Table of Contents

[2 References 4](#_Toc107587570)

[3 Modifiers 4](#_Toc107587571)

[3.1 d Flag 4](#_Toc107587572)

[3.2 g Flag 4](#_Toc107587573)

[3.3 i Flag 4](#_Toc107587574)

[3.4 m Flag 4](#_Toc107587575)

[3.5 s Flag 4](#_Toc107587576)

[3.6 u Flag 4](#_Toc107587577)

[3.7 y Flag 5](#_Toc107587578)

[4 Groups and Ranges 5](#_Toc107587579)

[4.1 OR Group - (x|y) 5](#_Toc107587580)

[4.2 Specify Range [abcd] 5](#_Toc107587581)

[4.3 Negated Range [^a-c] 5](#_Toc107587582)

[4.4 Capturing Group 5](#_Toc107587583)

[4.5 Capturing Group \nth 5](#_Toc107587584)

[5 Meta characters 5](#_Toc107587585)

[5.1 . 5](#_Toc107587586)

[5.2 \w 6](#_Toc107587587)

[5.3 \W 6](#_Toc107587588)

[5.4 \d 6](#_Toc107587589)

[5.5 \D 6](#_Toc107587590)

[5.6 \s 6](#_Toc107587591)

[5.7 \S 6](#_Toc107587592)

[5.8 \ b 6](#_Toc107587593)

[5.9 \B 7](#_Toc107587594)

[5.10 \0 7](#_Toc107587595)

[5.11 \n 7](#_Toc107587596)

[5.12 \f 7](#_Toc107587597)

[5.13 \r 7](#_Toc107587598)

[5.14 \t 7](#_Toc107587599)

[5.15 \v 7](#_Toc107587600)

[5.16 \xxx 7](#_Toc107587601)

[5.17 \xdd 7](#_Toc107587602)

[5.18 \udddd 7](#_Toc107587603)

[6 Quantifiers 8](#_Toc107587604)

[6.1 N+ 8](#_Toc107587605)

[6.1.1 N=john 8](#_Toc107587606)

[6.1.2 N=\d 8](#_Toc107587607)

[6.2 N\* 8](#_Toc107587608)

[6.2.1 digit + N=0 8](#_Toc107587609)

[6.2.2 N=0 8](#_Toc107587610)

[6.3 N? 9](#_Toc107587611)

[6.3.1 N=u 9](#_Toc107587612)

[6.4 N{X} 9](#_Toc107587613)

[6.4.1 N = u {X = 3} 9](#_Toc107587614)

[6.4.2 N = ou {X = 3} 10](#_Toc107587615)

[6.5 N {X, Y} 10](#_Toc107587616)

[6.5.1 N = ou {X = 3, Y= 6} 10](#_Toc107587617)

[6.6 N{X,} 10](#_Toc107587618)

[6.6.1 N = ou {X = 3, } 10](#_Toc107587619)

[6.7 N$ 10](#_Toc107587620)

[6.7.1 N$=is 11](#_Toc107587621)

[6.8 ^N 11](#_Toc107587622)

[6.8.1 ^N=is 11](#_Toc107587623)

[6.9 ?=N 11](#_Toc107587624)

[6.9.1 ?=N = all 11](#_Toc107587625)

[6.9.2 ?!N =all 12](#_Toc107587626)

# References

* [RegExr: Learn, Build, & Test RegEx](https://regexr.com/)

# Modifiers

## d Flag

Expression match should contain the start and end string

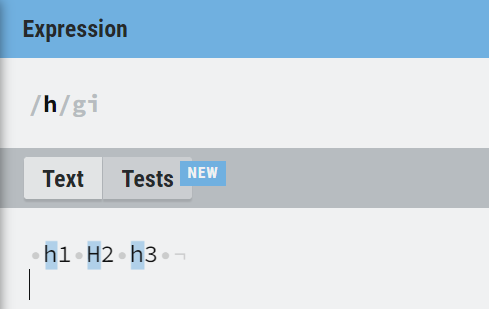
## g Flag

Expression should be tested against all possible matches in a string



## i Flag

Do a case-insensitive search



## m Flag

Multi-line search

## s Flag

Allows to match newline characters

## u Flag

Treat a pattern as a sequence of Unicode code points

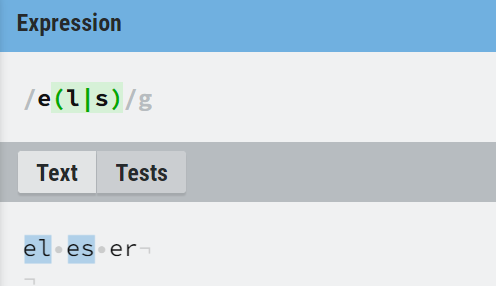
## y Flag

Perform a "sticky" search that matches starting at the current position in the target string

