

• תחומים - פורום חרדי מקצועי

• ביטויים רגולריים - Regex

- 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 אינדיזין 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 תווים 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.
 - מי שמכיר את שפת JS, שמה ממש קל לבדוק ולשחק עם regex שקיבל מקום מכובד בשפה, ואגב הפעולה שלו בשפה זו מהירה במיוחד (יותר מדוטנט למשל).
 - יש הרבה כלים לבדקה ונסיון של regex, חלקם אונליין, אני למשל משתמש על פי רוב באתר הזה: <https://regex101.com>. בתיבת הטקסט העליונה כותבים את המחרוזת לחיפוש, ובחלק שתחתיה כותבים טקסט כל שהוא שעליו יתבצע החיפוש. בצד יש הסבר שלאט לאט יובן, שמנתח את הטקסט שהוזן בחיפוש. העברית לא הכי נתמכת טוב, עדף לצרכי טסטים לכתוב נטו באנגלית (ולעברת רק בסוף ב"הלכה למעשה").
- אתר נוסף שלחלק מהאנשים עשוי להיות ידידותי יותר הוא <http://regexstorm.net/tester>, עוד אתר הוא <https://www.regexpal.com>, יש כמובן עוד עם הבדלי טעם וריח.

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 regex? 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 שתמצא". 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 תוים אפשריים. סוגריים מרובעות ובתוכם תוים, 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 אבגלצקרת (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, ג.

- 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 ה d 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 בסוגריים) שאמרנו לך לחפש, תמצא אותו ? פעמים". נשמע מסובך... ככה: אם אנו מחפשים מספר כל שהוא, אנחנו כותבים $d^$ או באריות $[0-9]$. אבל זה מביא לנו ספרה בודדת. אם נרצה מספר בעל אורך מסויים למשל מס' חשבונית שאנו יודעים שתמיד מכילה 9 ספרות, נצטרך לכתוב ככה $d\d\d\d\d\d\d\d\d\d$ ובגריסה הארוכה $[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]$. יאהו.
- בשביל זה יש מציני כמות.
- $\{X,Y\}$ או $\{Y,X\}$ מ X עד Y . כלומר האלמנט האחרון שהתבקש, צריך להיות לפחות X פעמים והכי הרבה Y פעמים. למשל, $\{2,3\}$ (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. האריות בגלל שיבושי הכיוונים בעברית).
- שימו לב, כל מציני הכמות מתייחסים לאלמנט הצמוד שקדם להם. הסימנים הם $*+?^$ כוכבית $*$ 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 d^* יחזיר כל רצף מספרי שקדם לה האות א' (בהמשך נלמד שכל החיפשוים ברגס 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 d^* ימצא גם א שאין אחריו מספר כלל, 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, בכתוב 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 מציני הכמות מנסים תמיד למצוא את המקסימום. למשל אם נחפש אלמנט HTML שזה בעצם טקסט מוקף זווית, אנו עשויים לחפש $<.+>$ - משמעות: סוגריים זוויתיות 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 דוגמא](#). 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](#). המשמעות היא להעדיף את ההתאמה המינימלית.

- טוב, אני כבר חרגתי, ההמשך יבוא אי"ה ויהיה על קבוצות.
- [מנטור אישי למתכנתים \(ולא רק\) – להתקדם לשלב הבא!](#)
- בכל נושא אפשר ליצור קשר dovid@tchumim.com

• 14

• 

• [dovid](#) 

• dovidכתב בנערך לאחרונה על ידי

• ביטול משמעות תו מיוחד - Escap

- 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 לוסן הפוך (ההוא של וינדוס בנתיבי הקבצים, ההפוך משל הכתובות אינטרנט) מקדים. למשל חיפוש. יחפש נקודה ולא תו כלשהוא. שימו לב, regex בנוי בצורה מאוד עקבית, וממילא כל תו מיוחד שנמצא בקונטקסט חסר משמעות (מהמשמעות המיוחדת) הופך אוטומטית לתו רגיל ופטור מאסקינג, זה חוסך המון! למשל - מקף, בתוך סוגריים מרובעות הוא מסמל טווח בין שתי תוים. אם נרצה לכלול את תו המקף, נשים אותו פשוט ראשון או אחרון (בכל מציין הכמות, אפשר לשים אותם ראשונים). אותו דבר סוגריים מרובעות שמשמעותם קבוצת תוים אפשרית, די בכך שנבטל או נשמיט את משמעות הסוגר הפותח, שכן זוגו הסוגר יהפך לרגיל. וכן הלאה.

• OR חלופות -

- 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](#). 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](#). בזה שעשינו סוגריים מסביב 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, אמשך מחר אי"ה.

- [מנטור אישי למתכנתים \(ולא רק\) – להתקדם לשלב הבא!](#)

- בכל נושא אפשר ליצור קשר dovid@tchumim.com

- 11

- 

- [dovid](#) 

- dovidכתב בנערך לאחרונה על ידי

- 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 שמיידי 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](#) של קיבוץ (הירוק מורה על קבוצה), [והנה](#) אחרי נטרול של הקבוצה.

• קבוצות עם שם

- (ממ, לא קיים בJS, אתכם הסליחה (גם לא בES8 למיטב ידיעתי, חבל. עריכה: זה מגיע בES2018). 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 זוויתיות 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](#).

• 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, נוכל 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 html נוכל לדרוש התאמה לרגקס ^05\d{7}\$.

