

כדי להשתמש בתבנית של re כמשתנה, עלינו לבצע 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 - נראה 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 (a-z A-Z 0-9 \_)

\W כל דבר שאינו \w

\s כל תו שהוא נראה כרווח (גם התו של שורה חדשה \n וגם tab או \t)

\S כל מה שאינו \s

התוים המיוחדים הבאים משמשים כעזר לגידור תוים אחרים.

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 סיפורה: \d

למשל: \d\d\d.\d\d\d\d\d\d\d  
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 (character 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 משהו 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 לסיפורה 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

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

+ 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](mailto:js3104@nyu.edu) [freddy.Krooger@yahoo.net](mailto:freddy.Krooger@yahoo.net) [aaron@gmail.com](mailto:aaron@gmail.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 הדוממין 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 הדוממין, שם הדוממין וקבוצה עבור הרמה הגבוהה של הדוממין. כאשר משתמשים בקבוצות, השימוש ב - findall יותר מורכב, כי הוא מחזיר את כל הקבוצות שתאמו.

### example8

יחד עם הקבוצות ניתן גם להשתמש בשיטה שניקראת 'התייחסות 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.

## דגלים

זהו פרמטר נוסף בסוגריים. למשל אם רוצים להתעלם מאותיות גדולות/קטנות, מוסיפים דגל של re.I ישנו דגל שמאפשר להשתמש ב-  $\wedge$  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.