

Regex (Regular Expression):

Regex זאת בעצם דרך לפלטר טקסטים גדולים לפי תנאים\תבנית קבועים\ה מראש:

המבנה שלו יראה כך: `<flag>/<regex>`

Regex flags:

g – global – מאפשר לחפש יותר מהופעה אחת

לדוגמא: `/cat/g` – ימצא כל מקום בטקסט שמופיע בו המילה cat

`/cat/` – ימצא רק את הפעם הראשונה שהמילה cat מופיעה.

i – case insensitive – מבטל רגישות לאותיות גדולות וקטנות.

לדוגמא: `/cat/` – ימצא רק איפה שמופיע cat באותיות קטנות.

`/cat/i` – ימצא איפה מופיע cat לא משנה אם האותיות קטנות אן גדולות או גם וגם.

m – multiline – מתייחס לטקסט ככל שורה בנפרד ולא כטקסט אחד שלם.

לדוגמא: `/^A/gm` – ימצא כל שורה שמתחילה באות **A**.

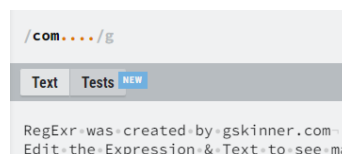
לעומת `/^A/g` – שימצא רק אם האות הראשונה של כל הטקסט היא **A**.

s – single line(dotall) – גורם לתו '.' להתאים גם לשורה חדשה (`/n`) מה שבלי הדגל לא קורה.

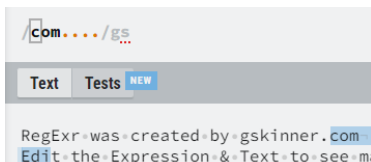
לדוגמא: אם יש בסוף שורה את המילה com ואני רוצה להשוות למה שאחריה אם אחפש:

`/com.../g` – הוא לא ימצא כלום בגלל שהתו . (שאמור להתאים לכל התווים) לא

מתאים לח/ (ירידת שורה):



אבל אם נוסיף את הדגל **s** הוא כן ימצא:



U – Unicode – מאפשר התאמה גם לתווי Unicode למשל עברית או שפות אחרות.

Special characters

+ - מאפשר לחפש הופעה 1 או יותר של התו שבא לפניו.

לדוגמא: `/e+/g` - ימצא את כל ההופעות של האות e (g flag is set) גם אם יש 1 או

יותר ברצף

? - 0 או 1 כלומר מאפשר לחפש תו אופציונלי אם הוא קיים יימצא אם לא יתעלם.

לדוגמא: `/ae?/g` - ימצא כל מקום שמופיע האות a ואם אחריה מופיעה e הוא יסמן גם אותה

ואם לא אז יתעלם.

* - סוג של שילוב של ? ו- + אומר בעצם על התו שבא לפניו שיימצא אם יש 0 או יותר.

לדוגמא: `/ae*/g` ימצא את כל הפעמים שהתו a מופיע ואם מופיע אחריו e הוא יסמן אותו

(גם אם יש כמה e ברצף כמו ה +) ואם לא יתעלם.

. - מוצא כל תו חוץ משורה חדשה.

לדוגמא: `/a..e/g` - יימצא כל מקום שמופיע a ואחריו שני תווים (לא משנה איזה חוץ משורה

חדשה) ואז e (למשל `axy`).

`/..a/g` - יימצא כל מקום שמופיע a ושני תווים לפניו (כולל רווחים לא כולל שורה

חדשה) למשל `xya`.

\ - מבטל את ההשפעה של התווים המיוחדים.

לדוגמא: אם נרצה לחפש את ההופעה של התו '!' ושלא תתפרש כ wildcard נחפש `/\!/g` -

יחפש כל מקום שמופיע **התו** '!' בטקסט. ואם היינו מחפשים `/!/g` בלי ה \ לפני ה .
הוא היה מוצא פשוט כל מקום שמופיע בו תו חוץ משורה חדשה.

\w - ימצא כל תו של מילה (כלומר אותיות)

לדוגמא: `/\w/g` - ימצא את כל האותיות שמופיעות בטקסט. ו `/\w*/g` ימצא את כל המילים

שמופיעות בטקסט (בגלל ש * מאפשר לחפש כמה ברצף).

\W - ההפוך מ \w - ימצא את כל מה שהוא לא אותיות בטקסט.

\s - ימצא כל הרווחים (whitespaces) בטקסט.

