Regex

**g -> global search: find all occurences**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**Without g -> no global search, just pick the 1st occurence**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**i -> case insensitive**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**? -> optional if ist after a sign. That sign will be optional**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**+ -> one or more occurences**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

🡪this one gives 2 matches on line 2

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

🡪with e+, it gives 1 match on line 2

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

🡪the single ‘e’s in the hello arent matched because its ‘one or more’

**\* -> zero or more**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

🡪on the other hand if we do it with \*, the single ‘e’s are matched (zero or more)

**. -> matches anything, the sign does not matter**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**\ -> escape special signs, like .**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

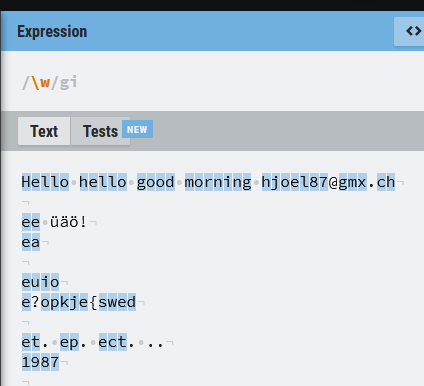
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

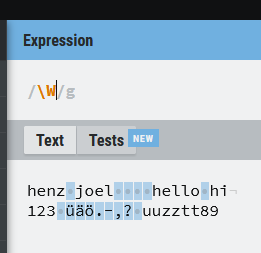
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**\w -> match any letter or digit (without special letters like üöä)**



**\W -> match anything but letters and digits**

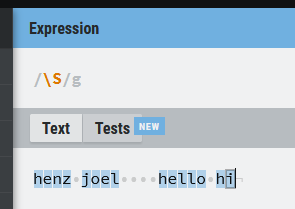


**\s -> match white spaces**

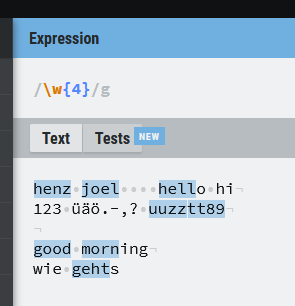
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

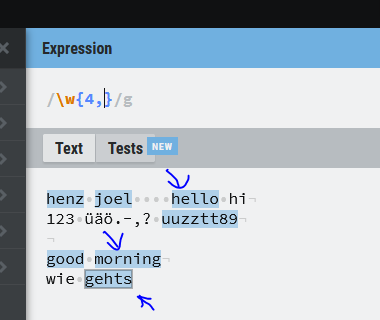
**\S -> match anything but whitespaces**



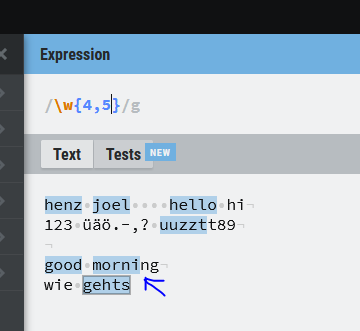
**\w{4} -> match words with length = 4**



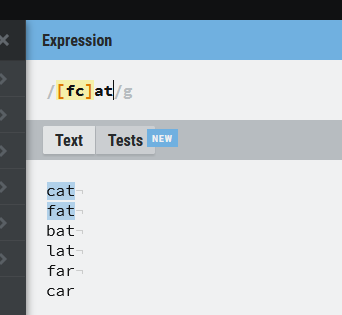
**\w{4,} -> match words with length = 4 or more**



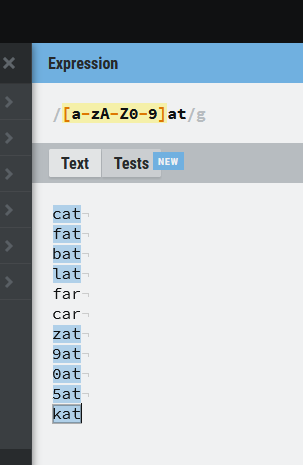
**\w{4,5} -> match words with length between 4 and 5**



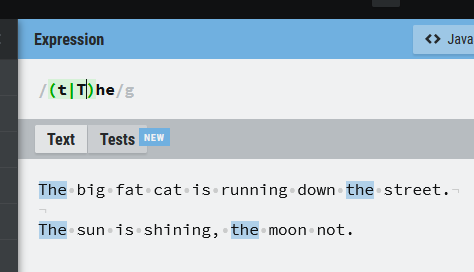
**[fc]at -> character sets: f and c for first char. This will match either if word starts with f OR c and ends with at**



