
Datu kartošana

Referats

Eva Martinsone

2021-02-24



Contents

Darba mērķis	2
Izmantotie moduļi	2
CSV faila imports	2
Datu indeksēšana	3
Nodrošina datu indeksēšanu pēc kolonas 'Organizācija'	3
Nodrošina konkrēto vērtību atlasīšanu pēc indeksētās kolonas 'Organizācija'	3
Nodrošina iepriekšējā punktā iegūto datu tālāko indeksēšanu to tālākai apstrādei pēc kolonas 'Darba grupa'	3
Nodrošina konkrēto vērtību atlasīšanu pēc indeksētajiem datiem, kas nepieciešams e-pastu saturam	3
Epastsu izsūtīšana	4
Nodrošina SMTP savienojuma izveidi	4
Nodrošina e-pasta pamatelementus	4
Nodrošina e-pasta servisamgi	4
Izsūtītāja e-pasta rekvizīti	4
Pieslegums e-pasta servisiem	5
E-pasta saturs	5
Nodrošina e-pasta izsūtīšanu	5

Darba mērķis

Veikt datu kārtošanu pēc vairākiem atlasē kritēriem. Iegūtās datu kopas nosūtīt uz e-pastu atbildīgajiem

Izmantotie moduļi

pandas.DataFrame modulis izmantots ar mērķi veikt darbības ar datiem tabulā gan pēc rindu, gan pēc kolonnu etiķetēm smtpplib modulis izmatots ar mērķi izsūtīt e-pastu

CSV faila imports

pandas.read_csv - nodrošina csv faila nolasīšanu, kurā ir atdalītas vērtības r - nodrošina nepabeigtu vērtību atdalīšanu (faila atrašanās ceļš) - importētā faila atrašanās vieta

Izpildītais kods:

```
1 saraksts = pd.read_csv (r'C:\Users\Lietotajs\Desktop\ievades_dati.csv')
2 print(saraksts)
```

Datu indeksēšana

Izmantotas funkcijas: `.index` - nodrošina katras tabulas kolonas/līnijas indeksēšanu kā atevišķu vienumu `.loc` - nodrošina indeksēto tabulas kolonu/līniju vienumu atlasī

Nodrošina datu indeksēšanu pēc kolonas 'Organizācija'

Izpildītais kods:

```
1 pak_kolonas = saraksts.set_index('Organizacija')
2 print(pak_kolonas)
```

Nodrošina konkrēto vērtību atlasī pēc indeksētās kolonas 'Organizācija'

Izpildītais kods:

```
1 iestade = pak_kolonas.loc[['Labklājības departaments', 'Rīgas Sociālais
    dienests', 'Rīgas pasvaldības policija', 'Sociālais dienests']]
2 print(iestade)
3 print(pak_kolonas)
```

Nodrošina iepriekšējā punktā iegūto datu tālāko indeksēšanu to tālākai apstrādei pēc kolonas 'Darba grupa'

Izpildītais kods:

```
1 pak_vaditaji = iestade.set_index('Darba grupai')
2 print(pak_vaditaj)
```

Nodrošina konkrēto vērtību atlasī pēc indeksētajiem datiem, kas nepieciešams e-pastu saturam

Izpildītais kods:

```
1 Normunds = pak_vaditaji.loc[['Datu bazu un sistemu nodala']]
2 print('DBSN pieteikumi', Normunds)
3
4 Mihails = pak_vaditaji.loc[['Tehnisko resursu uzturesanas sektors']]
5 print('TRU sektota pieteikumi', Mihails)
6
7 Jorens = pak_vaditaji.loc[['Fiziskas drosibas un integreto sistemu
   sektors']]
8 print('FDIS sektota pieteikumi', Jorens)
```

Epatsu izsūtīšana

Katram no iepriekšējā sadaļā iegūtajām datu kopām, kas iegūta pēc e-pasta satura tiek sagatavots e-pasta sagatave pēc turpmāk redzamajās apakšsadaļās

Nodrošina SMTP savienojuma izveidi

```
1 import smtplib
```

Nodrošina e-pasta pamatelementus

```
1 To = "mmmartinsone@gmail.com"
2 Subject = "neizpildītie uzdevumi"
3 Text = 'tetsa e-pasts'
```

Nodrošina e-pasta servisamgi

```
1 smtp_server = "smtp.gmail.com"
2 port = 587
```

Izsūtītāja e-pasta rekvizīti

```
1 e_pasta_sender = "mmmartinsone@gmail.com"
2 e_pasta_parole = "parole"
```

Pieslegums e-pasta servisiem

```
1 server = smtplib.SMTP('smtp.gmail.com', 587)
2 server.ehlo()
3 server.starttls()
4 server.ehlo()
5 server.login(e_pasta_sender, e_pasta_parole)
```

E-pasta saturs

```
1 Text = '\r\n'.join([
2     'To: %s' % To,
3     'From: %s' % e_pasta_sender,
4     'Subject: %s' % Subject,
5     '',
6     Normunds
7 ])
```

Nodrošina e-pasta izsūtīšanu

```
1 try:
2     server.sendmail(e_pasta_sender, [To], saturs)
3     print('epasts nosutits')
4 except:
5     print('e_pasts nav nosutits')
6 server.quit()
```

Lorem markdownum Letoia, et alios: figurae flectentem annis aliquid Peneosque ab esse, obstat gravitate. Obscura atque coniuge, per de coniunx, sibi **medias commentaque virgine** anima tamen comitemque petis, sed. In Amphion vestros hamos ire arceor mandere spicula, in licet aliquando.