# Groups and Ranges

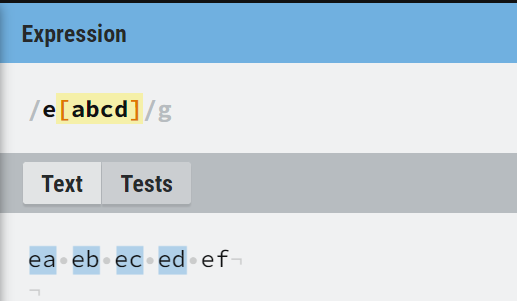
## OR Group - (x|y)

Matches either "x" or "y"



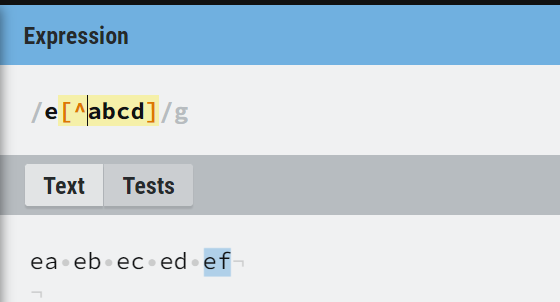
## Specify Range [abcd]

[abcd] is the same as [a-d]. They match the "b" in "brisket", and the "c" in "chop".



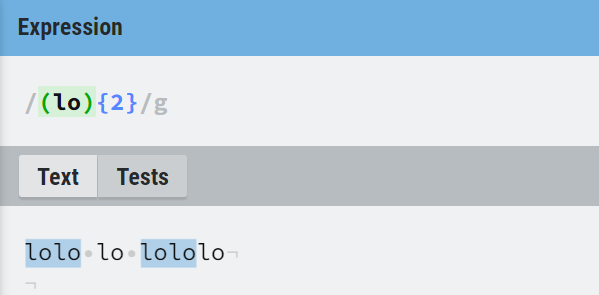
## Negated Range [^a-c]

[^a-c] They initially match "on" in "bacon" and "hop" in "chop" with global flag set



## Capturing Group

Matches x and remembers the match. (go)+ means go, gogo, gogogo and so on



## Capturing Group \nth

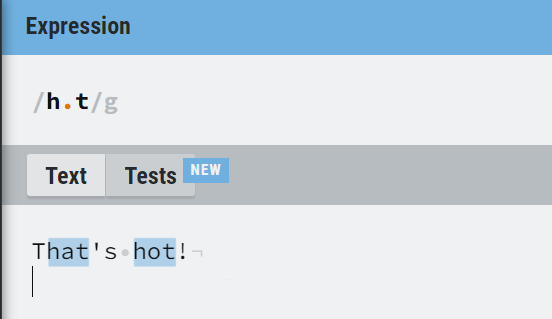
Where "n" is a positive integer. \1 refers to the first capturing group in the regular expression. \2 will refer to the second capturing group and \n will refer to an nth capturing group



# Meta characters

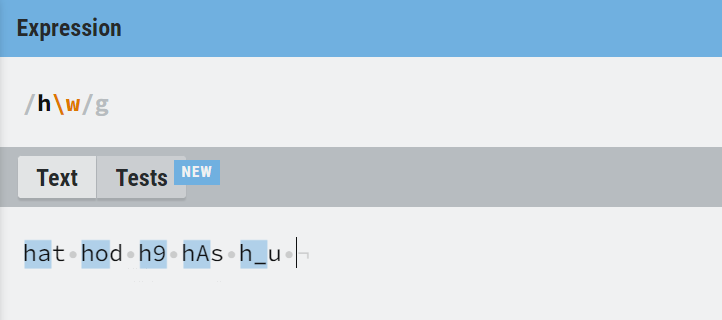
## .