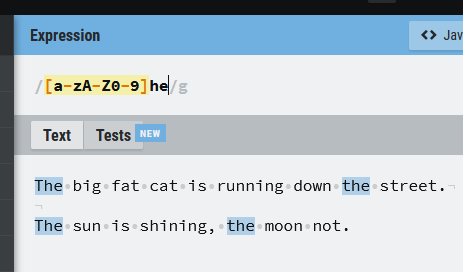
**[a-zA-Z0-9]at -> character sets with ranges: begin with any letter or digit**



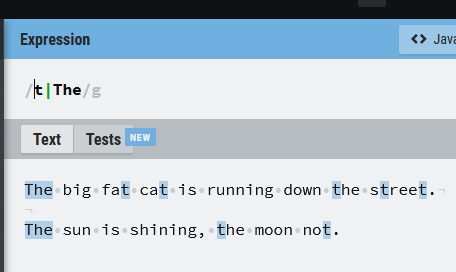
**(t|T)he ->capture grouping: match any with t or T at beginning**



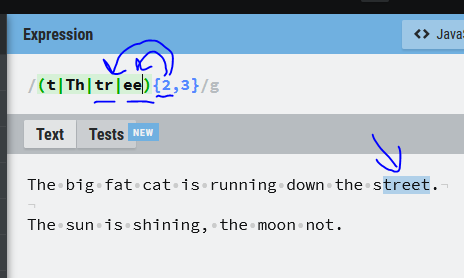
**🡪same we can do with [ ]**



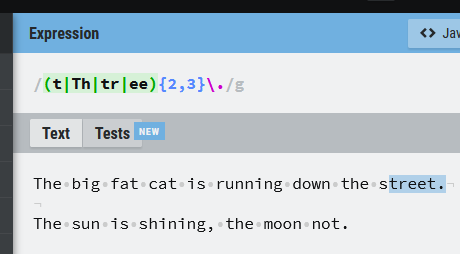
**🡪without grouping it would match either t or The**



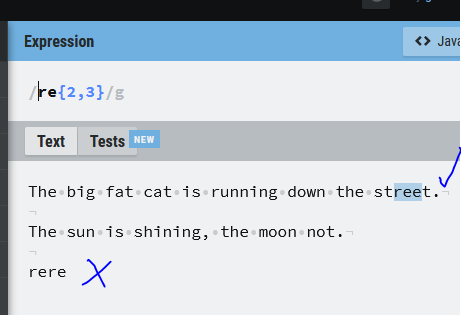
**🡪here we can also say ‚there is a match if 2 OR 3 occur from the group‘. E.g. we want a match if the word contains 2 or 3 of the following group : (t|Th|tr|ee)**



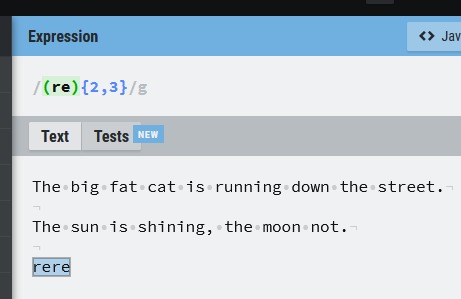
**🡪same, but match if there is a period after the word**



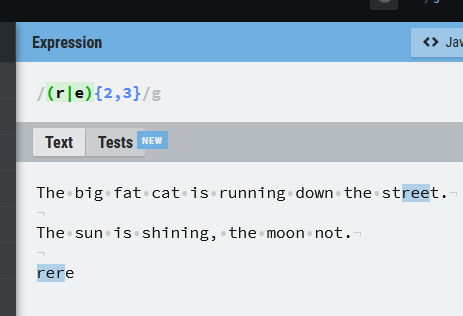
**🡪and again if we leave out the ( ), the quantifier only counts for the letter before (2 or 3 times e, with r at the beginning)**



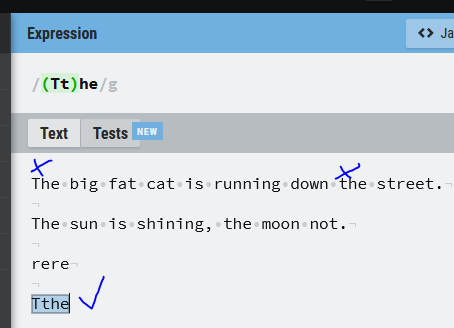
**🡪with ( ), we search a word with 2 or 3 times re**



