 רגקס בוורד - מדריך

###### סקירה כללית

התוסף רגקס בוורד נועד כדי לשדרג את חוויית החיפוש בוורד. חיפוש רגקס דומה מבחינת מה לחיפוש עם תווים כלליים בוורד ברם לחיפוש רגקס ישנם יכולות משודרגות כמו מילים חילופיות ועוד ועוד. בהמשך המדריך ישנו מדריך מפורט על תווי רגקס ועל משמעותם.

התוסף מנגיש את חוויית החיפוש בוורד על ידי חלונית צד נוחה ועשירה, בין הפיצרים, רשימת תוצאות, יועץ רגקס, חיפוש לפי עיצוב, קיצורי מקשים לחיפוש והחלפה מגוונים, ועוד ועוד.

הממשק תומך בשפות זרות ומאתר באופן אוטומטי את השפה של וורד בין השפות הנתמכות – עברית, אנגלית, ספרדית, צרפתית, יידיש. אפשר להוסיף קובצי תרגום נוספים בקלות לתיקיית ה-"locale" של התוסף.

הממשק תומך באופן חלקי בשינוי צבע אופיס – כלומר התוסף מזהה את צבע האופיס הנוכחי (צבעוני שחור וכו') ומעתיק אותו ברם לאחר הפתיחה הראשונית הצבע נשאר סטטי כל עוד התוסף לא נפתח במסמך חדש – אז הצבע מתעדן מחדש עבור כל החלונות.

התוסף "שומר" את המיקום המועדף עליכם – כדי לשנות את מיקום חלונית התוסף גררו אותה למיקום המועדף עליכם. בפעם הבאה שהתוסף ייפתח הוא ייפתח במיקום זה.

###### סרגל הכלים



בסרגל הכלים ישנם שני לשוניות "חיפוש" ו-"החלפה" במצב חיפוש אפשרויות ההחלפה מוסתרות ובמצב ההחלפה רשימת התוצאות מוסתרת.

שלושת הלחצנים:

* **לחצן "אודות"** - לחצן זה פותח קישור לאתר הבית של התוסף אפשר להשאיר שם גם פניות תמיכה. תחת issues.
* **לחצן עזרה –** לחצן זה פןתח את המדריך של התוסף הכולל בתוכו גם מדריך על איך לערוך חיפוש רגקס.
* **לחצן "קידודון רגקס" -** הפותח רשימה פעילה של קודי רגקס מצויים ושימושיים עם הסברים ודגמאות קצרות.לחיצה על כל פריט בתפריט זה מזינה את אותו הפריט (קוד רגקס) אל תוך התיבה הפעילה הנוכחית חיפוש או החלפה.

###### לשונית חיפוש - לשונית זו נועדה עבור חיפושים מגוונים וגמישים.

תיבת החיפוש

****

בתוך תיבת החיפוש ישנם שלושה לחצנים "חיפושים אחרונים", "הפעל או כבה תווי רגקס" – לפעמים יותר נוח לכבות את החיפוש המשוכלל ולחפש את הטקסט כמות שהוא, "לחצן חיפוש".

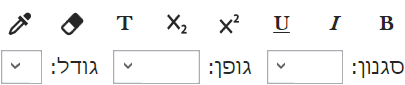
תוצאות החיפוש מוצגות ברשימה, רק כאשר לשונית החיפוש מסומנת, כאשר לשונית ההחלפה מסומנת התוצאות אינם מוצגות.

תיבת החיפוש משנה את כיוון הכתיבה באופן אוטומטי לפי שפת הכתיבה.

קיצורי מקשים עבור תיבת החיפוש:

* **מקש Esc –** ניקוי כל הטקסט
* **מקש אנטר –** חיפוש
* **מקשי חיצים למעלה או למטה** – ניווט בין התוצאות.

###### לחצני ותפריטי עיצוב



מודגש, נטוי, קו תחתי, כתב עילי, כתב תחתי, צבע טקסט, ניקוי כל העיצובים, העתקת עיצוב מתוך המסמך, סגנון, גופן, גודל גופן.

