

```
In [1]: import os
```

```
In [2]: os
```

```
Out[2]: <module 'os' from '/Users/admin/anaconda3/lib/python3.8/os.py'>
```

```
In [3]: def f(a,b):  
        '''  
        This is an amazing function  
        '''  
        return a+b
```

```
In [4]: f.__doc__
```

```
Out[4]: '\n    This is an amazing function\n    '
```

```
In [5]: print (f.__doc__)
```

```
    This is an amazing function
```

```
In [6]: 3 +4
```

```
Out[6]: 7
```

```
In [7]: int(3).__add__(4)
```

```
Out[7]: 7
```

```
In [9]: int(3).__add__ = lambda x,y : x-y
```

```
-----  
AttributeError                                Traceback (most recent call last)  
<ipython-input-9-6bb31bbfdefc> in <module>  
----> 1 int(3).__add__ = lambda x,y : x-y  
  
AttributeError: 'int' object attribute '__add__' is read-only
```

```
In [10]: a = [1,2,3]
```

```
In [11]: len(a)
```

```
Out[11]: 3
```

```
In [12]: a.append(6)
```

```
In [13]: a.append
```

```
Out[13]: <function list.append(object, /)>
```

```
In [36]: class Person:

        ADULT_AGE = 18

        def is_adult(self, ):
            return self.age>self.ADULT_AGE

        def set_age(self, age):
            self.age = age

        def get_age(self,):
            return self.age
```

```
In [37]: mitsos = Person()
```

```
In [27]: #mitsos.is_adult(50)
```

```
Out[27]: True
```

```
In [39]: mitsos.set_age(50)
```

```
In [40]: mitsos.get_age()
```

```
Out[40]: 50
```

```
In [41]: mitsos.is_adult()
```

```
Out[41]: True
```

```
In [42]: mitsos = Person()
        kostas = Person()
```

```
In [43]: mitsos.set_age(50)
        kostas.set_age(15)
```

```
In [44]: mitsos.is_adult()
```

```
Out[44]: True
```

```
In [45]: kostas.is_adult()
```

```
Out[45]: False
```

In [113...

```
class Person:

    ADULT_AGE = 18

    def __init__(self, name, age):
        self.set_age(age)
        self.name = name

    def __str__(self,):
        return f'Name: {self.name} Age: {self.age}'

    def __len__(self,):
        return 100

    def __add__(self, another_person):
        return Person(
            age=self.age + another_person.age,
            name = f'{self.name} {another_person.name}'
        )

    def is_adult(self, ):
        # All these three are equivalent
        #return self.age>self.ADULT_AGE

        #return self.is_adult_2(self.age)

        return Person.is_adult_2(self.age)

    @staticmethod
    def is_adult_2(age):
        return age>Person.ADULT_AGE

    def set_age(self, age):
        self.age = age

    def get_age(self,):
        return self.age
```

In [114...

```
Person.is_adult_2(40)
```

Out[114...

```
True
```

In [115...

```
import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.

If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

```
In [ ]: Protein.structure.length.visualize
```

```
In [47]: mitsos = Person()
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-47-8fa818a94c26> in <module>  
----> 1 mitsos = Person()
```

```
TypeError: __init__() missing 1 required positional argument: 'age'
```

```
In [48]: mitsos = Person(50)
```

```
In [52]: mitsos = Person(age=50)
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-52-a6bdf12f0997> in <module>  
----> 1 mitsos = Person(age=50)
```

```
TypeError: __init__() missing 1 required positional argument: 'name'
```

```
In [50]: mitsos.is_adult()
```

```
Out[50]: True
```

```
In [69]: mitsos = Person(age=50, name='Mitsos')
```

```
In [66]: print (mitsos)
```

```
Name: Mitsos Age: 50
```

```
In [70]: len(mitsos)
```

```
Out[70]: 100
```

In [149...

```
class Gene:

    def __init__(self, start, end):
        self.start = start
        self.end = end

    def __len__(self, ):
        return self.end - self.start

    def __iter__(self,):
        self.i = self.start
        return self

    def __next__(self):
        r = self.i
        self.i += 1

        if self.i > self.end:
            raise StopIteration

        return r

    def __str__(self,):
        return (f'This is a gene: start: {self.start} end: {self.end}')
```

```
    def __getitem__(self, index):
        r = self.start + index
        if r > self.end:
            raise IndexError
        return r
```

In [158...

```
class Protein(Gene):

    def __init__(self, start, end, name):
        super().__init__(start, end)
        self.name = name

    def __str__(self,):
        return (f'This is a protein: start: {self.start} end: {self.end}')
```

```
    def __len__(self, ):
        return (self.end - self.start)//3
```

In [152...

```
class New_Class(Protein):
    pass
```

In [153...

```
class Amazing_Class(Gene, Person):
    pass
```

In [155...

