```
1: ////////////////////////////////////////////////////////
2: Python Pascal Cheat Sheet
3: ____________________________________________________
4: maXbox Starter 90 - Python4Delphi CheatSheet. - Max Kleiner
5:
6: "bit or bat" "welcome bitman"
7:                                - mX47
8:
9:  https://my6.code.blog/2021/09/08/improver-4/
10:
11: Maybe you know something like this:
12:   Exception: <class 'ModuleNotFoundError'>: No module named 'pyquery'.
13:
14: Then you had to install pyquery:
15: C:\Users\Max\AppData\Local\Programs\Python\Python36-32>
16:                                          python -m pip install pyquery
17: Collecting pyquery
18:   Downloading https://files.pythonhosted.org/packages/58/0b/85d15e21f660a8ea68b1
19: e0286168938857391f4ec9f6d204d91c9e013826/pyquery-1.4.3-py3-none-any.whl
20: Collecting lxml>=2.1 (from pyquery)
21:   Cache entry deserialization failed, entry ignored
22:   Downloading https://files.pythonhosted.org/packages/35/ea/88c9245575f834443976
23: a04356eaac2b3a6b0d7210a65ec5e1cafd5e23fa/lxml-4.6.4-cp36-cp36m-win32.whl (3.2MB)
24:
25:     100% |||||||||||||||||||||||||||||||| 3.2MB 308kB/s
26: Collecting cssselect>0.7.9 (from pyquery)
27:   Cache entry deserialization failed, entry ignored
28:   Downloading https://files.pythonhosted.org/packages/3b/d4/3b5c17f00cce85b9a1e6
29: f91096e1cc8e8ede2e1be8e96b87ce1ed09e92c5/cssselect-1.1.0-py2.py3-none-any.whl
30: Installing collected packages: lxml, cssselect, pyquery
31: Successfully installed cssselect-1.1.0 lxml-4.6.4 pyquery-1.4.3
32: You are using pip version 9.0.1, however version 21.3.1 is available.
33: You should consider upgrading via the 'python -m pip install --upgrade pip'.
34: C:\Users\Max\AppData\Local\Programs\Python\Python36-32>
35: >>>>>>>>>>>>>>>>>>>>>>
36:
37: First of all, if you want to install a package especially for 32 bit go to the path where the
    32-bit installation is and call exlicitly the 32bit python exe:
38:
39: C:\Users\Max\AppData\Local\Programs\Python\Python36-32>  python -m pip install pyquery
40:
41: Check the path with path and list the installation with py:
42:
43: C:\Users\Max\AppData\Local\Programs\Python\Python36-32>py -9
44: Python 9 not found!
45: Installed Pythons found by py Launcher for Windows
46:
47:  -3.7-64 *
48:  -3.6-32
49:
50: 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 an example code.
51:
52: eng.Execstring('from tweetlib import tweetlib');
53: eng.Execstring('tweets = tweetlib.TweetLib(query_search="Programming", since="2021-10-22")');
54: println('tip11 getmytweeets: '+eng.EvalStr('tweets.get_tweets()'));
55:
56:
57: Python Cheat Sheet: Functions and Tricks
58: ----------------------------------------
59: http://www.softwareschule.ch/examples/cheatsheetpython.pdf
60: http://www.softwareschule.ch/examples/pydemo13_cheatsheet_Tutorial_90.txt
61: http://www.softwareschule.ch/examples/pydemo13_cheatsheet_Tutorial_90.htm
62:
63: Begin  //@Main
64: //<Executable statements>
65: //https://www.amazon.com/Patterns-konkret-Max-Kleiner/dp/3935042469
66: { ISBN-10 ? : 3935042469  ISBN-13 ? : 978-3935042468}
67:
68:   eg:= TPythonEngine.Create(Nil);
69:   eg.pythonhome:= PYHOME32;
70:   eg.opendll(PYDLL32)
71:   //eng.IO:= pyMemo;
72:   try
```

```
73:     eg.Execstring('import base64'+LF+'import urllib.parse');
74:     eg.Execstring('import urllib.request, os, textwrap, json, requests');
75:     eg.Execstring(REXDEF);
76:
77:    { eg.Execstring('import nacl');
78:     eg.Execstring('from nacl.encoding import HexEncoder'+CRLF+
