description	Example
Boolean "or"			The vertical bar denotes the boolean "or" operator.	a|b matches either "a" or "b".

Grouping	()	Parentheses are used for several purposes: 1) to define the scope and precedence of operators.2) to group characters and remember text.	h(a|e)y matches either "hay" or "hey".
Quantificati on	?	Zero or one occurrences of the preceding element.	colou?r matches both "color" and "colour".
	*	Zero or more occurrences of the preceding element.	ab*c matches "ac", "abc", "abbc", "abbbc", and so on.
	+	One or more occurrences of the preceding element.	ab+c matches "abc", "abbc", "abbbc", and so on, but not "ac".
	{n}	The preceding item is matched exactly n times.	a{3} matches "aaa".
	{min,}	The preceding item is matched min or more times.	a{1,} matches "a", "aa", "aaa" and so on.
	{min, max}	The preceding item is matched at least min times, but not more than max times.	a{1,3} matches "a", "aa" and "aaa", but not "aaaa".

Optional Parameters	Type	Description
Global	Boolean	If set to True, all matches will be returned ("greedy matching"). If set to False, only the first match will be returned ("lazy matching"). The default value is False.
IgnoreCase	Boolean	If set to True, the pattern matching will be case-insensitive. The default value is False (case-sensitive).
MultiLine	Boolean	If set to True, it changes the interpretation of the **^** ("beginning of string") and **$**("end of string") meta-characters in the search pattern so that they match the beginning and end of a **line** instead. The default value is False.

Character classes

```
Character classes or character sets are specified with square brackets [ ]. Some of the most common ones are:

[a-z]
The set of lower-case letters ranging from a to z.

[A-Z]
The set of upper-case letters ranging from A to Z.

[0-9]
The set of single digits ranging from 0 to 9.
```