ביטוי רגולארי או באנגלית: Regular Expression

כאשר רוצים לחפש על נושא מסוים (למשל בגוגל), כותבים ׳בערך׳ מה שרוצים ואז מנוע החיפוש מביא לנו המון קישורים שיש בהם את מה שחיפשנו בצורות שונות ומגוונות.

אם איננו יודעים בדיוק מה אנחנו מחפשים אבל יודעים שזהו מספר טלפון. האם יש דרך לעשות זאת?

יש לנו קובץ באורך 50,000 דפים ויש בו מספרי טלפון של שלוש חנויות שמוכרות טלפון סלולארי בחצי מחיר. בדיפדוף רגיל יקח לנו בערך 7 שעות למצוא את הראשון. איך אפשר לזרז את החיפוש?

מסתבר שישנה דרך תכנותית לעשות זאת. אפשר ליצור תבנית שניראת כמו מספר טלפון ולבקש חיפוש של תבנית זו.
בפייתון אפשר לייבא את המודול שנקרא re לשם כך.

אם נרצה לחפש תבנית כלשהי בתוך טקסט בצורה מפורשת (כקבוע), לדוגמה:

```
re.match(r'From\s+', 'From amk Thu May 14 19:12:10 1998‘)
```

מנסה לחפש את המחרוזת משמאל בהתחלת המחרוזת מימין. (\s זה סימן לסוף מילה - נדבר על זה בהמשך).

כדי להשתמש בתבנית של re כמשתנה, עלינו לבצע מה שנקרא קומפילציה של הביטוי

```
pattern = re.compile(‘abc’)
text = ‘Ha HaHa’
matches = re.findall(pattern, text)
matches = re.finditer(pattern, text)
```

match - מחזירה או אובייקט או None. בודק רק מהתחלת המחרוזת. (לא שימושי)
search מחזירה אובייקט מסוג match (עבור המופע הראשון משמאל שתואם).
findall מחזיר רשימה
finditer מחזיר אובייקט שהוא איטרטור ומתוכו אפשר לקבל אינפורמציה. למשל את הפוזיציות של מה שהתאים על ידי המתודה:( )span את התת מחרוזות התואמת על ידי ()group

תוים שזקוקים ל ׳הגנה׳ בתוך ביטוי רגולרי (מיוחדים) הם:

( ) | \ [ ] { } ? + * $ ^ .
ההגנה נחוצה על מנת שיאבדו את תפקידם המיוחד והיא מורכבת מציון התו \ (back slash) לפני התו המיוחד - ניראה זאת בהמשך.

. התו נקודה תואם כל תו מלבד תו שורה חדשה.
\d - כל ספרה
\D - כל דבר שאינו סיפרה.
\w - אותיות קטנות או גדולות או סיפרה או קו תחתון (_ 0-9 A-Z a-z)
\W כל דבר שאינו \w
\s כל תו שהוא ניראה כרווח (גם התו של שורה חדשה \n וגם tab או \t
\S כל מה שאינו \s

<u>התוים המיוחדים הבאים משמשים כעזר לגידור תוים אחרים.</u>

דוגמה: example1

\b נועד כדי לגדר מילה (word boundary). למשל אם נתונה המחרוזת: 'Ha HaHa'

pattern = '\bHa'

מדוע קיבלנו רק שתי התאמות ולא שלוש?

התו \b מציין גבול מילה לפני המילה, כלומר גם התחלת המחרוזת וגם תו הרווח הם כאלה. ה- Ha השלישי אינו כזה.

\B כל מה שאינו בגידור של מילה. זה למשל יתן לנו את ה- Ha השלישי בלבד.

דוגמה: example2
^ - נועד לגדר משהו בתחילת המחרוזת
$ - נועד לגדר בסוף המחרוזת.

דוגמה: **example3**

חיפוש תבניות כלליות

נניח שאנחנו רוצים לחפש משהו שנראה כמו מספרי טלפון מהסוג: 054-8381185
או מהסוג: 050.4040398 (כלומר מפריד של נקודה או מקף בין קידומת ומיספר)

ננסה להשתמש במה שמציין סיפרה: \d

למשל: \d\d\d.\d\d\d\d\d\d\d שימו לב ששמנו את התו המיוחד 'נקודה' בין השלוש הראשונות והשבע הבאות (זה נועד להכיל את המקף או הנקודה, כיוון שנקודה משמע כל תו כמעט)

מחלקות תוים
אנו עלולים לקבל יותר מידי. כדי לציין שרק מקף או נקודה רצויים נוכל להשתמש במה שניקרא: מחלקת תוים (charcter set class) שמצויינת על ידי סוגריים מרובעים.

שימו לב שהסוגריים המרובעים מציינים רק תו אחד והבחירה היא ׳או׳ בין התוים שבתוכם.

עוד דוגמה
אפשר כך לבדוק תקינות של מספר טלפון. עבור סלולרי, הקידומת היא 05

בתוך מחלקת תוים התו ׳מקף׳ מקבל משמעות מיוחדת. הוא נועד לומר ממשהו ועד משהו. כך אפשר לשפר עוד יותר את הדיוק של בדיקת הטלפון. בואו נניח שכל הסלולריים בעלי קידומות 050, 051, 052, 053, 054, 055

ניתן לסיפרה הימנית ׳לרוץ מ-0 ועד 5)

באותה מידה אפשר להשתמש בטווח אותיות:
[a-z] [A-G] [a-zA-Z]

שלילת רישיון לתו
example4
גם התו ^ (caret) מקבל משמעות מיוחדת בתוך מחלקת תוים. פירושו שלילת התוים, כלומר כל מה שהוא לא התוים שבאים אחריו. למשל [abc^] פירושו כל מה שאינו a, אינו b ואינו c.

למשל זיהוי הזמנות. חייבת להתחיל בסיפרה 7, 8 או 9, וחייבת להיות בעלת 4 ספרות.

מצייני כמויות

+, *, ? {1,3} {4}

+ אחד או יותר
* אפס או יותר
? אפס או אחד
{n} מספר פעמים שבסוגריים
{n,m} כמספר בין n ל- m

אפשר למשל לקצר את הבדיקה של מספר הטלפון כמו בדוגמה הראשונה של **example5.**

נעבור לדוגמה קצת יותר חכמה. נניח שרוצים להיות מסוגלים לתפוס כותרת של מכתב שמופנית לאדם והאפשרויות הן:

Mr. Jones
Mr Eduards
Ms Smith
Mr. T
Mrs. Robinson

קודם ננסה לתפוס את .Mr, Mr והשם משפחה.

לאחר מכן ננסה גם את ה - Ms וה- .Mrs
לשם כך נלמד מושג נוסף שנקרא **'קבוצה'** והיא מוגדרת בין סוגריים רגילות. בתוך הסוגריים אפשר להשתמש בסימן 'או' |

אחרי שהצלחנו לתפוס את כל הכותרות (עוד נחזור לנושא של קבוצות ביתר פירוט).

כתובות email

בואו ננסה לכתוב ביטוי שיתפוס את 3 הכתובות הבאות:

js3104@nyu.edu freddy.Krooger@yahoo.net aaron@gmai.com

example6

ישנם באינטרנט הרבה ביטויים שמנסים להכיל כמה שיותר אפשרויות עבור כתובת mail ולקרוא אחד כזה יותר קשה מאשר לכתוב בדרך כלל.

דוגמה אינטרנטית עבור כתובת mail

r'[a-zA-Z0-9-.]+@[a-zA-Z0-9]+\.[a-zA-Z0-9-.]+'
שימו לב שלא צריך הגנה על הנקודה בתוך הסוגריים המרובעים, אבל צריך מחוצה להם.

חלוקה לקבוצות בעזרת סוגריים
example7
בדוגמה הבאה ברצוננו לתפוס כתובות אינטרנט מכל מיני סוגים, למשל:

https://www.google.com
http://yahoo.com
https://youtube.com
https://www.nasa.gov

רואים שלעיתים יש http ולעיתים https בתור פרוטוקול.
נעשה את ה-s אופציונלי. בשם הדוממין לעיתים יש www ולעיתים לא. גם רמת הדומיין הגבוהה לעיתים com. ולעיתים gov.

ניכתוב ביטוי שלא רק יבצע בדיקת התאמה, אלא גם ׳יתפוס׳ קבוצות של סוגריים עגולים. לשם הבנת התוצאה נשתמש במתודה group של אובייקט match.
קבוצת האפס היא תמיד כל מה שתואם לביטוי כולו.

```
pattern = re.compile('https?://(www\.)?(\w+)(\.\w+)')
```

מלבד קבוצת האפס יש לנו כאן את קבוצת הרמה הנמוכה של הדומיין, שם הדומיין וקבוצה עבור הרמה הגבוהה של הדומיין.
כאשר משתמשים בקבוצות, השימוש ב - findall יותר מורכב, כי הוא מחזיר את כל הקבוצות שתאמו.

example8
יחד עם הקבוצות ניתן גם להשתמש בשיטה שניקראת ׳התיחסות לאחור׳ (back reference) שמאפשרת לנו באותה נשימה של יצירת הקבוצות, גם להשתמש בהן.
ניראה זאת בעזרת דוגמה שגם משלבת החלפה של המחרוזות שתאמו בקבוצה או קבוצות שהתקבלו. הקבוצות מצויינות על פי סדר יצירתן ניתנות להתייחסות כ: \1 \2, \3 וכדומה.

התנהגות חמדנית - greedy

ננית שהביטוי הרגולרי הוא:

```
r'".+"'
```

והמחרוזת הניבדקת היא:

```
r'English "Hello", Spanish "Hola"'
```

סביר לצפות שהביטוי יתאים ל- "Hello" ול- "Hola"
אבל בפועל הוא יתאים ל: "Hello", Spanish "Hola"

הסיבה היא שהאופרטור + הוא חמדן ומנסה ׳לאכול׳ כמה שיותר תוים ולכן הוא לא עוצר בגרשיים הראשונים או השניים , אלא ממשיך עד האחרונים.

נוכל לשנות את התנהגותו על ידי תוספת של סימן שאלה:
r’”.+?”’
שינוי זה יגרום לו לבצע את מה שחשבנו תחילה, התאמה של “Hello” ושל-
“Hola”

דגלים
זהו פרמטר נוסף בסוגריים. למשל אם רוצים להתעלם מאותיות גדולות/קטנות,
מוסיפים דגל של re.I ישנו דגל שמאפשר להשתמש ב- ^ ו- $ עבור כל שורה
בטקסט מרובה שורות על כל שורה במקום רק על התחלה וסיום מחרוזת.
ישנם עוד דגלים שלעיתים ניתן להיעזר בהם.

תירגול של ביטויים רגולריים

.1
נתונה מחרוזת, יש לכתוב פונקציה בפייתון שתחזיר אמת אם יש במחרוזת רק
ספרות בינריות, ושקר אחרת.

הבדיקה תיעשה בעזרת ביטוי רגולרי מתאים.

.2
נתונה מחרוזת ובה ישנם מספרים בני 3 ספרות. יש לכתוב פונקציה שתיעזר
בביטוי רגולרי ותחזיר ברשימה את כל המספרים הללו. הם חייבים להיות מוקפים
ברווחים לפני ואחרי.