79:                   'from nacl.exceptions import CryptoError'+CRLF+
80:                   'from nacl.encoding import Base64Encoder'+CRLF+
81:                   'from pydub import AudioSegment');  }
82:
83:    //eg.Execstring('from Crypto.PublicKey import RSA');
84:
85:    println(eg.evalStr('base64.b64decode("2e8WuEr0+5nc14VBxQrOl4ob6guOTySr")'));
86:   //eng.Execstring('priv_key = nacl.public.PrivateKey.generate()');
87:   //openWeb('http://www.softwareschule.ch/examples/cheatsheetpython.pdf');
88:
89:   //# 1.map(func, iter) Executes the function on all elements of iterable
90:    println(eg.evalStr('list(map( lambda x: x[0],["red","green","blue"]))'));
91:
92:    >>> ['r', 'g', 'b']
93:
94:   //# 2.map(func, i1,...,Executes the function on all k elements of k iterables
95:    println(eg.evalStr('list(map(lambda x,y: str(x)+" "+y + "s",[0,2,2],'+
96:                                        '[ "apple" , "orange" , "banana" ]))'));
97:
98:    >>> ['0 apples', '2 oranges', '2 bananas']
99:
100:  //# 3.string.join(iter), Concatenates iterable elements separated by string
101:   println(eg.evalStr('" marries " .join(list([ "Alice" , "Bob" ]))'));
102:
103:   >>> Alice marries Bob
104:
105:  //# 4.filter(func,iterable),Filters out elements in iterable for func returns False (or 0)
106:   println(eg.evalStr('list(filter(lambda x: True if x> 17 else False,[1,15,17,18]))'));
107:
108:   >>> [18]
109:
110:  //# 5.string.strip(), Removes leading and trailing whitespaces of string
111:   println(eg.evalStr('( " \n \t 42 \t " .strip())'));
112:
113:   >>> 42
114:
115:  //# 6.sorted(iter), Sorts iterable in ascending order
116:   println(eg.evalStr('sorted([ 8 , 3 , 2 , 42 , 5 ])'));
117:
118:   >>> [2, 3, 5, 8, 42]
119:
120:  //# 7.sorted(iter,key=key) , Sorts according to the key function in ascending order
121:   println(eg.evalStr('sorted([ 8,3,2,42,5 ], key=lambda x: 0 if x== 42 else x)'));
122:
123:   >>> [42, 2, 3, 5, 8]
124:
125:  //# 8.help(func) , Returns documentation of func
126:  // println(eg.evalStr('help(''print'')'));
127:   saveString(exepath+'pyhelp.py', 'help(''print'')');
128:   print(getDosOutput('py '+exepath+'pyhelp.py', exePath));
129:
130:   >>> Help on built-in function print in module builtins:
131:
132:    print(...)
133:    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
134:
135:    Prints the values to a stream, or to sys.stdout by default.
136:    Optional keyword arguments:
137:    file:  a file-like object (stream); defaults to the current sys.stdout.
138:    sep:   string inserted between values, default a space.
139:    end:   string appended after the last value, default a newline.
140:    flush: whether to forcibly flush the stream.
141:
142:
143:  //# 9.zip(i1, i2, ...), Groups the i-th elements of iterators i1,i2,...together
144:   println(eg.evalStr('list(zip([''Alice'',''Anna''],[''Bob'',''Jon'',''Frank'']))'));
145:
146:   >>> [('Alice', 'Bob'), ('Anna', 'Jon')]
147:
```

```
148:   //# 10.Unzip, Equal to: 1) unpack the zipped list, 2) zip the result
149:    println(eg.evalStr('list(zip(*[(''Alice'',''Bob''),(''Anna'',''Jon'')]))'));
150:
151:    >>> [('Alice', 'Anna'), ('Bob', 'Jon')]
152:
153:   //# 11.enumerate(iter), Assigns a counter value to each element of iterable
154:    println(eg.evalStr('list(enumerate(["Alice","Bob","Jon"]))'));
155:
156:    >>> [(0, 'Alice'), (1, 'Bob'), (2, 'Jon')]
157:
158:    You can start your File hosting server by using the following command in the Python
    interpreter_:
159:
160:    # 12.python -m http.server<P>,Want to share files between PC and phone?