\S - ימצא את כל מה שהוא לא whitespace.

{<min>,<max>} - מאפשר לחפש מילים לפי גודל.

לדוגמא: `/\w{4,}/g` - ימצא את כל המילים עם 4 אותיות או יותר.

`/\w{4,6}/g` - ימצא את כל המילים שהגודל שלהם הוא בין 4 ל 6 (כולל).

[<character>]<characters> - מאפשר לחפש לפי קבוצה של תווים.

לדוגמא: `/[ab]at/g` - ימצא את כל המקומות שמופיע a או b ואחריו at למשל aat או bat.

<char>[<range>] - אפשר גם לבצע חיפוש לפי טווח .

לדוגמא: `/[a-z]at/g` - ימצא כל מקום שמופיע בו אות בין a ל z (באותיות קטנות כמובן)

ואחריה **at** . למשל **cat** .

$/[a-zA-Z]at/g$ – ימצא כל מקום שמופיעה אות גדולה או קטנה ואחריה **at** . למשל

Fat או **nat** .

() – מתייחס למה שבפנים כאובייקט נפרד.

לדוגמא: $/(\t|T)he/g$ – ימצא כל מקום שמופיעה המילה **the** או **The** ה | מסמן או אבל צריך

את הסוגריים בגלל שאם היינו מחפשים $/t|The/g$ – הוא יחפש איפה

מופיעה האות **t** או המילה **The** .

$/(a|b|c){2,4}/g$ – מאפשר לחפש גם לפי טווח כלומר יחפש איפה מופיע התווים

שבתוך הסוגריים בין פעמיים לארבע פעמים, למשל **abb** , **caba**

cab וכו'.

^ - מוצא התחלה או של כל הטקסט או של כל שורה (אם הדגל **m**).

לדוגמא: $/^A/g$ – ימצא רק אם הטקסט כולו מתחיל ב **A** .

$/^A/gm$ – ימצא כל שורה בטקסט שמתחילה באות **A** .

\$ - מוצא את הסוף של המחרוזת כולה או של שורה אם הדגל **m** .

לדוגמא: $/\$.g$ – ימצא אם הסוף של המחרוזת כולה נגמר ב ' .

$/\$.gm$ – ימצא כל שורה שמסתיימת ב ' .

:Look behind positive&negative

חיובי:

(?<=) – ימצא את מה שנמצא אחרי הביטוי שנרצה לחפש.

לדוגמא $/(?<=a)\.g$ - ימצא כל מקום שמופיע בו התו ' . ' אחרי האות 'a' .

שלילי:

(?!<) – בדיוק הפוך.

לדוגמא: $/(!a)\.g$ – ימצא כל מקום שמופיע בו ' . ' ואין לפניו את האות 'a' .

:Look ahead positive&negative

חיובי:

(?=) – ימצא את מה שנמצא לפני הביטוי שנרצה לחפש.

לדוגמא: $/\.(?=a)/g$ – ימצא כל מקום שמופיע בו התו ' . ' ויש אחריה את האות 'a' .

שלילי:

(?!=) – אותו דבר רק הפוך .

לדוגמא: $/\.(?!a)/g$ – ימצא כל מקום שמופיע בו התו ' . ' ואין אחריה את האות 'a' .

/b – מציין סוף ותחילה של מילה כלומר מאפשר להשוות מילה ספציפית ולא רק חלק (בודק שאחרי\לפני המילה אין שום תו או המשך אלא או פסיק או נקודה או רווח או התחלה או סיום של הטקסט).

לדוגמא: **/in/g** – ימצא כל מקום שמופיע בו in גם אם זה לא כל המילה למשל **interesting**.

/\bin/g – ימצא רק מילים שמתחילות ב in למשל **interesting** (את ה in השני הוא

לא יסמן מכיוון שהוא באמצע מילה ולא בהתחלה).

/\bin\b/g – ימצא כל מקום שמופיעה בו המילה **in** לבד.

דוגמא לשאילתה למציאת כל כתובות ה IPv4 התקינות בתוך טקסט (לא מכילות אוקטטות עם מעל 255 , מבנה נכון x.x.x.x וכו'):

/((?<!\.)\b((25[0-5] | 2[0-4][0-9] | (1[0-9]{2} | [1-9]?[0-9]))\.(3(25[0-5] | 2[0-4][0-9] | (1[0-9]{2} | [1-9]?[0-9]))\b(?:!\.)/g