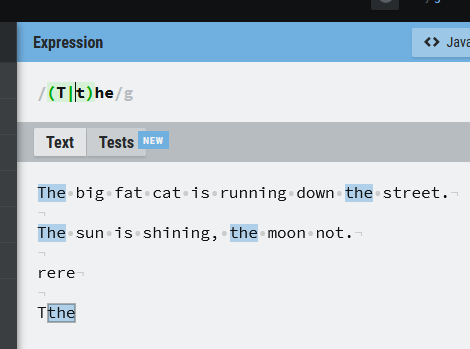
**🡪if we add a pipe, we search for 2 or 3 times of r OR e (rer is 2 times r + 1 time e = 3 times ; ree is 2 times e + 1 time r = 3 times)**



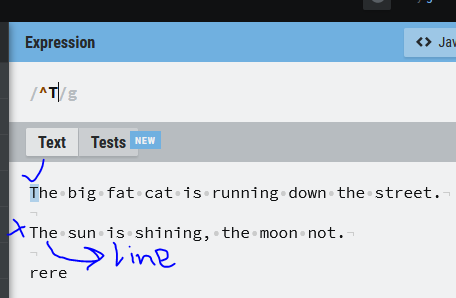
**🡪if we do not make a | , the logical operator AND is used**



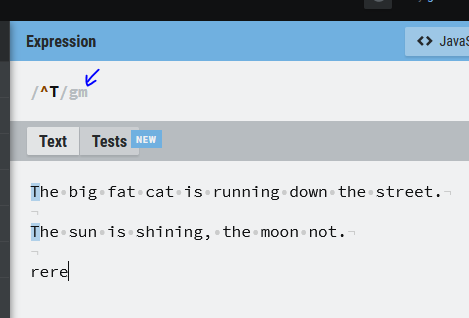
**🡪with |, ist OR operator**



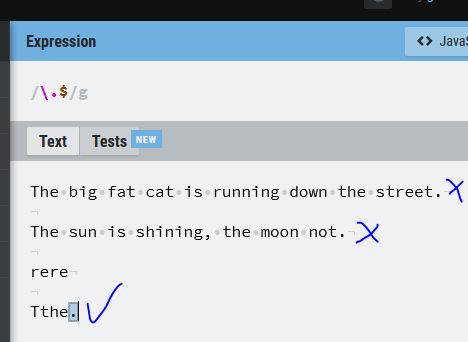
**^T->carrot: match the beginning of a whole text (not line!!!)**