161:   //https://docs.python.org/3/library/http.server.html
162:    print(getDosOutput('py -m http.server<8080>', exePath));
163:    ExecuteShell('py', '-m http.server 8080');
164:
165:   //# 13.Read comic Open the comic series xkcd in your web browser
166:   //eg.Execstring('import antigravity');
167:
168:   //# 14.Zen of Python import this
169:    eg.execString('from this import *');
170:    println('14. import this: '+CRLF+
171:         StringReplace(eg.EvalStr('repr("".join([d.get(c,c) for c in s]))'),
172:                                                 '\n',CR+LF,[rfReplaceAll]));
173:
174:   >>> 14. import this:
175:    "The Zen of Python, by Tim Peters
176:
177:    Beautiful is better than ugly.
178:    Explicit is better than implicit.
179:    Simple is better than complex.
180:    Complex is better than complicated.
181:    Flat is better than nested.
182:    Sparse is better than dense.
183:    Readability counts.
184:    Special cases aren't special enough to break the rules.
185:    Although practicality beats purity.
186:    Errors should never pass silently.
187:    Unless explicitly silenced.
188:    In the face of ambiguity, refuse the temptation to guess.
189:    There should be one-- and preferably only one --obvious way to do it.
190:    Although that way may not be obvious at first unless you're Dutch.
191:    Now is better than never.
192:    Although never is often better than *right* now.
193:    If the implementation is hard to explain, it's a bad idea.
194:    If the implementation is easy to explain, it may be a good idea.
195:    Namespaces are one honking great idea -- let's do more of those!"

196:
197:   //# 15.Swapping numbers, Swapping vars is a breeze in Python. No offense, Java!
198:   eg.execString('a, b = ''Jane'' , ''Alice'''+CRLF+'a, b = b, a');
199:   println(eg.evalStr('a, b'));
200:
201:   >>> ('Alice', 'Jane')
202:
203:   //# 16.Unpacking arguments, Use a sequence as function arguments!
204:   eg.execString('def f (x, y, z) : return x + y * z');
205:   println(eg.evalStr('f(*[ 1 , 3 , 4 ])'));
206:   println(eg.evalStr('f(**{ ''z'': 4 , ''x'': 1 , ''y'': 3 })'));
207:
208:   >>> 13
209:   >>> 13
210:
211:   //# 17.Generate QRCode!
212:   eg.Execstring('import psutil');
213:   eg.Execstring('import pyqrcode');
214:
215:   eg.Execstring('Qr_Code = pyqrcode.create("maXbox4")');
216:   eg.Execstring('Qr_Code.svg("qrmx42.svg", scale=8)');
217:
218:   >>> http://www.softwareschule.ch/examples/qrmx42.svg
219:   >>> http://www.softwareschule.ch/examples/qrmx42.png
220:
```

```
221:   //# 18.Checking Free RAM
222:   println('Virtual Mem: '+
223:               eg.EvalStr('(__import__("psutil").virtual_memory())'));
224:
225:   >>> Virtual Mem: svmem(total=17098358784, available=9643110400, percent=43.6,
226:                                             used=7455248384, free=9643110400)
227:
228:  { eg.execString('import psutil, os')
229:   eg.execString('adlst = []')
230:   eg.execString('p = psutil.Process( os.getpid())');
231:   eg.execString('for dll in p.memory_maps():'+CRLF+' print(dll.path)');
232:   eg.execString('for dll in p.memory_maps():'+CRLF+' adlst.append(dll.path)');
233:   //println(eg.evalStr(' print(dll.path)'));
234:   println(eg.evalStr('p'));
235:   println(eg.evalStr('adlst'));
236:   println('dll list detect: '+
237:              StringReplace(eg.EvalStr('adlst'),',',CR+LF,[rfReplaceAll]));   }
238:
239:  { eng.Execstring(DEF_RSAKEYS);
240:       eng.Execstring('d=generate_RSA(bits=2048)')
241:       println('RSA Publickey '+eng.evalStr('d[1]'));  }
242:    //# Get the maximum number of complete TODOs.
243:    //println('user_max_complete = '+eng.evalStr('top_users[0][1]'));
244:
245:   except
246:     eg.raiseError;
247:   finally
248:     eg.Free;
249:     //aPythonVersion.Free;
250:   end;
251:   //GetJSONData;
252:   //maXcalcF('2^64 /(60*60*24*365)')
253: //<Definitions>
254: End.
255:
256:
257: New compiled and recompiled for mX4.7.6.10:
258: 13/11/2021 22:31 477,572    fMain.dcu
259: 13/11/2021 22:38 30,865     MathsLib.dcu
260: 13/11/2021 17:01 37,230     neuraldatasets.dcu
261: 13/11/2021 19:38 316,093    neuralnetworkCAI.dcu
262: 13/11/2021 22:35 6,568      PXLTiming.dcu
263: 13/11/2021 22:35 7,434      PythonAction.dcu
264: 13/11/2021 22:35 278,938    PythonEngine.dcu
265: 13/11/2021 17:41 23,968     uPSI_neuraldatasets.dcu
266: 13/11/2021 17:53 153,048    uPSI_NeuralNetworkCAI.dcu
267: 13/11/2021 17:41 9,359      uPSI_neuralthread.dcu
268: 13/11/2021 22:35 8,642      uPSI_PXLTiming.dcu
269: 13/11/2021 22:35 175,123    uPSI_PythonEngine.dcu
270: 13/11/2021 22:13 11,007     uPSI_uSysTools.dcu
271: 13/11/2021 22:31 2,720      uPSI_uWinNT.dcu
272: 13/11/2021 18:46 22,817     uSysTools.dcu
273: 13/11/2021 22:21 2,445      uWinNT.dcu
274: 13/11/2021 22:35 46,234     VarPyth.dcu
275: 13/11/2021 22:35 40,900     OverbyteIcsMimeUtils.dcu
276: 13/11/2021 22:35 77,136     OverbyteIcsUtils.dcu
277: 13/11/2021 18:44 56,530     RegExpr.dcu
278:
279: Script Ref: http://www.softwareschule.ch/examples/pydemo13_cheatsheet_Tutorial_90.txt
280:
281: https://python.plainenglish.io/13-python-advanced-code-snippets-for-everyday-problems-
     fb9874ea0b18
282: http://www.softwareschule.ch/examples/1073__CAI_3_LearnerClassifier22_Tutor_89_2.txt
283: https://entwickler-konferenz.de/blog/machine-learning-mit-cai/
284: https://www.freecodecamp.org/news/convolutional-neural-network-tutorial-for-beginners/
285:
286: ****************************************************************
287: Release Notes maXbox 4.7.6.10 November 2021 mX476
288: ****************************************************************
289: Add 5 Units + 2 Tutorials
290:
291: 1441 unit uPSI_neuralgeneric.pas; CAI
292: 1442 unit uPSI_neuralthread.pas; CAI
293: 1443 unit uPSI_uSysTools; TuO
294: 1444 unit upsi_neuralsets; mX4
```

```
1445 unit uPSI_uWinNT.pas mX4
Total of Function Calls: 34880
SHA1: of 4.7.6.10  CF939E3A8D4723DB1DEF383C5FC961E06728C58F
CRC32: 38F88218 30.5 MB (32,022,344 bytes)

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
  |              .MMMMMMMM##\                   `*"?HM
  |          ..  ,HMMMMMMMMMMMo\.                    |M
  |              |MMMMMMMMMMMMMMMMHo                 |M
  |               ?MMMMMMMMMMMMMMMM*                 |H
  |.               `#MMMMMMMMMMMMM'                .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)
```