

```
1 # -*- coding: utf-8 -*-
2
3 from datetime import datetime
4 import pytz
5 import os
6 from os import listdir
7 from os.path import isfile, join
8 import pathlib
9 import cw_file_utils as cw_fu
10 import pkg_resources
11 import pickle
12 import gzip
13
14 class Jarvis:
15     # lowercase are internal to class params
16     timezone = 'America/New_York'
17     starttime = datetime.now(pytz.timezone(timezone))
18     longDateFormat = "%d-%b-%Y, %H:%M"
19     shortTimeFormat = "%H:%M"
20     utility_files_dir = 'utility_files'
21     data_dir = 'data'
22     data_subdir1 = '01_original'
23     data_subdir2 = '02_working'
24     data_subdir3 = '03_train'
25     data_subdir4 = '04_test'
26     data_subdir5 = '05_experiments'
27
28     data_subdirs = [data_subdir1, data_subdir2,
29                     data_subdir3, data_subdir4, data_subdir5]
30
31     PROJECT_NAME = ''
32     ROOT_DIR = ''
33     ROOT_DATA_DIR = ''
34     DATA_DIR = ''
35     DATA_DIR_ORIG = ''
36     DATA_DIR_WORK = ''
37     DATA_DIR_TEST = ''
38     DATA_DIR_TRAIN = ''
```

```
38     WORKING_DIR = ''
39     UTILITY_DIR = ''
40
41
42     def __init__(self,
43                   ROOT_DIR,
44                   PROJECT_NAME
45                   ):
46
47         self.ROOT_DIR = ROOT_DIR
48         self.PROJECT_NAME = PROJECT_NAME
49         self.UTILITY_DIR = join(self.ROOT_DIR, self.
utility_files_dir)
50
51         self.startup()
52         #self.showenvironment()
53         self.environmentScan()
54         self.greeting()
55
56     def startup(self):
57         print("Wha...where am I?")
58         self.ROOT_DATA_DIR = join(self.ROOT_DIR, self.
data_dir)
59         self.DATA_DIR = join(self.ROOT_DATA_DIR, self.
PROJECT_NAME)
60         self.WORKING_DIR = join(self.ROOT_DIR, self.
PROJECT_NAME)
61         self.DATA_DIR_ORIG = join(self.DATA_DIR, self.
data_subdir1)
62         self.DATA_DIR_WORK = join(self.DATA_DIR, self.
data_subdir2)
63         self.DATA_DIR_TRAIN = join(self.DATA_DIR, self.
data_subdir3)
64         self.DATA_DIR_TEST = join(self.DATA_DIR, self.
data_subdir4)
65         self.DATA_DIR_EXP = join(self.DATA_DIR, self.
data_subdir5)
66         print("I am awake now.")
```

```

67
68     def whattimeisit(self):
69         now = datetime.now(pytz.timezone(self.timezone))
70         print("The current time is " + now.strftime(self
    .shortTimeFormat))
71
72     def showenvironment(self):
73         print("I am inspecting the local environment..."
    )
74         print('')
75         print("Your environment has been configured: ")
76         print("PROJECT_NAME:      " + self.PROJECT_NAME)
77         print("ROOT_DIR: " + self.ROOT_DIR)
78         print("WORKING_DIR:  " + self.WORKING_DIR)
79         print('')
80         print("ROOT_DATA_DIR: " + self.ROOT_DIR)
81         print("DATA_DIR:      " + self.DATA_DIR)
82         print("DATA_DIR_ORIG:" + self.DATA_DIR_ORIG)
83         print("DATA_DIR_WORK:" + self.DATA_DIR_WORK)
84         print("DATA_DIR_TRAIN:" + self.DATA_DIR_TRAIN)
85         print("DATA_DIR_TEST:" + self.DATA_DIR_TEST)
86         print('')
87         print("UTILITY_DIR:  " + self.UTILITY_DIR)
88
89     def displayProjects(self):
90         onlyDirs = [f for f in listdir(self.ROOT_DIR) if
    not isfile(join(self.ROOT_DIR, f))]
91         onlyDirs.sort()
92         print("Project listing:")
93         for x in onlyDirs:
94             print("--> " + x)
95
96     def environmentScan(self):
97         self.setupWorkingDir()
98         self.setupDataDir()
99
100         #self.showProjectWorkFiles()
101         #self.showProjectDataFiles()

```

```
102
103     # Set directory to WORKING_DIR
104     os.chdir(self.WORKING_DIR)
105     print("I have set your current working directory
to {0}".format(os.getcwd()))
106
107     def setupWorkingDir(self):
108         if not os.path.exists(self.WORKING_DIR):
109             # Create a new directory because it does not
exist
110             os.makedirs(self.WORKING_DIR)
111             print("The working directory is not present
. Directory created.")
112
113     def setupDataDir(self):
114         if not os.path.exists(self.DATA_DIR):
115             os.makedirs(self.DATA_DIR)
116             print("The project data directory is not
present. Directory created.")
117
118         for x in self.data_subdirs:
119             if not(os.path.exists(join(self.DATA_DIR, x
))) :
120                 os.makedirs(join(self.DATA_DIR, x))
121                 print("Data subdirectory " + x + " has
been created")
122
123         print('')
124
125     def greeting(self):
126         greeting = ''
127         x = int(self.startuptime.strftime("%H"))
128
129         if (x > 7) and (x <= 8):
130             greeting = 'An early morning I see.'
131         elif (x > 8) and (x <= 12):
132             greeting = 'Extra caffeine may help.'
133         elif (x > 12) and (x <= 18):
```

```
134         greeting = 'Reminder, no more coffee.'
135     elif (x > 18) and (x <= 20):
136         greeting = 'I hope you had dinner.'
137     elif (x > 20) and (x <= 24):
138         greeting = 'I see you are having a productive
evening.'
139     elif (x <= 7):
140         greeting = 'You should really be sleeping.'
141
142     self.whattimeisit()
143     print("Hello sir. " + greeting)
144     print('')
145
146     def showProjectDataFiles(self):
147         print("Here are all your project data files")
148         cw_fu.exploreDirectory(self.DATA_DIR)
149
150     def showProjectWorkFiles(self):
151         print("Here are all your project work files")
152         cw_fu.exploreDirectory(self.WORKING_DIR)
153
154     def showAllDataFiles(self):
155         print("Here are all your available data files")
156         cw_fu.exploreDirectory(self.ROOT_DATA_DIR)
157
158     def compressProjectDataFiles(self, removeOriginal=
False):
159         cw_fu.exploreDirectory(self.DATA_DIR, compress=
True, removeOriginal=removeOriginal)
160
161     def compressAllDataFiles(self, removeOriginal=False
):
162         cw_fu.exploreDirectory(self.ROOT_DATA_DIR,
compress=True, removeOriginal=removeOriginal)
163
164     def getPackageVersion(self, pkgName):
165         print(pkgName + " version: " + str(pkg_resources
.get_distribution(pkgName)))
```

```
166
167     def saveExperiment(self, dataExperiment, fileName,
168         fileExtension='.jexp'):
169         pickle.dump(dataExperiment, gzip.open(f'{self.
170             DATA_DIR_EXP}/{fileName}{fileExtension}.gz', 'wb'))
171
172     def loadExperiment(self, fileName, fileExtension='.
173         jexp'):
174         obj = pickle.load(gzip.open(f'{self.DATA_DIR_EXP
175            }/{fileName}{fileExtension}.gz', 'rb'))
176         return obj
```