ג.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

- [מנטור אישי למתכנתים \(ולא רק\) – להתקדם לשלב הבא!](#)

- בכל נושא אפשר ליצור קשר dovid@tchumim.com

- 11

- 

- [dovid](#) 

- dovidכתב בנערך לאחרונה על ידי

• Regex Flags

- בRegex יש נושאים "פתוחים", ובדרך כלל השפות שמריצות Regex מאפשרות לקבוע "דגלים" כדי להחליט על האופן הרצוי (זה קרוי באנגלית בשמות flags או modifiers).

למשל, הגיוני שירצו שהRegex יתעלם מאותיות קטנות או גדולות, כלומר במקום לכתוב כל פעם את האותיות בשתי אופציות (כשזה a-z זה קל, אפשר לכתוב A-Za-z ואפילו פשוט A-z אבל כשזה מילים מסוימות זה מתחיל להיות קשה עד בלתי אפשרי, כי כל אות צריכה שתי אופציות), אנו נרצה שמבחינת הRegex אין הבדל כלל בין אות קטנה לאות גדולה. בשביל זה יש דגל בשם i, שהוא ראשית המילה insensitive = לא רגיש.

ניתן להחיל (חלק מה)דגלים רק על חלק מהביטוי (זה נקרא inline), זאת ע"י סימן שאלה שמיידי אחריו באים הדגלים, נקודתיים, והחלק עליו יחולו הדגלים. למשל:

- Regex (?:javascript)

- זה ימצא javascript לא משנה עם אותיות קטנות גדולות או ערבוב שלהם, אבל יחייב שאח"כ יש Regex שהכל קטן והוא Rn גדול.

- (ניתן מאידך להעביר דגלים על כל הביטוי כולו - שלא כחלק מהביטוי, צורת העברת הדגל למנגנון הRegex משתנה ממימוש שפה לאחרת. בJS בליטרל ניתן לכתוב מייד אחרי הביטוי (אחר הסלש את האותיות שמייצגות דגלים) בהמשך בע"ה כשאגיע למימוש בJS אראה דוגמה). הדגלים הנפוצים הם:

g - פירוש, global והכוונה, שאנו רוצים יותר מתוצאה אחת. לולי הדגל הזה, בשפות בהם הוא נתמך, חזרת התוצאה הראשונה המתאימה בלבד, לחיסכון בביצועים. הg זה דגל שכמובן לא שייך על חלק מהביטוי.

m - מרובה שורות - multiline. זה משנה את התנהגות ה\$^ במקום לתחילת וסוף הטקסט כולו לתחילת וסוף כל שורה ושורה בטקסט.

i - אי רגישות לאותיות קטנות\גדולות, נתמך גם על חלק מהביטוי.

s - זה סיפור רציני. הפירוש של זה הוא single line, כלומר "ראה את כל הטקסט כיחידה, אחת, כאילו הוא בשורה אחת". לא, אין הכוונה לשנות את התפקוד של \n או של \$. אלא זה בא לשנות התנהגות ברירת מחדל משונה בRegex שהנקודה (שהיא הגוקר מספר אחד בRegex), אף שכוללת כל תו ביקום, לא כוללת קפיצת שורה! וזה אומר שחיפוש *. יחזיר שורות שורות במקום תוצאה אחת ארוכה. והחיפוש של A.*B לא יחזיר תוצאה אם ביניהם יש קפיצת שורה. הדגל הזה מתקן את זה למצב ההגיוני בו נקודה שווה כל תו בעולם כולל קפיצת שורה.

U (גדול!) - פירוש: Ungreedy - לא חמדן. זה דגל שהופך את התנהגות ה+ וה* וה{}

ל"עצלנים", לוקחים כמה שפחות (זה שקול לסימן שאלה אחרי מציין הכמות שפועל בדומה לזה).

- (אגב, אחד המימושים המהירים ביותר של Regex הוא כנראה [php של pcre](#), ויש לו גם יותר דגלים שמראים שהRegex קיבל יחס ממש חם במימוש הזה).
- בפעם הבאה אני מקווה להתחיל להראות איך לעבוד ספציפית בJS.
- [מנטור אישי למתכנתים \(ולא רק\) – להתקדם לשלב הבא!](#)
- בכל נושא אפשר ליצור קשר dovid@tchumim.com

• תגובה 1

• 10

• מנותק

• [dovid](#)

• כתב בנערך לאחרונה על ידי

• קישורים למדריכים עבריים אחרים ברשת:

<http://www.guides.co.il/download.php?guide=161>

<http://webdesk.co.il/ביטויים-רגולריים-regular-expressions-הקדמה/>

• [מנטור אישי למתכנתים \(ולא רק\) – להתקדם לשלב הבא!](#)

• בכל נושא אפשר ליצור קשר dovid@tchumim.com

• 7

• מנותק

• ב.ל

• בנערך לאחרונה על ידי ב.ל [dovid](#) השיב ל

• [dovid@](#) אמר בRegex - ביטויים רגולריים:

• בפעם הבאה אני מקווה להתחיל להראות איך לעבוד ספציפית בJS.

• פה יש דוגמאות מעשיות

<https://reshetech.co.il/regex-in-javascript-tutorial>

<https://shushan.co.il/הסבר-ולימוד-regex-regular-expression>

• ליצירת קשר - baruchlamdan@gmail.com

• 0

• ב.ל.התייחס לנושא זה ב

• [התייחס לנושא זה בchagold](#)

• [התייחס לנושא זה בchagold](#)

• [התייחס לנושא זה בpcinfofmach](#)

● בא תתחבר לדף

היומי!

- [רוצה לזכור קריאת שמע בזמן? לחץ כאן!](#)

-
-