```
p = Amazing_Class(100, 150)
```

In [157...

```
Amazing_Class.is_adult_2(30)
```

Out[157... True

In [160...

```
P1 = Protein(100,160, 'P1')
APOE = Gene(1000, 1500)
```

```
In [145... print (len(P1))  
print (len(APOE))
```

```
20  
500
```

```
In [ ]:
```

```
In [146... print (P1)
```

```
This is a protein: start: 100 end: 160
```

```
In [151... print (APOE)
```

```
This is a gene: start: 1000 end: 1500
```

```
In [123... len(P1)
```

```
Out[123... 150
```

```
In [105... APOE = Gene(1000, 1500)
```

```
In [96]: for i in APOE:  
        print (i)
```

```
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037
```

1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101

1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165

1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229

1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293

1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357

1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421

1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485

1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498

In [73]: len(APOE)

Out[73]: 500

In [75]: mitsos = Person(age=50, name='Mitsos')
kostas = Person(age=15, name='Kwstas')

In [77]: new_person = mitsos + kostas

In [78]: print (new_person)

Name: Mitsos Kwstas Age: 65

In [79]: for i in mitsos:
print (i)

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-79-f24fb0255b90> in <module>  
----> 1 for i in mitsos:  
      2     print (i)
```

TypeError: 'Person' object is not iterable

In [106... APOE[100]

Out[106... 1100

In [107... APOE[1000]

```
-----  
IndexError                                Traceback (most recent call last)  
<ipython-input-107-0bea4233ac89> in <module>  
----> 1 APOE[1000]  
  
<ipython-input-104-3e543d6f2487> in __getitem__(self, index)  
    24         r = self.start + index  
    25         if r > self.end:  
----> 26             raise IndexError  
    27         return r
```

IndexError:

In [103... a = [1,2,3]
a[100]

```
-----  
IndexError                                Traceback (most recent call last)  
<ipython-input-103-7e439a244abe> in <module>
```

```
1 a = [1,2,3]
----> 2 a[100]
```

IndexError: list index out of range

```
In [108... dir(mitsos)
```

```
Out[108... ['ADULT_AGE',
 '__add__',
 '__class__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__len__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__weakref__',
 'age',
 'get_age',
 'is_adult',
 'name',
 'set_age']
```

API calls

```
In [161... import requests, sys

server = "https://rest.ensembl.org"
ext = "/vep/human/hgvs/ENSP00000401091.1:p.Tyr124Cys?"

r = requests.get(server+ext, headers={ "Content-Type" : "application/json"})

if not r.ok:
    r.raise_for_status()
    #sys.exit()

decoded = r.json()
#print(repr(decoded))
```

```
In [163... #decoded
```

```
In [164... import requests, sys

server = "https://rest.ensembl.org"
ext = "/sequence/region/human/X:1000000..1000100:1?"

r = requests.get(server+ext, headers={ "Content-Type" : "text/plain"})

if not r.ok:
    r.raise_for_status()
    #sys.exit()

print(r.text)

CTGTAGAAACATTAGCCTGGCTAACAAAGGTGAAACCCCATCTCTACTAACAAATACAAAATATTGGTTGGGCGTGG
TGGCGGGTGCTTGTAATCCCAGCTAC
```

```
In [165... import pandas as pd
```

```
In [166... dfs = pd.read_html('https://en.wikipedia.org/wiki/List_of_busiest_airports_
```

```
In [167... len(dfs)
```

Out[167... 8

```
In [168... dfs.__len__()
```

Out[168... 8

```
In [171... dfs[2]
```

	Rank	Airport	Location	Country	Code(IATA/ICAO)	Totalpassengers
0	1.0	Hartsfield–Jackson Atlanta International Airport	Atlanta, Georgia	United States	ATL/KATL	107394029
1	2.0	Beijing Capital International Airport	Chaoyang-Shunyi, Beijing	China	PEK/ZBAA	100983290
2	3.0	Dubai International Airport	Garhoud, Dubai	United Arab Emirates	DXB/OMDB	89149387
3	4.0	Los Angeles International Airport	Los Angeles, California	United States	LAX/KLAX	87534384
4	5.0	Tokyo Haneda Airport	Ōta, Tokyo	Japan	HND/RJTT	87502720
5	6.0	O'Hare International Airport	Chicago, Illinois	United States	ORD/KORD	83339186
6	7.0	Heathrow Airport	Hillingdon, London	United Kingdom	LHR/EGLL	80126320
7	8.0	Hong Kong International Airport	Chek Lap Kok, Islands, New Territories	Hong Kong SAR, China	HKG/VHHH	74517402