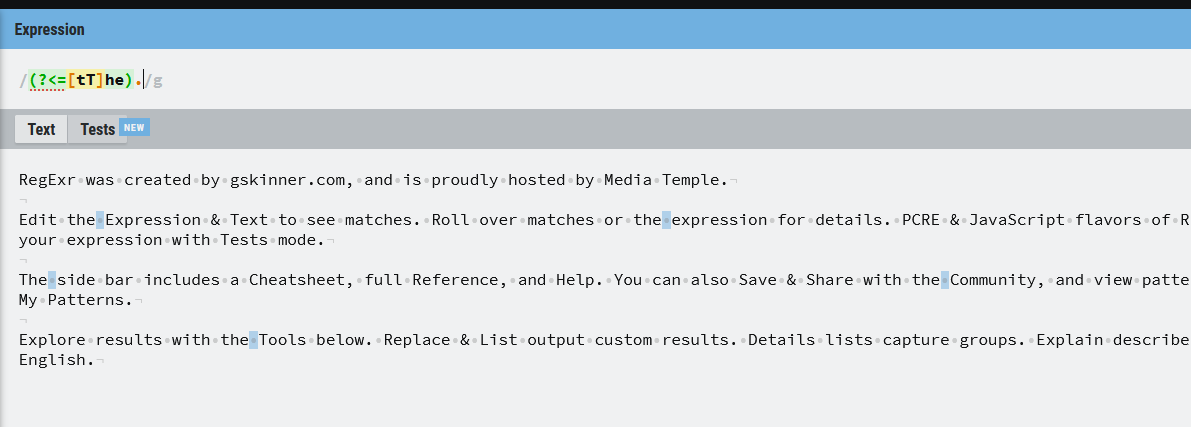
**🡪if we want it multiline, add an m**



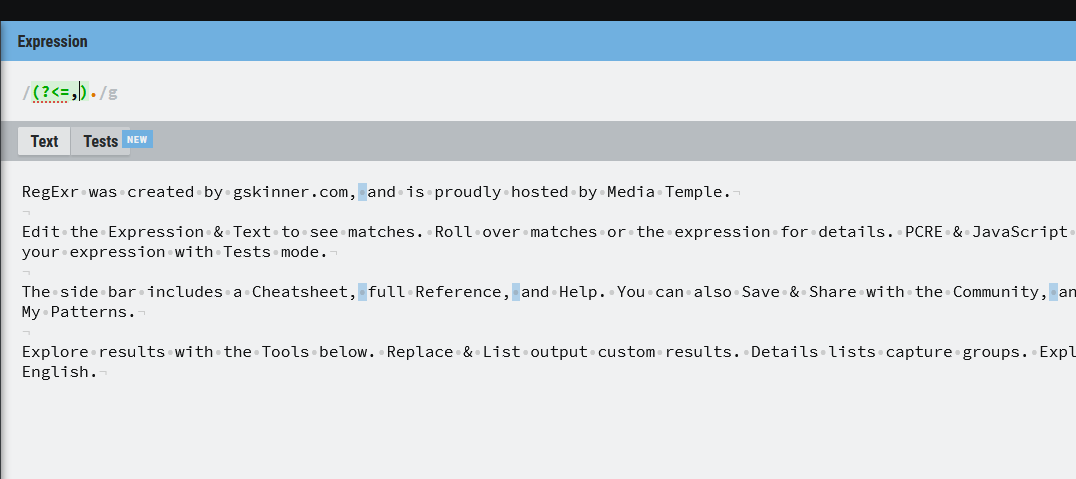
**\.$->match if this is at the end (of the text, not line!!!)**



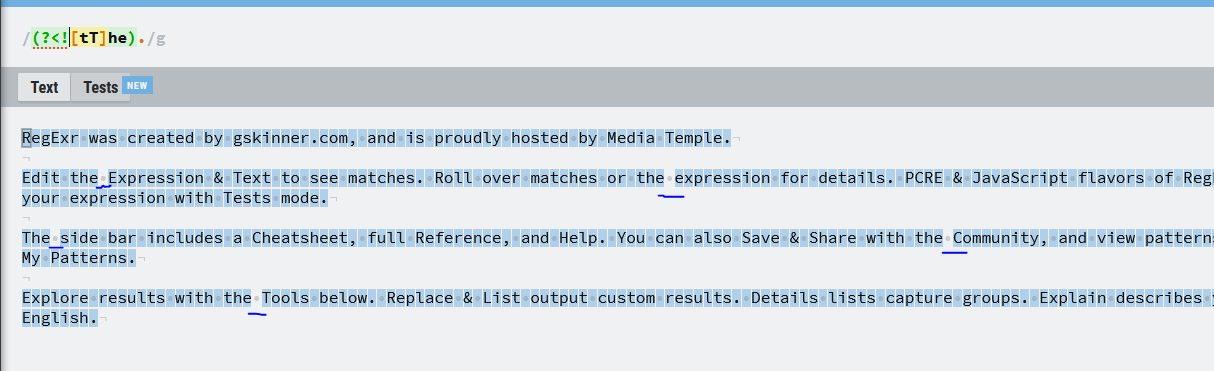
**(?<=[tT]he). ->positive lookbehind. Match anything (with . ) that follows after The or the**