Find a single character, except newline or line terminator



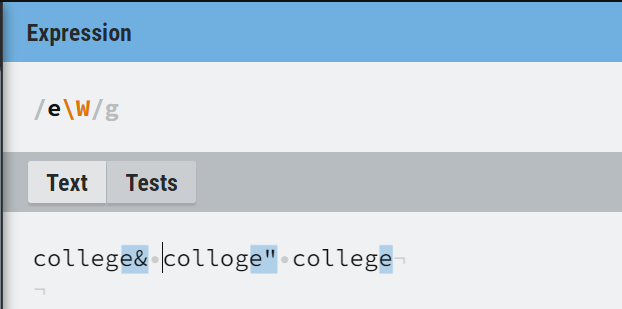
## \w

Find a word character. A word character is a character a-z, A-Z, 0-9, including \_ (underscore)



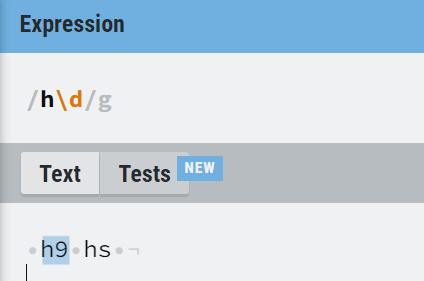
## \W

Find a non-word character



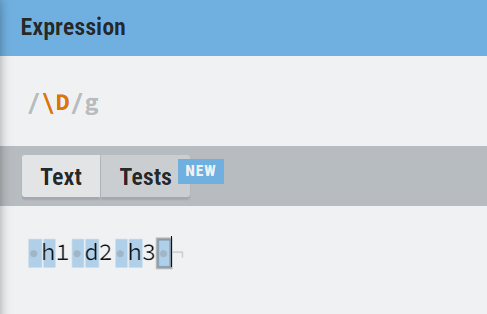
## \d

Find a digit



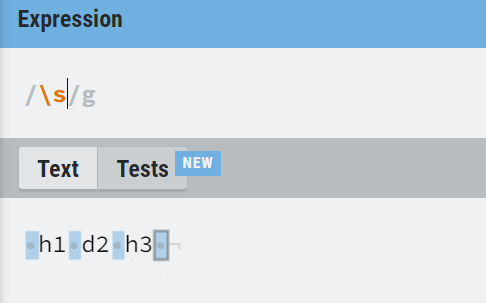
## \D

Find a non-digit character



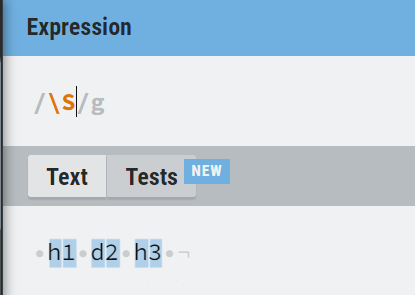
## \s

Find a whitespace character



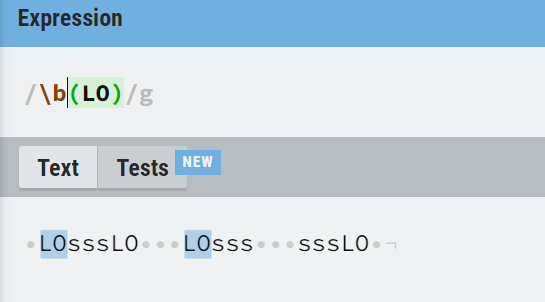
## \S

Find a non-whitespace character

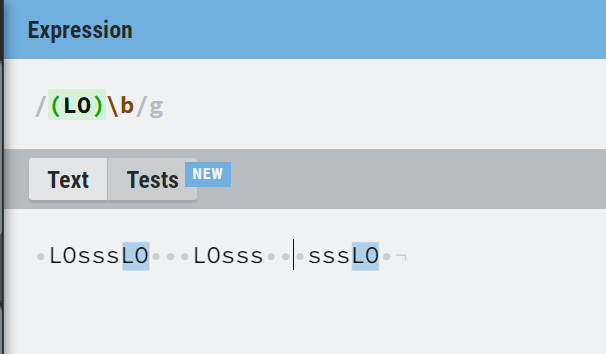


## \ b

Find a match at the beginning or end of a word



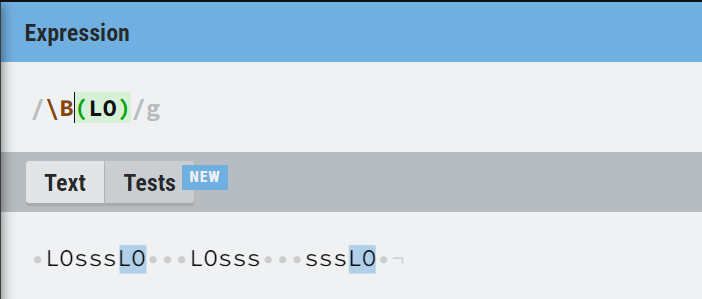