	Rank	Airport	Location	Country	Code(IATA/ICAO)	Totalpassengers
8	9.0	Shanghai Pudong International Airport	Pudong, Shanghai	China	PVG/ZSPD	74006331
9	10.0	Charles de Gaulle Airport	Roissy-en-France, Île-de-France	France	CDG/LFPG	72229723
10	11.0	Amsterdam Airport Schiphol	Haarlemmermeer, North Holland	Netherlands	AMS/EHAM	71053147
11	12.0	Indira Gandhi International Airport	Delhi	India	DEL/VIDP	69900938
12	13.0	Guangzhou Baiyun International Airport	Baiyun-Huadu, Guangzhou, Guangdong	China	CAN/ZGGG	69769497
13	14.0	Frankfurt Airport	Frankfurt, Hesse	Germany	FRA/EDDF	69510269
14	15.0	Dallas/Fort Worth International Airport	Dallas-Fort Worth, Texas	United States	DFW/KDFW	69112607
15	16.0	Seoul Incheon International Airport	Incheon	South Korea	ICN/RKSI	68350784
16	17.0	Istanbul Atatürk Airport	Yeşilköy, Istanbul	Turkey	IST/LTBA	68192683
17	18.0	Soekarno-Hatta International Airport	Tangerang, Banten	Indonesia	CGK/WIII	66908159
18	19.0	Singapore Changi Airport	Changi, East Region	Singapore	SIN/WSSS	65628000
19	20.0	Denver International Airport	Denver, Colorado	United States	DEN/KDEN	64494613
20	21.0	Suvarnabhumi Airport	Bang Phli, Samut Prakan	Thailand	BKK/VTBS	63378923
21	22.0	John F. Kennedy International Airport	Queens, New York, New York	United States	JFK/KJFK	61623756
22	23.0	Kuala Lumpur International Airport	Sepang, Selangor	Malaysia	KUL/WMKK	60013397
23	24.0	Madrid Barajas Airport	Barajas, Madrid	Spain	MAD/LEMD	57862951
24	25.0	San Francisco International Airport	San Mateo County, California	United States	SFO/KSFO	57708196

	Rank	Airport	Location	Country	Code(IATA/ICAO)	Totalpassengers
25	26.0	Chengdu Shuangliu International Airport	Shuangliu-Wuhou, Chengdu, Sichuan	China	CTU/ZUUU	52950529
26	27.0	Barcelona–El Prat Airport	Barcelona	Spain	BCN/LEBL	50148228
27	28.0	Chhatrapati Shivaji Maharaj International Airport	Mumbai, Maharashtra	India	BOM/VABB	49876769
28	29.0	McCarran International Airport	Las Vegas, Nevada	United States	LAS/KLAS	49863090
29	30.0	Seattle–Tacoma International Airport	SeaTac, Washington	United States	SEA/KSEA	49849520
30	31.0	Toronto Pearson International Airport	Mississauga, Ontario	Canada	YYZ/CYYZ	49467097
31	32.0	Shenzhen Bao'an International Airport	Bao'an, Shenzhen, Guangdong	China	SZX/ZGSZ	49348950
32	33.0	Mexico City International Airport	Venustiano Carranza, Mexico City	Mexico	MEX/MMMX	47700834
33	34.0	Orlando International Airport	Orlando, Florida	United States	MCO/KMCO	47694573
34	35.0	Kunming Changshui International Airport	Guandu, Kunming, Yunnan	China	KMG/ZPPP	47215986
35	36.0	Taiwan Taoyuan International Airport	Dayuan, Taoyuan	Taiwan	TPE/RCTP	46535180
36	37.0	Charlotte Douglas International Airport	Charlotte, North Carolina	United States	CLT/KCLT	46446721
37	38.0	London Gatwick Airport	Crawley, West Sussex	United Kingdom	LGW/EGKK	46432630
38	39.0	Munich Airport	Freising, Bavaria	Germany	MUC/EDDM	46253623
39	40.0	Newark Liberty International Airport	Newark, New Jersey	United States	EWR/KEWR	46065175
40	41.0	Sheremetyevo International Airport	Khimki, Moscow Oblast	Russia	SVO/UUEE	45836255

	Rank	Airport	Location	Country	Code(IATA/ICAO)	Totalpassengers
41	42.0	Miami International Airport	Miami-Dade County, Florida	United States	MIA/KMIA	45044312
42	43.0	Phoenix Sky Harbor International Airport	Phoenix, Arizona	United States	PHX/KPHX	44943686
43	44.0	Xi'an Xianyang International Airport	Weicheng, Xianyang, Shaanxi	China	XIY/ZLXY	44653927
44	45.0	Ninoy Aquino International Airport	Pasay/Parañaque, Metro Manila	Philippines	MNL/RPLL	44488321
45	46.0	Sydney Kingsford-Smith Airport	Mascot, New South Wales	Australia	SYD/YSSY	44475976
46	47.0	George Bush Intercontinental Airport	Houston, Texas	United States	IAH/KIAH	43807539
		Shanghai Hongqiao International Airport	Changning-			

In []: