```
1: ////////////////////////////////////////////////////////
2: Python Pascal Synthetic Data SynDat
3: ____________________________________________________
4: maXbox Starter 91 - Generate Synthetic Data. - Max Kleiner
5:
6: "make the fake"   -mX47
7:
8:  https://my6.code.blog/2021/11/12/duesseldorf-ekon-25/
9:  https://pypi.org/project/Faker/
10:
11: Faker is a Python library that generates fake data. Fake data is often used for
    testing or filling databases with some dummy data. Faker is heavily inspired by
    PHP's Faker, Perl's Data::Faker, and by Ruby's Faker.
12: We are also able to sample from a model and create synthetic data, hence the name
    SynDat. The most obvious way that the use of synthetic data benefits data science
    is that it reduces the need to capture data from real-world events, and for this
    reason it becomes possible to generate data and construct a dataset much more
    quickly than a dataset dependent on real-world events and in addition you dont get
    lost with data protection.
13:
14: Then you had to install faker:
15: C:\Users\Max\AppData\Local\Programs\Python\Python36-32>
16:                                          python -m pip install faker
17:
18:     Install a 32 bit package module in a 64 bit environment:
19:     1. Change to your 32 bit path with cd:
20:        cd C:\Users\Max\AppData\Local\Programs\Python\Python36-32>
21:
22:     2. Call the Pip (e.g. faker module) explicitly with python.exe:
23:        python -m pip install faker
24:
25: And it runs:
26: Downloading https://files.pythonhosted.org/packages/27/ab/0371598513e8179d9053
27: 911e814c4de4ec2d0dd47e725dca40aa664f994c/Faker-9.9.0-py3-none-any.whl (1.2MB)..

28: You are using pip version 9.0.1, however version 21.3.1 is available.
29: You should consider upgrading via the 'python -m pip install --upgrade pip'.
30: C:\Users\Max\AppData\Local\Programs\Python\Python36-32>
31: >>>>>>>>>>>>>>>>>>>>>>>
32:
33: Now we start the program:
34: The fake.Faker (fake = Faker()) creates and initializes a faker generator, which
    can generate data by accessing properties named after the type of data.
35:
36: Faker delegates the data generation to providers. The default provider uses the
    English locale. Faker supports other locales; they differ in level of completion.
37:
38:     eg.execStr('from faker import Faker');
39:     eg.execStr('import simplejson as json');  //# instead of import json
40:     eg.execStr('fake = Faker()');
41:     println(eg.evalStr('fake.profile()'));
42:
43: Check the output with path and list the profile, the example outputs a fake name,
    address, and many more items of a persons profile:
44:
45: 'job': 'Chemist, analytical', 'company': 'Austin-Jarvis', 'ssn': '107-81-7090',
    'residence': '4958 Jose Club Apt. 222\nSouth Beverlyborough, CO 86898',
    'current_location': (Decimal('14.167332'), Decimal('8.098724')), 'blood_group':
    '', 'website': ['https://rose.com/', 'http://www.long.info/'], 'username': 'rmoran',
    'name': 'Jo Graham', 'sex': 'F', 'address':
     NY 95661', 'mail': 'rpierce@yahoo.com', 'birthdate': datetime.date(1965, 12, 25)}
46:
47: This is json and we can convert it. I tried first with json and simplejson, got
    some date and decimals serialize exceptions (Object of type date is not JSON
    serializable.), then used dumper, but got a next exception Exception: <class
    'AttributeError'>: 'NoneType' object has no attribute 'write'.:
48:
```

```
Collecting dumper
  Downloading https://files.pythonhosted.org/packages/54/74/99188ad91edbdf45db3b
95a3fe648fa5d61e340beef13bc119a06b6033e8/Dumper-1.2.0-py2.py3-none-any.whl
Installing collected packages: dumper
Successfully installed dumper-1.2.0

So used f in format string:
    println(eg.evalStr('f"my profile: {fake.profile()}"'));

The next line is a simple demonstration of Faker credit card:

   println(eg.evalStr('fake.credit_card_number()'));
   >>> 4784693663655

Faker also support for dummy hashes and uuids:

#!/usr/bin/env python

from faker import Faker

faker = Faker()
print(f'md5: {faker.md5()}')
print(f'sha1: {faker.sha1()}')
print(f'sha256: {faker.sha256()}')
print(f'uuid4: {faker.uuid4()}')

So we test the whole cheatsheet line by line with eval and exec. Eval() function
accepts a string argument and if the string argument is expression then eval() will
evaluate the expression. Below I showed the example code.
http://www.softwareschule.ch/examples/pydemo32.txt

begin  //@main
  //myloadscript2:= filetostring(PYSCRIPT2);
  PyForm:= loadForm2(200,300, clgreen,'PyFrm4D_SynDat_Tester_EKON25');
  pyMemo:= TMemo.create(PyForm);
  apd:= TApdMeter.create(pyform);
  apd.parent:= pyform;
  apd.barcolor:= clred;
  apd.position:= 30;
  apd.SetBounds(10,100, 150, 150);
  apd.visible:= true;
  pyMemo.parent:= PyForm
  PyForm.show;
  eg:= TPythonEngine.Create(Nil);
  try
    eg.pythonhome:= PYHOME;
    eg.loadDLL;
    println('test import '+eg.EvalStr('__import__("decimal").Decimal(0.1)'));
    //https://medium.com/@swathiarun63/10-fantastic-python-packages-af2a16a1183a
    apd.position:= 60;
    pyMemo.lines.add('call test with execStr() from faker import Faker');
    sw:= TStopWatch.Create();
    sw.Start;
    eg.execStr('from faker import Faker');
    eg.execStr('import simplejson as json');  //# instead of import json
    eg.execStr('import dumper');
    eg.execStr('fake = Faker()');
    println(eg.evalStr('fake.profile()'));
    eg.execStr('profile1 = fake.simple_profile()');
    //println(eg.evalStr('dumper.dump(profile1)'))
    println(eg.evalStr('f"my profile: {fake.profile()}"'));
    //println(eg.evalStr('json.dumps(fake.profile(),indent=4)'));
    println(eg.evalStr('fake.credit_card_number()')); //}
    sw.Stop;
    //sw.ElapsedMilliseconds;
    writeln('Stop Watch Faker Tester1: '+sw.getValueStr)
  except
```

```
114:     eg.raiseError;
115:     writeln(ExceptionToString(ExceptionType, ExceptionParam));
116:   finally
117:     eg.Free;
118:     sw.Free;
119:     sw:= Nil;
120:     apd.position:= 100;
121:   end;
122: End.
123:
124: The program generates an instance of a python engine in line 90 and generates list
     of fake profiles. The list is passed to an eval() template in line 104 to be
     processed and performance is measured at line 112ff. The template is located in the
     precompiled engine of the box. The generated content is written as println() to the
     console and an external file with saveString().
125:
126: Python Cheat Sheet: Functions and Tricks
127: ----------------------------------------
128: http://www.softwareschule.ch/examples/cheatsheetpython.pdf
129: http://www.softwareschule.ch/examples/pydemo13_cheatsheet_Tutorial_90.txt
130: http://www.softwareschule.ch/examples/pydemo13_cheatsheet_Tutorial_90.htm
131:
132: Conclusion:
133: In this tutorial, we have used Python Faker to generate fake data in Python and
     maXbox with measuring time behaviour.
134: Finally, synthetic datasets can minimize privacy concerns. Attempts to anonymize
     data can be ineffective, as even if sensitive/identifying variables are removed
     from the dataset, other variables can act as identifiers when they are combined.
     This isn't an issue with synthetic data, as it was never based on a real person, or
     real event, in the first place. A concept could mean, firms, institutes or simply
     users dont deal with real person data, they got an avatar which makes an
     relationship between a hash and a guid in a proxy blockchain (pb1). A real person
     is protected behind the SynDat proxy with a guid record.
135: Python for .NET is also a package that gives Python programmers nearly seamless
     integration with the .NET Common Language Runtime (CLR) and provides a powerful
     application scripting tool for .NET developers.
136:
137: Script Ref:
138: http://www.softwareschule.ch/examples/pydemo32.txt
139:
140: Concept of SynDat:
141: http://www.softwareschule.ch/examples/syndat.png
142:
143: https://medium.com/geekculture/blockchain-explained-in-50-lines-of-code-1dbf4eda0201
144: https://entwickler-konferenz.de/blog/machine-learning-mit-cai/
145: https://www.unite.ai/what-is-synthetic-data/
146:
147: ************************************************************
148: Release Notes maXbox 4.7.6.10 II November 2021 mX476
149: ************************************************************
150: Add 10 Units + 3 Tutorials
151: 1441 unit uPSI_neuralgeneric.pas; CAI
152: 1442 unit uPSI_neuralthread.pas; CAI
153: 1443 unit uPSI_uSysTools; TuO
154: 1444 unit upsi_neuralsets; mX4
155: 1445 unit uPSI_uWinNT.pas mX4
156: 1446 unit uPSI_URungeKutta4.pas ICS
157: 1447 unit uPSI_UrlConIcs.pas ICS
158: 1448 unit uPSI_OverbyteIcsUtils.pas ICS
159: 1449 unit uPSI_Numedit2 mX4
160: 1450 unit uPSI_PsAPI_3.pas mX4
161: Total of Function Calls: 35078
162: SHA1: of 4.7.6.10  D4B0A36E42E9E89642A140CCEE2B7CCDDE3D041A
163: CRC32: B8F2450F 30.6 MB (32,101,704 bytes)
164:
165:
166: Appendix: Verifying EU Digital COVID-19 Certificate with Python CWT
```

```
# 1. Loads a DSC as a COSEKey for verifying a signature in EUDCC.
public_key = load_pem_hcert_dsc(dsc)
# 2. Verifies and decodes a target EUDCC.
decoded = cwt.decode(eudcc, keys=[public_key])
# 3. Get the payload of the EUCC. It is a JSON-formatted Electronic Health
Certificate as follows:
claims = Claims.new(decoded)
# claims.hcert[1] == decoded[-260][1] ==
# {
#     'v': [
#         {
#             'dn': 1,
#             'ma': 'ORG-100030215',
#             'vp': '1119349007',
#             'dt': '2021-02-18',
#             'co': 'AT',
#             'ci': 'URN:UVCI:01:AT:10807843F94AEE0EE5093FBC254BD813#B',
#             'mp': 'EU/1/20/1528',
#             'is': 'Ministry of Health, Austria',
#             'sd': 2,
#             'tg': '840539006',
#         }
#     ],
#     'nam': {
#         'fnt': 'MUSTERFRAU<GOESSINGER',
#         'fn': 'Musterfrau-Gößinger',
#         'gnt': 'GABRIELE',
#         'gn': 'Gabriele',
#     },
#     'ver': '1.0.0',
#     'dob': '1998-02-26',
# }

                _od#HMM6&*MMMH::-_
            _dHMMMR??MMM? ""| `"'-?Hb_
         .~HMMMMMMMMHMMM#M?          `*HMb.
       ./?HMMMMMMMMMMM"*"""            &MHb.
      /'|MMMMMMMMMM'                -    `*MHM\
     /  |MMMMMMHHM''                      .MMMHb
    |   9HMMP    .Hq,                     TMMMMMH
   /     |MM\,H-""&&6\__                   `MMMMMMb
  |       `""HH#,       \                - MMMMMMM|
  |           `HoodHMM###.                  `9MMMMMH
  |              .MMMMMMM##\                  `*"?HM
  |          ..  ,HMMMMMMMMMMo\.                  |M
  |              |MMMMMMMMMMMMMMMHo               |M
  |               ?MMMMMMMMMMMMMMM*               |H
  |.               `#MMMMMMMMMMMM'              .M|
   \                  `MMMMMMMMMM*               |P
   `\                  MMMMMMMMT"'              ,H
    `\                `MMMMMMH?                 ./
      \.              |MMMH#"                  ,/
       `\.            |MMP'                 ./''
         `~\          `HM:.-     .        ,/''
            "-\_          '_\  .       _.-"
                "-\-#odMM\_,oo==-"

    |   n   p |                   |   n   p |    Stemmer more false positive
    |   e   o |                   |   e   o |
    |   g   s |                   |   g   s |
----+---------+               ----+---------+
neg |<119>131 |               neg |<110>140 |
pos |   5<245>|               pos |   5<245>|
----+---------+               ----+---------+
(row = reference; col = test)
```