

## Tartalomjegyzék

- 1 Bevezetés
- 2 Feladatok
  - ♦ 2.1 Pár sorosak
  - ♦ 2.2 Hosszabbak
  - ♦ 2.3 Adat rendezés

## Bevezetés

- Ami mindenképpen kelleni fog a python file-od elejére:

```
import re
```

- A függvények amiket használni fogunk a keresésre / átírára:

```
re.search("http://wiki.math.bme.hupattern"http://wiki.math.bme.hu, "http://wiki.math.bme.hustring"
re.match("http://wiki.math.bme.hupattern"http://wiki.math.bme.hu, "http://wiki.math.bme.hustring"
re.findall("http://wiki.math.bme.hupattern"http://wiki.math.bme.hu, "http://wiki.math.bme.hustring"
re.sub("http://wiki.math.bme.hupattern"http://wiki.math.bme.hu, "http://wiki.math.bme.hureplace"ht
```

- Ezek egy match objektumot adnak vissza, feltéve hogy a match változóba mentjük, akkor a következő metódusok használhatók rajta:

```
match = re.search("http://wiki.math.bme.hui.a"http://wiki.math.bme.hu, "http://wiki.math.bme.hucio
match.group()
match.group(0)
match.string
```

- Ez egy hasznos függvény lehet a próbálgatáshoz:

```
def show_match(string, pattern):
    match = re.search(pattern, string)
    if match:
        return string[:match.start()] + "http://wiki.math.bme.hu<<"http://wiki.math.bme.hu
    else:
        return "http://wiki.math.bme.huno match"http://wiki.math.bme.hu
```

- További emlékeztetés a következő file-ban: regex.txt vagy az első előadáson: Előadás1  
Közkívánatra való tekintettel, show\_match több találatra (nem 100% a működése): show.py

## Feladatok

### Pár sorosak

Ezeknél a feladatoknál többnyire elég a show\_match függvényt használni, de nem kötelező. A példa bemenetek soronként értendők, nem mint egy nagy bemenet. Mindig az első találatot kell csak megtalálni a sorban.

- Írj reguláris kifejezést, ami illeszkedik a "http://wiki.math.bme.hukutya"http://wiki.math.bme.hu szóra.

## WebProg-2012/Gyakorlat2

Volt egyszer egy kiskutya, elment a vasarba.

- Írj reguláris kifejezést, ami illeszkedik bármilyen egybefüggő számsorozatra (pl: 540548394).

A meleg evszakban, amikor 5456 az idojaras-szabalyzo szatellitaknak a nagy tengereken 545534tombolo szelekkel kellett megharcolnia, az este gyakran hozott esot 54343 a Sareer peremen.

- Írj egy reguláris kifejezést, ami illeszkedik bármilyen számra, legyen az egész, tizedestört, vagy két szám hányadosaként felírt racionális!

A boszi zsilip 22547645 hibaja miatt akadozik a budapesti hajkozlekedes  
546.433 eve nem oltek amerikai nagykovetet  
Atadjuk a 65/87 nuklearis tudasunkat Vietnammak  
Monda Lajos 65 a nagy 546.34 kiraly

- Írj reguláris kifejezést ami illeszkedik telefonszámokra, amik a következő formátumban vannak:  
06###-###-####

Egy Liga nem-hajo semleges területen voltak, ami mellett 0623-434-5432 Bene Gesserit- es Tleilax-m  
A Liga 0532-323-4344 meghunyaszkodva igyekezett 06-23-545-5432 megbekiteni a Bene Gesseritet.  
A kis ovalis 06sa-344-7654 terem, ahol 0643-43-65443 talalkoztak, szokasosan rezfalu es "http://wi

- Írj reguláris kifejezést, ami e-mail címekre illeszkedik. Azt nem kell ellenőriznie, hogy valós cím lehet-e, viszont azt ellenőrizzük, hogy a karakterek érvényesek legyenek, azaz vagy betű vagy szám, és az utolsó részben (ahol a .hu, meg .com stb van) ne lehessen szám.

Ez legalabb@visszajelzes.volt.  
Scytale megfelelokepp@valaszolt.86, de@feszulten.figyelt kozben.  
Nem tetszett.neki a@helyzet. @A.e haz@egy.zsakutca vegeben allt.

- Írj python scriptet, ami egy sorban kicseréli az összes előfordulását a 42-nek vagy betűvel írt "http://wiki.math.bme.hunegyvenketto"http://wiki.math.bme.hu-nek vagy "http://wiki.math.bme.hunegyvenket"http://wiki.math.bme.hu-nek, 23-ra. Arra vigyázzunk, hogy a negyvenketto es negyvenket bárhogy állhat nagy- és kisbetűkből.

vOlt eGySzeR 42 kiskUTYa, eLMENTeK negYvenKEt vasArBA. oTt TaLaLKOZtAk 43 FARKaSAl, aKIKnek Volt

Bónusz:

- Az előző feladat példa bemenetét alakítsuk normális szöveggé, azaz mindenhol kisbetűvel legyenek a szavak, kivéve a mondatok elején (mondat eleje a sor elején van és a .-ok után). Használhatjátok az str.lower() függvényt, ami visszaadja a beadott stringet mindenhol kisbetűvel, majd ezután kellene megoldani, hogy a mondatok eleje nagybetűs legyen.

## Hosszabbak

- Írj python scriptet ami megszámolja hányszor szerepel az "http://wiki.math.bme.hua"http://wiki.math.bme.hu és "http://wiki.math.bme.huaz"http://wiki.math.bme.hu névelő a szövegben:

A hamisitas elleni kuzdelmet segitheti amerikai kutatok fejlesztese, amelyben nanoreszecskek es fluoreszcens tinta felhasznalasaval allitottak elo szabad szemmel nem lathato vonalkodokat. A QR-kodok fekete es fehér blokkjai egyre inkabb elterjednek a kereskedelemben, mert szazzsor

több információt hordoznak a hagyományos vonalkódokhoz képest. A nyomtatott formában megjelenő hirdetésekben elhelyezett QR-kódokat okostelefonokkal lehet leolvasni és közvetlenül a hirdető cég honlapjára irányítják az érdeklődőt. Később kidolgozták, hogy csak lezerfényben legyen látható a QR-kód nyomtatásához alkalmazott tinta, így a bankjegyhamisítás és más termékek hamisítása elleni küzdelemben is bevezethető a módszer.

A Del-Dakotai Egyetem és a Del-Dakotai Banyaszati és Muszaki Egyetem kutatói az általuk kifejlesztett láthatatlan QR-kódhoz fluoreszcens tintába kevert nanométeres méretű (a nanométer a milliméter milliommód részé) részecskéket használtak. Az így készült kód csak akkor látszik, ha infravörös közeli hullámhosszú lezerrel világítják meg a felületet. Az adott hullámhosszon a nanorészecskék képesek elnyelni, abszorbeálni a fotonokat, majd rövidebb, látható hullámhosszon kibocsátják őket.

A kód, amikor megvilágítják a lezerrel, mobiltelefonnal leolvasható és nagyobb biztonságot nyújt a korábbi eljárásnál, mert egészen apró, csak mikroszkóp alatt látható részleteket lehet elrejteni benne az új technika segítségével. További előnye, hogy a láthatatlan kód még akkor is leolvasható, ha a hordozó papírt sokszor összegyurtek. Raadásul papíron kívül más anyagra is nyomtatható, például üvegre vagy műanyag filmre, ezért számos különböző területen használhatják a hamisítás elleni védekezésre. Az új technológiát a Nanotechnology című szakfolyóiratban ismertette Jeevan Meruga kutatásvezető és csoportja.

- Írj python scriptet, ami kiírja az összes mondatot amiben szerepel a nano szó (bárhogy lehet, nagy- és kisbetűkkel). A mondatok sor elejétől .-ig, vagy .-től .-ig terjednek.

A hamisítás elleni küzdelmet segítheti amerikai kutatók fejlesztése, amelyben nanorészecskék és fluoreszcens tinták kombinációját használják. A QR-kódok fekete és fehér blokkjai egyre inkább elterjednek a kereskedelemben, mert százszor több információt hordoznak, mint a hagyományos vonalkódok. A Del-Dakotai Egyetem és a Del-Dakotai Banyaszati és Muszaki Egyetem kutatói az általuk kifejlesztett láthatatlan QR-kódhoz fluoreszcens tintába kevert nanométeres méretű (a nanométer a milliméter milliommód részé) részecskéket használtak. Az így készült kód csak akkor látszik, ha infravörös közeli hullámhosszú lezerrel világítják meg a felületet. Az adott hullámhosszon a nanorészecskék képesek elnyelni, abszorbeálni a fotonokat, majd rövidebb, látható hullámhosszon kibocsátják őket.

- Írd át az előző scriptet függvényé (ha még nem az), úgy hogy tetszőleges szóra tudj keresni.

## Adat rendezés

- Töltsd le a következő file-t: [adat.txt](#)
- Véletlen adatok és szavak vannak benne, a feladat az, hogy soronként kibányásszuk az e-mailcímeket, telefonszámokat, és dátumokat és mindezt tároljuk egy 2 dimenziós listában (listák listája).
- Az e-mailcímek és telefonszámok a fenti feladatokban már említett módon néznek ki, míg a dátumok 00/00/0000 formátumban vannak.
- A kimenetet file-ba írni nem feltétlen kell, de az lenne a legszebb megoldás, ha soronként lenne a megtalált e-mailcím majd telefonszám és végül a dátum.
- A feladathoz olvasnod kell tudni file-ban a python segítségével, ebben a következő kód segít.

```
bemenet = open('./adat.txt', 'r')
kimenet = open('./kimenet.txt', 'w')

def fuggveny(s):
    return "http://wiki.math.bme.hu/valami" + s

for line in bemenet:
    line2 = fuggveny(line)
    kimenet.write(line2)

adat.close()
kimenet.close()
```

- Parancsonként leírom mi mit csinál a kódban:

```
bemenet = open('./adat.txt', 'r')
```

- Megnyitja a bemenet változóba az adat.txt file-t ami egy mappában van a python scripttel, az 'r' (read) azt mondja meg, hogy olvasni akarunk a file-ban.

```
kimenet = open('./kimenet.txt', 'w')
```

- Mint az előbb, csak 'w' (write) segítségével írásra nyitjuk meg.

```
def fuggveny(s):  
    return "http://wiki.math.bme.hu/valami"http://wiki.math.bme.hu + s
```

- Csak valamilyen függvény ami csinál valamit a bemenetével majd visszaad valamit.

```
for line in bemenet:  
    line2 = fuggveny(line)  
    kimenet.write(line2)
```

- Végigmegyünk a bemenet sorain (itt a line csak egy változónév, lehetne i is, vagy koafkdofd is akár), és lefuttatjuk a fuggveny-t minden sorra, a kimenetet elmentjük a line2-be, majd a kimenet file-ba írjuk a line2-t.

```
adat.close()  
kimenet.close()
```

- Bezárjuk mindkét file-t, ez azért kell, mert csak azután lehetünk biztosak hogy megtörténtek a file-ba írások miután így bezártuk a file-t.