

Programarea algoritmilor

Laborator 8 -

translate, maketrans

`str.translate(tabel_traducere)`

- `tabel_traducere` – de obicei obținut cu `str.maketrans(x[, y[, z]])`
(sau dicționar cu coduri Ascii)

`str.maketrans(x[, y[, z]])`

- `x` – dicționar sau șirul cu caracterele de înlocuit
- `y` – șirul cu caracterele noi (de aceeași lungime cu `x`) => caracterul `x[i]` se va înlocui cu `y[i]`
- `z` – șir cu caracterele care vor fi șterse

EXEMPLE

1. Se citește un text conținând separatorii uzuali(„,:;”) Sa se înlocuiască toți separatorii cu spațiu.
2. Se citește un cuvânt format cu litere mici. Să se înlocuiască fiecare vocală din cuvânt cu următoarea literă din alfabet.
3. Aceeași cerință ca la 2, dar în plus să se șteargă semnele: virgula, punct, două puncte.
4. Se citește o propoziție. Să se înlocuiască fiecare cifră < 5 care apare în text cu denumirea ei (1-unu, 2-doi, 3- trei, 4 -patru)

Regex (Regular Expressions)

- *modul de lucru cu* **expresii regulate** (Regular Expressions)
- Ce sunt **expresiile regulate**?
 - **microlimbaj specializat** încorporat în Python și pus la dispoziție în modulul **re**
 - utilizat pentru a **detecta tipare în stringuri**, pentru a **înlocui acele tipare** cu alte stringuri, pentru a **selecta un string după anumite tipare**, etc

- limbaj relativ mic și limitat: nu orice sarcina de procesare de stringuri poate fi realizată folosind expresii regulate
- **Anatomia** unei expresii regulate (RE)
 - alcătuită din **caractere simple** și **metacaractere**
 - **caracterele simple** se reprezintă pe ele însele, fără vreo altă semnificație suplimentară
 - **metacaracterele** ajută la definirea unor tipare mai complexe, detaliate mai jos
 - lista completă a metacaracterelor: `.^$*+?{}[]\|()` (acestea trebuie escapeuate de un backslash)

Documentația pentru modulul de expresii regulate se regăsește la acest link:
<https://docs.python.org/3/library/re.html>

- Un software utilitar foarte bun pentru a testa diverse Regex-uri se regăsește aici:
<https://regex101.com/>
- Un cheatsheet cu toate regulile de definire ale unei expresii regulate se regăsește aici:
https://images.datacamp.com/image/upload/v1665049611/Marketing/Blog/Regular_Expressions_Cheat_Sheet.pdf

De ce sunt bune aceste expresii regulate? Să zicem că avem de citit dintr-un fișier sau de la standard input un text care conține date calendaristice, cum am putea extrage aceste informații?

- Putem formula o expresie regulată care descrie această structură, tipar de informație (de obicei o dată calendaristică este structurată sub forma zz.ll.aaaa). Expresia regulată poate fi următoarea: `\d{2}\.\d{2}\.\d{4}`. Bineînțeles că această expresie regulată **NU** garantează corectitudinea informației extrase (i.e. Regex-ul anterior validează următorul șir de caractere 99.99.9999, care nu este o dată calendaristică validă). Astfel am putea folosi puterea acestor expresii regulate doar pentru extragerea informației, urmând să validăm aceste valori separat.

Exemple:

- **split** (împărțirea unei propoziții după o mulțime separator, după o combinație de separatori dintr-o mulțime, după orice caracter care nu este cifră este separator (+combinații) pentru a obține cuvinte care sunt numere, după orice caracter care nu este literă sau semn de punctuație este separator (+combinații))
- **sub** (înlocuirea unui cuvânt cu alt cuvânt, înlocuirea unui cuvânt cu alt cuvânt doar dacă este precedat de un anumit șir)
- **findall - căutare** (cuvintele care încep cu litera A sau a, șirurile binare din text, șirurile binare din text de lungime minim 3, numerele dintr-un text, numerele care se continua cu “ lei” pentru a le face suma)
- **match vs findall**
 - detectarea datei dintr-un text de forma de forma:
Nume eveniment - data: detalii eveniment

Probleme

1. Verificați în documentația Python 3.x, modulul re, funcțiile **compile**, **search**, **sub**, **findall**, **match**, **group**, **start**, **end** și evidențiați ce rol au.
2. Sa verificam daca o parola e
 - a) puternica: daca contine cel putin o litera mica, cel putin o litera mare, cel putin o cifra, cel putin un simbol special (fara spatii), are lungimea minim 10 caractere
 - b) medie: daca contine cel putin o litera mica, una mare, o cifra, are lungimea intre 6 si 9 caractere
 - c) slaba: daca are doar litere mici sau doar mari sau doar cifre, iar lungimea este maxim 5