לחצנים אלו משמשים עבור חיפוש ממוקד ובתיבת ההחלפה עבור החלת עיצובים.

בלחצנים מודגש, נטוי, קו תחתי, כתב עילי, כתב תחתי, יש שלושה מצבים. מה שמאפשר חיפוש בשלילה כל טקטסט שאינו מודגש וכדומה.

###### הגדרות חיפוש



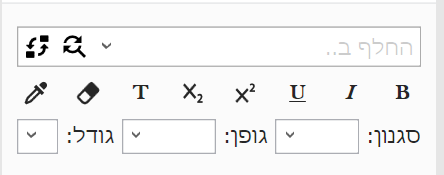
הגדרות חיפוש מאפשרות חיפושים ממקודים על כל המסמך, כלפי מעלה, וכלפי מטה, או לפי בחירה על הטקסט המסומן בלבד.

שימו לב. התוכנה מזהה כאשר טקסט מסומן במסמך ומשנה את אופי החיפוש לחיפוש לפי סימון באופן אוטומטי. כאשר הסימון משתנה ואין טקסט מסויים מסומן במסמך החיפוש חוזר למצבו הקודם. כמובן אפשר בכל זאת לשנות את אפשרויות החיפוש באופן ידני.

מרחק בין מילים – מאפשר חיפוש דמיש דוגמת בר אילן על ידי הגדרת מרחק מקסימלי בין מילים. ברירת המחדל הינה אפס.

זהירות! כאשר מרחק בין מילים מוגדר. תכונת ההחלפה מזהה את כל הרצף כרצף אחד ולא רק המילים המוזנות בתיבת החיפוש אלא גם את המלים שביניהם.

###### לשונית החלפה



לשונית זו נועדה עבור החלפות גורפות וגמישות על ידי תווי רגקס. ובלי. כמו בתיבת החיפוש ישנם אפשרויות עיצוב מגוונות כולל הסרת הדגשה. (לחצני מודגש, נטוי, קו תחתי, כתב עילי ותחתי, כולם כוללים שלושה מצבים: כלום, מודגש, לא מודגש וכדומה).

בתיבת ההחלפה ישנו תפריט ששומר את ההחלפות האחרונות כל עוד המסמך פתוח. לידו לחצן החלף וחפש, התנהגות הלחצן היא אם הטקסט המסומן נמצא שווה לערך החיפוש ההחלפה תוחל, בים כך ובין כך הוא פועל ]עולת חיפוש ומסמן את המופע הבא. הלחצן האחרון הינו לחצן "החלף הכל".

פעולת "החלף הכל" תלויה בתפריט אפשרויות החיפוש. כלפי מטה כלפי מעלה הכל או רק באיזור הסימון.

קיצורי מקשים עבור תיבת ההחלפה

* **מקש Esc** - ניקוי כל הטקסט
* **מקש אנטר -** החלפה, וחיפוש המופע הבא
* **חיצים למעלה ולמטה** - שינוי כיוון החיפוש אפשרות זו נועדה כדי לאפשר ניווט מהיר בין המופעים במסמך בשילוב לחצן האנטר.
* **מקש טאב –** מעבר בין תיבת החיפוש לתיבת ההחלפה וחוזר חלילה – מאוד שימושי כנ"ל במקרה שישנו צורך לנווט במסמך במהירות ולהחליף רק חלק מן המופעים.

[](https://tchumim.com/) **[תחומים - פורום חרדי מקצועי](http://tchumim.com/)**

# [מדריך](http://tchumim.com/) Regex - ביטויים רגולריים (מדריך חלקי, למדריך המלא היכנס לאתר)

[לכתבה המלאה לחץ כאן](https://tchumim.com/topic/1463/regex-%D7%91%D7%99%D7%98%D7%95%D7%99%D7%99%D7%9D-%D7%A8%D7%92%D7%95%D7%9C%D7%A8%D7%99%D7%99%D7%9D)

ביטוי רגולרי, שזה באנגלית Regular Expression זה בעצם פרוטוקול של שפה לתיאור טקסט רצוי (חיפוש, החלפה, או בדיקה האם טקסט עומד בחוקים), ע"י סימנים מיוחדים שמבטאים כל מיני משמעויות גמישות לתוצאה המתבקשת.

במגוון גדול מאוד של תוכנות וכלי תוכנה, יש מקום לregex או לתת וריאציה שלה (כמו מקרים מצומצמים של תמיכה בגוקר \*, ? שיש בשורת הפקודה בwindows או בוורד שיש סט רחב יותר תחת השם תווים כלליים, מי שמכיר את אינדיזיין פגש שם את גרפ (שהשם הוא לקוח מכלי נפוץ במערכות לינוקס לעבודה עם רג'קס), ניתן אפילו להצביע על תוכנות כמו פרויקט השו"ת של בר אילן ובני משפחתו שבעצם תומכים בחיפוש בשפה שהם יצרו לעצמם אבל היא בעצם עושה את אותו רעיון שזה תיאור של טקסט רצוי בצורה די מתמטית).

הרעיון הוא די פשוט, הסיטואציה: המתכנת כבר כתב קוד לחיפוש עבור המשתמש. אבל המשתמש עוד לא אמר למתכנת בעת כתיבת הקוד מה לחפש... אז כמובן המתכנת השאיר לו לבחור מה לחפש. הבעיה שהמשתמש לפעמים גם לא יודע מה הוא מחפש, אבל רק יש לו כללים מאוד ברורים לגבי התוצאה "אני רוצה מילה בת שלושה תוים שמתחילה בא", או "אני רוצה למצוא מספר טלפון". אז המתכנת הכין סט של אפשרויות טקסטואליות שמבטאות כללים פשוטים של מתחיל ב, נגמר ב, מורכב מטווח ו... ההמשך בregex. העובדה שבסופו של דבר כל חיפוש על טקסט מגיע בסוף לאותם אפשרויות אז יצרו שפה די מוסכמת (עם שינויים קטנים ספציפיים לכל מימוש), וממילא גם יצרו ספריות שגם "מבינות" וגם "מבצעות", כלומר המתכנת בדוגמה שלנו לא ימציא את הגלגל מחדש אלא סה"כ ישתמש עם כלי מוכן לאיתור טקסט בתוך טקסט ע"י שימוש בתווים הכלליים הללו.

טקסט תמיד מורכב מאוסף של תווים, וביטוי רגולרי בא בהגדרתו למצוא רצף של כאלה תואם למה שביקשנו.

הביטוי עשוי לכלול אותיות רגילות לגמרי. למשל ביטוי כזה 'א' ימצא את כל אותיות הא' בטקסט בו מחפשים. כמו"כ הביטוי 'אני' יחפש את המילה אני או ליתר הדיוק את הרצף של האותיות א' נ' י'. עד כאן זה בעצם התנהגות של חיפוש רגיל ומוכר.

אבל, עיקר הכח של הביטוי הרגולרי בא מתווים/סימנים מיוחדים. נסקור את העיקריים שבהם, לפי סוג.

###### ג'וקרים

נתחיל מהפשוט: אנחנו רוצים למצוא תו כל שהוא (נקודה), תו כל שהוא אבל רק מתוך קבוצת תוים (סוגריים מרובעות [XYZ]), או תו כל שהוא למעט קבוצה מוחרגת (סוגריים מרובעות עם חץ למעלה בהתחלה [^XYZ]). נקרא לזה ג'וקרים, שימו לב, כל הג'וקרים מייצגים תו בודד בלבד! (כל התוים המיוחדים יוצרים בעיה, מה קורה כשרוצים לחפש תו פשוט שלרוע מזלו הינו מיוחד מבחינת הregex? צריך לבטל את ייחודיותו ע"י escaping נדבר על זה בהמשך).

###### . נקודה - תו אחד כל שהוא

נקודה בביטוי רגולרי לא מייצגת את התו נקודה אלא היא סימן מיוחד, שמשמעותו "מתאים לי שתמצא איזה תו שתרצה". חיפוש של ביטוי רגולרי עם נקודה בלבד, יחזיר את כל התוים בטקסט (מס' התוצאות יהיה מס' התוים בטקסט הנתון), וזה לא הכי שימושי, אז נגיד דוגמה הגיונית יותר: חיפוש של 'א.' ימצא את כל אותיות הא' שבטקסט **עם** האות הסמוכה להם, תהיה אשר תהיה. כמו"כ חיפוש של 'א.י' יחזיר רצפים של א', תו כלשהוא, ואח"כ י', למשל אני, אלי, אוי (ואולי אפילו סיום של מילה בא' עם הרוח שאחריה ותו היוד של תחילת המילה הבאה).

###### [XXX] תו מקבוצה מוגדרת

תו אחד מקבוצה של תווים אפשריים. סוגריים מרובעות ובתוכם תווים, למשל [אהחער] מייצגת תו בודד ובלבד שהוא יהיה אחת מהאותיות (במקרה הזה, הגרוניות..) שבסוגריים. אם נחפש 'ב[אהחער]' זה יחפש אות ב' שאחריה ישנם אחת מהאותיות הללו, למשל בא, בה, בח.

[a-c] **טווח:** בסוגריים ניתן לכלול טווח תוים. טווח תוים זה אומר מתו עד תו, ללא צורך בהקלדתם בסוגריים. למשל במקום לכתוב [אבגד], ניתן לכתוב [א-ד]. מאיפה מנוע הרגקס יודע את סודות האלפא ביתא וסדרם? זה מתבסס על סדר התוים מבחינת המחשב. (וכמו השאלה בנקודה, אפשר לעשות גם טווחים וגם אותיות בודדות וגם כמה טווחים בסוגריים אחד, למשל [א-גלצ-ת] כולל את האותיות אבגלצקרשת (ומשמעותו 'מצא לי תו בודד שמופיע ברשימה זו').

[^XXX] **שלילה** בעוד שראינו שהסוגריים המרובעות מייצגות תו מתוך קבוצה מוגדרת שמופיעה בסוגריים. לפעמים ההגדרה הקצרה ביותר היא הפוכה, תו מתוך כלל התוים בעולם ובלבד שלא יהיה מאלו המוגדרות בסוגריים! בשביל זה עושים סוגריים שמייד אחרי הפתיחה שלהם מופיע חץ למעלה ^. למשל, [^א-ג] ימצא כל תו למעט א, ב, וג'.

**תוים מיוחדים** יש תוים שאי אפשר להקליד. למשל אם אתם מחפשים קפיצת שורה, או טאב, תכתבו תו מיוחד. למשל קפיצת שורה כותבים ע"י \n ואילו טאב \t יש עוד כמה כאלה. (רווח אפשר לכתוב כפי שהוא בביטוי החיפוש כמו כל אות רגילה).

**קיצורים** למעשה עם הסוגריים מרובעות אפשר לבטא כל תו שרוצים או שלא, אבל יש חיפושים כ"כ נפוצים שהחליטו לעשות להם קיצור לשיפור השמישות וגם הקריאות. למשל. מספר כל שהוא, ניתן לקבל ע"י [0987654321] או בקיצור ע"י טווח [0-9]. אז קבעו לזה קיצור \d. כלומר לוכסן (הפוך, מהסוג של וינדוס לשימוש בנתיב קבצים) ואחריו אות הd באנגלית, משמשים קיצור ל[0-9] כלומר הם מייצגים תו בודד מ0 עד 9 שניתן להגדיר זאת כמספר כל שהוא. או למשל דוגמא נוספת: לפעמים מחפשים תוי רווח. תוי רווח זה שם כללי לסימנים ה"שקופים" שמפרידים בין מילים - חוץ מהרווח המוכר מהמקלדת, יש את הקפיצת שורה וטאב ועוד כמה. כולם כלולים בקיצור \s. רשימת הקיצורים: \d מספר כל שהוא, שקול ל[0-9] \w תו של מילים, לאפוקי סימני פיסוק וגרוע מכך. קיצור של [a-zA-Z0-9\_] (לא יודע למה כללו את הקו התחתי) \s תו רווח כל שהוא, שקול ל[\r\n\t\f\v] (אל תשאלו אותי מה זה כל אחד...)

**קיצורי שלילה** אם תרצו כל תו שאיננו מספר, אתם תרצו לכתוב משהו כמו [^0-9]. אבל הרי יש לנו \d שמשמעותו [0-9], אז תוכלו לכתוב [^\d]. אבל עליכם לדעת שגם לזה עשו קיצורים... קיצורי השלילה זהים לקיצורים הקודמים רק אות גדולה במקום קטנה, כלומר:

\D כל תו שאיננו מספר \W כל תו שאיננו תו מילה - "נורמלי". \S כל תו שאיננו רווח קפיצת שורה טאב וכל החבורה.

###### חזרה וכמות ({}, +, \*, ?)

**כמה פעמים האלמנט (צריך/יכול) להופיע?** זו קבוצת סימנים, שההגדרה שלה היא "האלמנט (ברוב המקרים התו, לפעמים זה קבוצת תוים עטופה סוגריים) שאמרנו לך לחפש, תמצא אותו ? פעמים". נשמע מסובך... ככה: אם אנו מחפשים מספר כל שהוא, אנחנו כותבים \d או באריכות [0-9]. אבל זה מביא לנו ספרה בודדת. אם נרצה מספר בעל אורך מסויים למשל מס' חשבונית שאנו יודעים שתמיד מכילה 9 ספרות, נצטרך לכתוב ככה \d\d\d\d\d\d\d\d\d ובגריסה הארוכה [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]. יאהוו. בשביל זה יש מצייני כמות.

{X} או {Y,X} או {,Y} מX עד Y. כלומר האלמנט האחרון שהתבקש, צריך להיות לפחות X פעמים והכי הרבה Y פעמים. למשל, א{2,3} (הסדר הוא קודם המינימום, פסיק מקסימום) יחזיר מקרים בהם יש שתי אותיות א רצופות או שלוש (מקבוצה של ארבע הוא ייקח\יימצא\יתאים רק את השלושה תוים ראשונים). אם אכפת לנו רק המינימום, ואין לנו הגבלה עד כמה, נשמיט את הפסיק והמספר השני. אם להיפך, אנו רוצים גם מספר אחד, אבל לא יעלה על 9, נשמיט את הראשון אך נשאיר אתהפסיק (נכתוב תו לחיפוש, סוגר מסולסל פותח ופסיק, מס' מקסימלי סגירת סוגר מסולסל. האריכות בגלל שיבושי הכיוונים בעברית  ). שימו לב, כל מצייני הכמות מתייחסים לאלמנט הצמוד שקדם להם. הסימנים הם ?+\*{}

כוכבית \* 0 או יותר. האלמנט האחרון יכול להופיע 0 פעמים (אז הוא גם אופציונלי בעצם) אבל עד אין סוף. חיפוש .\* יחזיר פשוט את כל הטקסט כתוצאה אחת - שהרי פירוש הביטוי הוא תו כל שהוא כפול מספר לא ידוע של פעמים. עוד דוגמא א\d\* יחזיר כל רצף מספרי שקדם לה האות א' (בהמשך נלמד שכל החיפשוים ברגקס הם חמדניים -לוקחים כמה שיותר. ממילא ממספר של 4 ספרות לא תיהיה התאמה רק של 3 גם אם זה נכון מבחינת כללי החיפוש, כל עוד יכולה להיות התאמה גדולה יותר- של 4).

סימן הפלוס + אחד או יותר. אותו דבר כמו כוכבית, אבל בניגוד אליה, היא מחייבת שהאלמנט יופיע לכל הפחות פעם אחת. בעוד א\d\* ימצא גם א שאין אחריו מספר כלל, א\d+ ימצא רק א' שאחריו יש לפחות ספרה בודדת ואילך.

סימן שאלה ? אפס או אחד. זה בעצם אומר שהאלמנט האחרון הוא אופציונלי. אם נרצה לחפש בוקר, בכתיב מלא או חסר, נחפש בו?קר שמשמעותו ב, שאחרי זה תיתכן האות ו ותיתכן שלא ואחרי זה ק'ר'. או שליט"?א ימצא גם שליטא ללא ראשי תיבות. בר ?כל ימצא גם ברכל שהרי הרווח הוא אופציונלי. אם נחשוש שיהיו תוצאות כאלה בר-כל, בר\_כל, ניטיב לכתוב בר.?כל. זה ימצא כל תו, וגם בלעדיו כלומר ברכל רצוף.

הערה, יש מושג **כימות עצלן**: כל מצייני הכמות מנסים תמיד למצוא את המקסימום. למשל אם נחפש אלמנט HTML שזה בעצם טקסט מוקף זויות, אנו עשויים לחפש "<.+>" - משמעות: סוגריים זויתיות שביניהם 1 או יותר של תו כל שהוא. כעת אם יש כמה זוגות של כאלה בטקסט במקום למצוא כל אחד לעצמו הוא יתפוס מהראשון עד האחרון ברציפות! [הנה דוגמה](https://regex101.com/r/GKMy02/1). זה התנהגות הברירת מחדל שנקראת חמדנית. ההיפך של זה נקרא התנהגות עצלנית והיא מושגת ע"י הוספת סימן שאלה אחרי מציין הכמות - הכוכבית או הפלוס, הנה [דוגמא עובדת](https://regex101.com/r/TWxZiy/1). המשמעות היא להעדיף את ההתאמה המינימלית.

## [#3](https://tchumim.com/post/10371)

###### Escap - ביטול משמעות תו מיוחד

לפעמים אנחנו צריכים לחפש למשל נקודה או סוגריים וכדומה. והתוים הללו לא מתפרשים כמשמעם שהרי יש להם משמעות מיוחדת בregex. מה עושים? קוראים לזה escaping וזה מתבצע בregex ע"י לוכסן הפוך (ההוא של וינדוס בנתיבי הקבצים, ההפוך משל הכתובות אינטרנט) מקדים. למשל חיפוש . יחפש נקודה ולא תו כלשהוא. שימו לב, הregex בנוי בצורה מאוד עקבית, וממילא כל תו מיוחד שנמצא בקונטקסט חסר משמעות (מהמשמעות המיוחדת) הופך אוטומטית לתו רגיל ופטור מאסקייפ, זה חוסך המון! למשל - מקף, בתוך סוגריים מרובעות הוא מסמל טווח בין שתי תוים. אם נרצה לכלו את תו המקף, נשים אותו פשוט ראשון או אחרון (בכל מצייני הכמות, אפשר לשים אותם ראשונים). אותו דבר סוגריים מרובעות שמשמעותם קבוצת תווים אפשרית, די בכך שנבטל או נשמיט את משמעות הסוגר הפותח, שבן זוגו הסוגר יהפך לרגיל. וכן הלאה.

###### חלופות OR

אם אנחנו רוצים תו מבין שתיים אנחנו עושים ככה [אב] שזה אומר או א' או ב'. אבל אם אנחנו רוצים לתת שתיאפשרויות לרצף של יותר מתו בודד, למשל לחפש אור או מנורה, נכתוב עמוד כזה | בין האפשרויות, ככה: אור|מנורה. דוגמה מעשית יותר - https?|ftp זה יאתר או ftp או http (ויש s אופציונלית ע"י הסימן שאלה שאחריה). בשביל לקבץ אפשרויות יש להשתמש בסוגריים, אבל חובה קודם להכיר את המושג הבא, קבוצות.

###### קבוצות

קבוצות זה קיבוץ אלמנטים בחיפוש שלנו, המשמעות של הקיבוץ הזה היא בשביל התייחסות אל הקבוצה בהמשך ב1)ביטוי החיפוש עצמו, 2) תוצאה 3)ביטוי ההחלפה. אני אתייחס בעיקר ל3, שזה מה שקורה בהחלפה. אם אנו רוצים להחליף כל רצף מספרי במסמך למוקף סולמיות. אז אנו מחפשים \d+ אך במה מחליפים? אז יש שתי דרכים להתייחס לכלל התוצאה בביטוי ההחלפה. או $0, או \0. אז אנחנו נכתבו במקרה שלנו בהחלפה #$0#, [הנה דוגמה](https://regex101.com/r/lfvT8a/1). בינתיים לא היינו צריכים קבוצות (אם כי $0 משמעותו קבוצה אפס, וזה מתייחס תמיד לכלל הטקסט של התוצאה). אך כעת משימה סבוכה יותר, אנחנו צריכים למצוא מספרים מוקפים במרכאות ולעטוף את המספר שבתוכם בסולמיות. כלומר יש לנו "123" והרצוי כעת הוא "#123#" כמובן שהמספר לא תמיד 123... אז אנחנו מחפשים "\d+", ופה בהחלפה אנחנו צריכים מתכון כזה: מרכאות, סולמית, המספר שהיה בין המרכאות, סולמית מרכאות. פה יעזור לנו קיבוץ. נכתוב ככה: "(\d+)" ונחליף ככה: "#$1#". [דוגמה](https://regex101.com/r/9IxIXx/1). בזה שעשינו סוגריים מסביב ה\d+ הגדרנו שכל מה שעונה על הקירטריונים המוקפים מתוך התוצאה, יוגדרו כתת קבוצה בתוצאה. וכעת בהחלפה, יכלנו להתייחס לקבוצה הזו ע"י $1. מה זה ה$1? ובכן דולר בביטוי ההחלפה שמגיע מייד אחריו מספר, משמש כשומר מקום לקבוצה מהתוצאה (שהוגדרה בביטוי החיפוש). בגלל שבביטוי החיפוש יכולים להיות כמה קבוצות, אז המספר מציין את מספר הקבוצה לפי מיקומה (הראשונה 1 וכו'. 0 מתייחס לכלל הביטוי כולו).

אני צריך לפרוש, אני לא גמרתי על קבוצות, אמשיך מחר אי"ה.

## [#4](https://tchumim.com/post/10388)

לפעמים נקבץ אלמנטים לא בשביל להפוך אותם לקבוצה, אלא כדי להחיל עליהם כמות, כמו {}, כוכבית, פלוס, סימן שאלה, או | (המקרה הכי נפוץ זה סימן | שנותן שתי אופציות אבל רק בחלק מהביטוי או סימן שאלה שהופך קבוצת אלמנטים שלמה לאופציונלית). במקרה כזה, ייתכן שזה יפריע לנו בכלל העובדה שזה הופך לקבוצה (למה? לא יודע, אולי בגלל שיש הרבה קבוצות בביטוי שלנו וצריך כבר לאחוז ראש. אם כי הגיוני בכזה מצב לעשות קבוצות שמיות שמייד אסביר). בשביל לקבץ אלמנטים לא על מנת להפכם לקבוצה אלא כדי להכיל עליהם אופרטורי כמות או |, ניתן לכתוב מייד בתחילת הסוגריים ?:, זה נקרא non-capturing-group, [הנה דוגמה](https://regex101.com/r/3DW45n/1) של קיבוץ (הירוק מורה על קבוצה), [והנה](https://goo.gl/JrpQKg) אחרי נטרול של הקבוצה.

###### קבוצות עם שם

(מממ, לא קיים בJS, אתכם הסליחה (גם לא בES8 למיטב ידיעתי, חבל. עריכה: זה מגיע ב[ES2018](https://github.com/tc39/proposal-regexp-named-groups)). אבל נמצא כמדומני בכל הפלטפורמות האחרות). לעיתים, זה לא נח לעבוד עם קבוצות לפי אינדקס. גם צריך לאחוז ראש את הסדר, גם במקרה שהביטוי מטופל בקוד הוא הופך קשה לתחזוקה כי כל שינוי בביטוי מחייב היערכות בהתייחסות, וגם לפעמים כל הסדר משתנה מתוצאה לאחרת כי אחת הקבוצות נמצאת במקטע אופציונלי. בשביל זה יש אופציה לתת שמות לקבוצות. זה נורא פשוט, בתחילת הסוגריים שמים סימן שאלה ואחריו סוגריים זויתיות שבתוכם שם הקבוצה - "מצוות (סוכה|לולב|שופר)" נכתוב "מצוות (?<מצוה>סוכה|לולב|שופר)" ואח"כ נוכל להתייחס לקבוצה הזאת באופן שמי, למשל בהחלפה במקום $1, נכתוב ${מצוה}. [הנה דוגמה](https://regex101.com/r/RYMD7T/1).

###### סימני עגינה ^-$

אם נרצה לחפש מילה שלמה, למשל סוכה ולא בסוכה, ולא מהסוכה, אז נקדים לה רווח. אלא שאז נאבד את המקרה בה היא מופיעה מייד בתחילת הטקסט. נוכל לציין שלפניה יש או תו רווח או את תחילת הטקסט ע"י סימן העגינה של תחילת הטקסט - ^. נכתוב ככה (^| )סוכה. אותו דבר לגבי הסיום, נכול לדרוש שיש אחריו רווח או סוף הטקסט ע"י הסימן $, (^| )סוכה( |$). שכיח בתכנות שרוצים לבדוק את כלל המחרוזת ולא למצוא רק התאמה בתוכה, ואז כותבים משהו כמו ^סוכה$ - משמעות הביטוי היא התאמה לכלל טקסט הבדיקה, שהרי נכללים בו אין ההתחלה והן סיום הטקסט. כמו"כ בשביל לבדוק מתחיל ב ונגמר ב גם כותבים את אחד מתווי העגינה הללו. למשל באינפוט של טלפון בhtml נוכל לדרוש התאמה לרגקס ^05\d-\d{7}$. נ.ב. יש סימן מיוחד שמסמן סוף\תחילת מילה, והוא \b. הוא בעצם קיצור ל: או רווח או קצה הטקסט (התחלה או סיום) או פסיק וכו'. יש לדבר הזה בעיה קטנה, זה לא עובד בשום שפה חוץ מאנגלית...

פעם הבאה זה על הדגלים - regex flags

## [#5](https://tchumim.com/post/14002) [#6](https://tchumim.com/post/46722)

קישורים למדריכים עבריים אחרים ברשת: <http://www.guides.co.il/download.php?guide=161> [http://webdesk.co.il/ביטויים-רגולריים-regular-expressions-הקדמה/](http://webdesk.co.il/%D7%91%D7%99%D7%98%D7%95%D7%99%D7%99%D7%9D-%D7%A8%D7%92%D7%95%D7%9C%D7%A8%D7%99%D7%99%D7%9D-regular-expressions-%D7%94%D7%A7%D7%93%D7%9E%D7%94/)

פה יש דוגמאות מעשיות <https://reshetech.co.il/regex-in-javascript-tutorial> [https://shushan.co.il/הסבר-ולימוד-regex-regular-expression](https://shushan.co.il/%D7%94%D7%A1%D7%91%D7%A8-%D7%95%D7%9C%D7%99%D7%9E%D7%95%D7%93-regex-regular-expression)