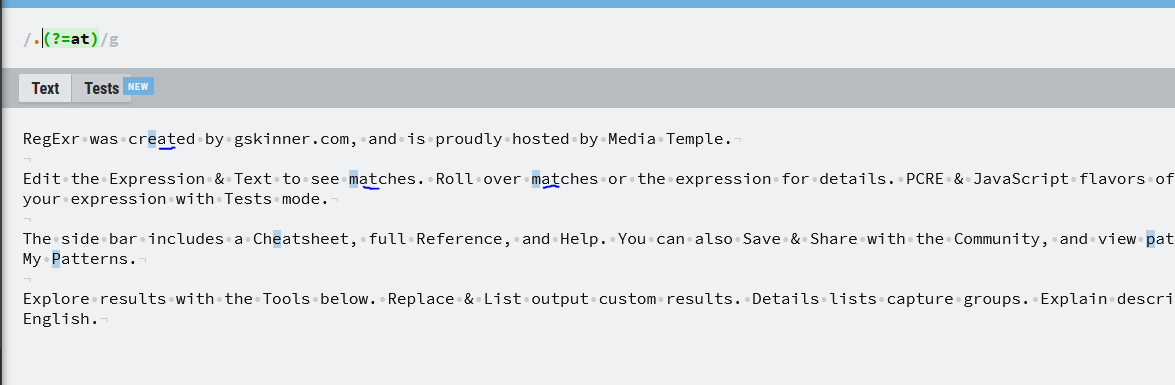
**🡪another example**



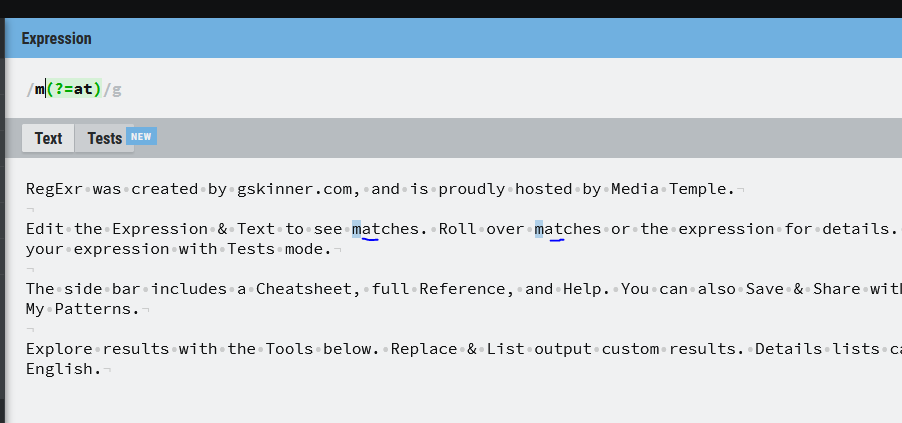
**(?<![tT]he). ->negative lookbehind. Match anything (with . ) that does not follow after The or the**



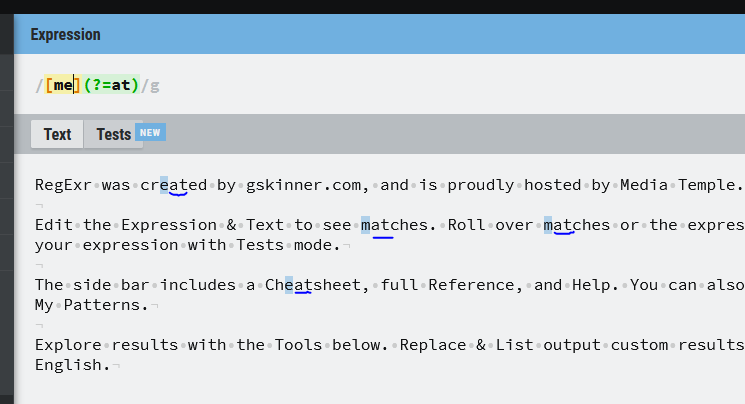
**.(?=at) ->positive look ahead. Match anything ( . ) that is before at**



**🡪same, but it must be a m**

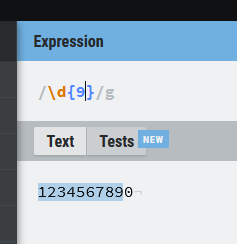


**🡪same, but it must be either e or m**

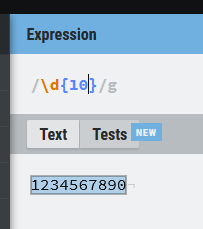


**Phone number excercise**

**🡪match 9 digits**



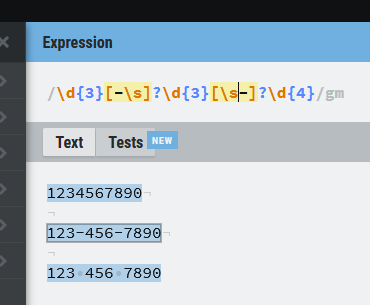
**🡪match 10 digits**



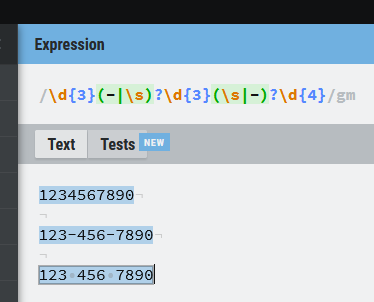
**🡪now add multi line and match numbers which CAN contain dashes. Thats why we use the ? for optional. So this will also match the first line**



**🡪we can also write it with spaces**



**🡪or do it with grouping**



**🡪now ist good practice to make groups also fort he digits**



**🡪you see that there are 5 groups. This could be helpful if you want to replace a whole group with something else. You can also give a group name tot he groups**