BEGINING of the WORD

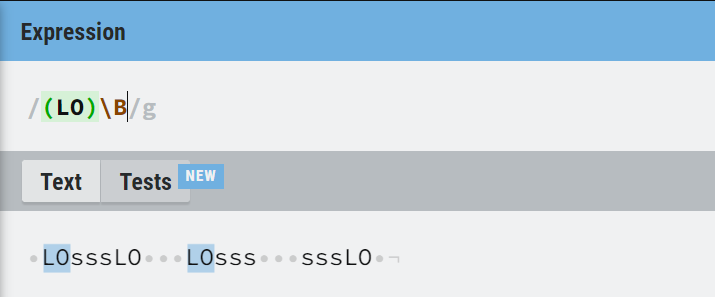


END of the WORD

## \B

Find a match, but not at the beginning or end of a word



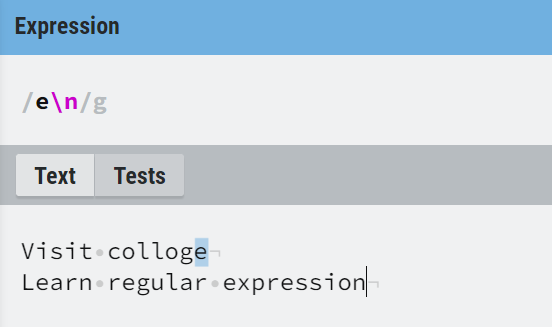


## \0

Find a NULL character

## **\n**

Find a new line character



## **\f**

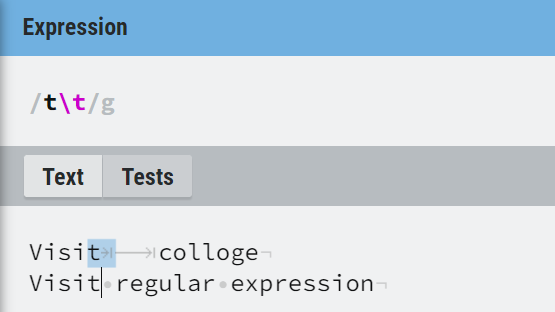
Find a form feed character

## **\r**

Find a carriage return character

## **\t**

Find a tab character



## **\v**

Find a vertical tab character

## **\xxx**

Find the character specified by an octal number xxx

## **\xdd**

Find the character specified by a hexadecimal number dd

## **\udddd**

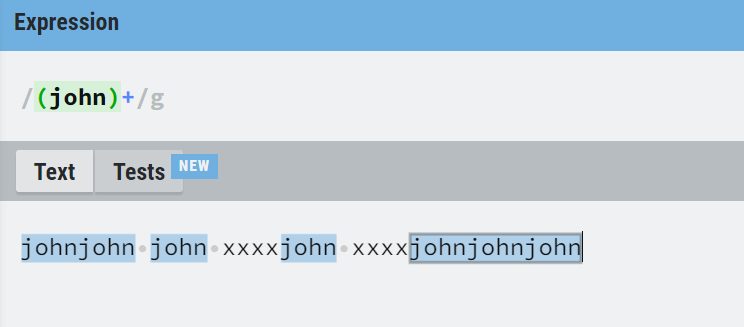
Find the Unicode character specified by a hexadecimal number dddd

# Quantifiers

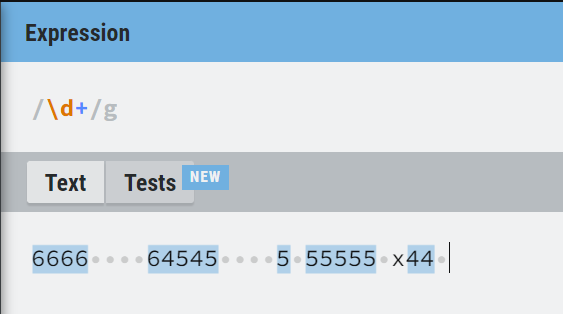
## N+

contains at **least one** N

### N=john



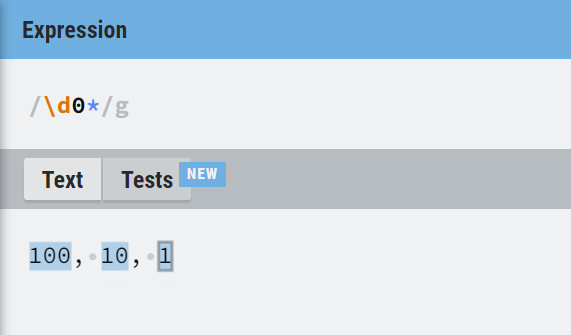
### N=\d



## N\*

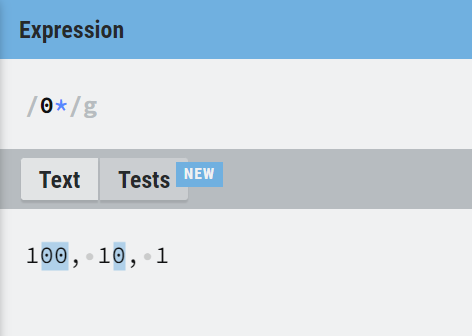
contains **zero or more occurrences** of N

### digit + N=0



### N=0

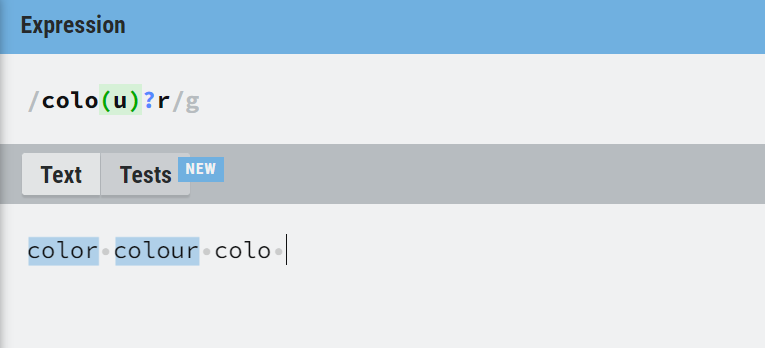
Need something before N for zero occurrences



## N?

contains **zero or one occurrences**

### N=u

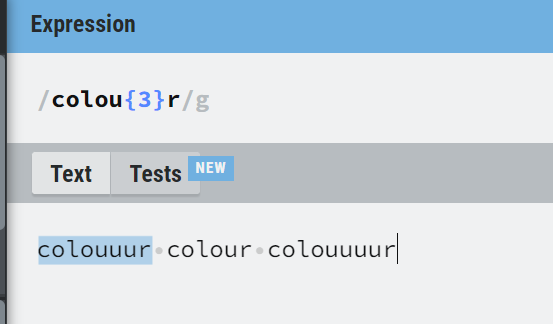


## N{X}

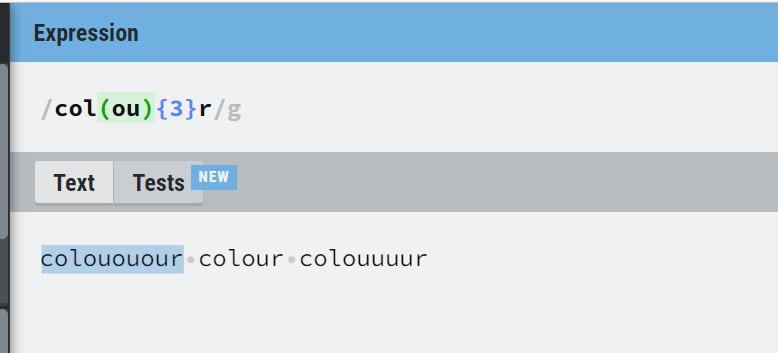
contains a **sequence of X** N's

### N = u {X = 3}





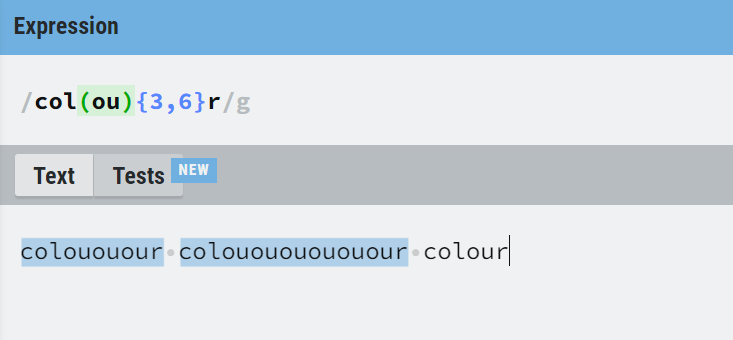
### N = ou {X = 3}



## N {X, Y}

contains a sequence **of X to Y** N's

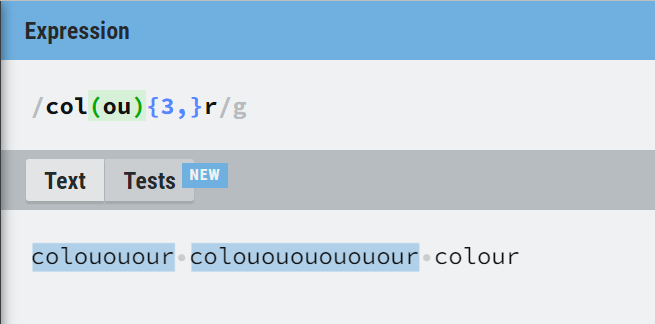
### N = ou {X = 3, Y= 6}



## N{X,}

contains a **sequence of at least X** N's

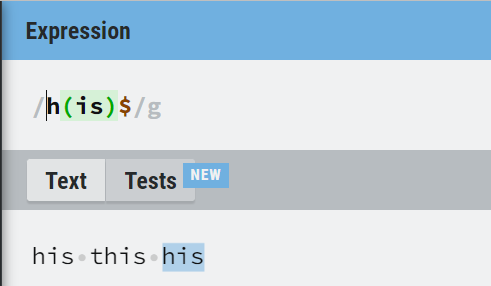
### N = ou {X = 3, }



## N$

string with N at the **end** of it

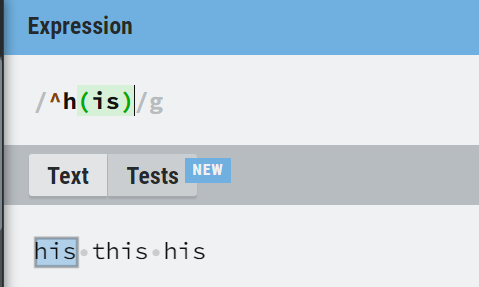
### N$=is



## ^N

string with N at the **beginning** of it

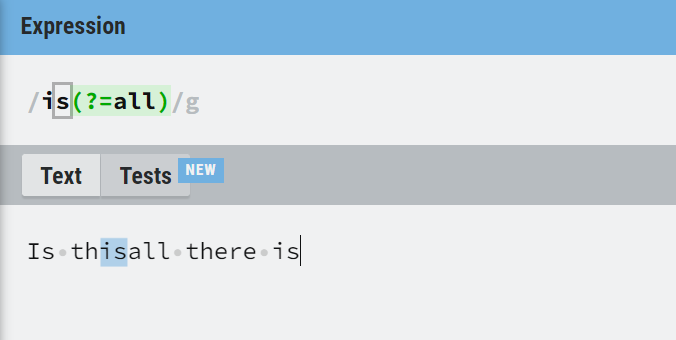
### ^N=is



## ?=N

string that is **followed by a specific string** N

### ?=N = all



### ?!N =all

