In [43]:
```
class Human():
    pass
```

In [44]:
```
alex = Human()
```

In [50]:
```
alex.name = 'Alex'
```

In [51]:
```
alex.l = [1,2,3,4]
```

In [53]:
```
alex.name
```

Out[53]:
```
'Alex'
```

In [283]:
```
class Human():

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

    def set_age(self, age):
        if age<0 or age>120:
            print ('Invalid values')
        else:
            self.age = age

    def set_name(self, name):
        if not name[0].upper() == name[0]:
            print ('Invalid value')

        else:
            self.name=name

    @staticmethod
    def is_adult(age):
        return age>=18

    def __str__(self):
        return self.name + ' ' + str(self.age)

    def __len__(self):
        return self.age
```

In [155]:
```
class Student(Human):

    def __init__(self, age, name, grade):
        #self.set_age(age)
        #self.set_name(name)
        super().__init__(age, name)
        self.set_grade(grade)

    def set_grade(self, grade):
        if grade<0 or grade>20:
            print ('Invalid value')
        else:
            self.grade = grade

    def __str__(self):
        return super().__str__() + ' ' + str(self.grade)
```

In [158]:
```
alex = Student(50, 'Alex', 15)
```

In [160]:
```
mitsos = Human(40, 'Mitsos')
```

In [161]:
```
print (alex)
```

```
Alex 50 15
```

In [162]:
```
print (mitsos)
```

```
Mitsos 40
```

In [ ]:

In [ ]:

In [157]:
```
print (alex)
```

```
Alex 50 15
```

In [128]:
```
alex.set_grade(15)
```

In [130]:
```
alex.grade
```

Out[130]:
```
15
```

In [ ]:

In [122]:
```
Human.is_adult(50)
```

Out[122]:
```
True
```

In [ ]:

In [115]:
```
alex = Human()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-115-12a27f13d5ce> in <module>()
----> 1 alex = Human()

TypeError: __init__() missing 2 required positional arguments: 'age' and 'name
'
```

In [116]:
```
alex = Human(age=50, name='Alex')
```

In [117]:
```
print (alex)
```

```
Alex 50
```

In [111]:
```
print (alex)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-111-5659e036b518> in <module>()
----> 1 print (alex)

<ipython-input-96-cff36c6d4e05> in __str__(self)
     17
     18     def __str__(self):
---> 19         return self.name + ' ' + str(self.age)
     20
     21     def __len__(self):

AttributeError: 'Human' object has no attribute 'name'
```

In [107]:
```
alex.set_age(50)
alex.set_name('Alex')
```

In [108]:
```
print (alex)
```

```
Alex 50
```

In [102]:
```
len(alex)
```

Out[102]:
```
50
```

In [ ]:
```

```

Τα παρακάτω είναι ισοδύναμα!

In [77]:
```
alex.set_age(10)
```

In [72]:
```
Human.set_age(alex, 50)
```

In [73]:
```
alex.age
```

Out[73]:
```
50
```

In [78]:
```
alex.is_adult()
```

Out[78]:
```
False
```

In [66]:
```
alex.set_age(500)
```

```
Invalid values
```

In [70]:
```
Human.set_age(50)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-70-0b860501b493> in <module>()
----> 1 Human.set_age(50)

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

In [57]:
```
alex.age
```

Out[57]:
```
50
```

In [285]:
```
l = [Human(x+10, 'Mitsos') for x in range(30)]
```

In [286]:
```
for x in l:
    x.set_age(20)
    print (x)
```

```
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
Mitsos 20
```

In [287]:
```
l
```

Out[287]:
```
[<__main__.Human at 0x102b58be0>,
 <__main__.Human at 0x102b58ba8>,
 <__main__.Human at 0x102b58898>,
 <__main__.Human at 0x102b58a58>,
 <__main__.Human at 0x102b58ac8>,
 <__main__.Human at 0x102b58b00>,
 <__main__.Human at 0x102b58a90>,
 <__main__.Human at 0x102b58908>,
 <__main__.Human at 0x102b58940>,
 <__main__.Human at 0x102b58978>,
 <__main__.Human at 0x102b589e8>,
 <__main__.Human at 0x102b58a20>,
 <__main__.Human at 0x102b589b0>,
 <__main__.Human at 0x102b587b8>,
 <__main__.Human at 0x102b587f0>,
 <__main__.Human at 0x102b58828>,
 <__main__.Human at 0x102b58cf8>,
 <__main__.Human at 0x102b58e80>,
 <__main__.Human at 0x102b58da0>,
 <__main__.Human at 0x102b58dd8>,
 <__main__.Human at 0x102b58e48>,
 <__main__.Human at 0x102b58e10>,
 <__main__.Human at 0x102b58d30>,
 <__main__.Human at 0x102b58d68>,
 <__main__.Human at 0x102b58eb8>,
 <__main__.Human at 0x102b58f60>,
 <__main__.Human at 0x102b58f98>,
 <__main__.Human at 0x102b58fd0>,
 <__main__.Human at 0x102b58ef0>,
 <__main__.Human at 0x102b58f28>]
```

Exceptions

In [169]:
```
def f(x,y):
    try:
        r = x/y
    except ZeroDivisionError:
        print ('Cannot divide with 0')
        return None

    return r
```

In [181]:
```
import os
def count_lines(fn):

    #if not os.path.exists(fn):
    #    print ('Does not exist')
    #    return

    with open(fn) as f:
        return len(f.readlines())
```

In [174]:
```
%%bash

cat > t.txt << EOF
asdfa
sdf
asdf
asd
fas
EOF
```

In [175]:
```
! cat t.txt
```

```
asdfa
sdf
asdf
asd
fas
```

In [179]:
```
count_lines('t.txt')
```

Out[179]: 5

In [183]:
```
try:
    count_lines('t2.txt')
except FileNotFoundError:
    print ('Den yparxei')
```

```
Den yparxei
```

In [188]:
```
try:
    #1
    sdfs
except NameError:
    print ('Den yparxei h metablhth')
```

```
Den yparxei h metablhth
```

In [198]:
```
try:
    #asdfasdf
    #open('asdfasdfasdf.asdassd')
    a=1/0
    #a = [1,2,3]
    #a[8]
except NameError:
    print ('Den yparxei metablhth')
except FileNotFoundError:
    print ('Den bre8hke to arxeio')
except Exception:
    print ('Kapoio allo la8os exei ginei')
```

```
Kapoio allo la8os exei ginei
```

In [203]:
```
try:
    sdfasdfasd
except Exception:
    pass
```

In [208]:
```
try:
    #sdfasdfasd
    #a=1/0
    open('sdasdasd')
except Exception as mitsos:
    print (mitsos)
print ('hello')
```

```
[Errno 2] No such file or directory: 'sdasdasd'
hello
```

In [ ]:
```

```

In [201]:
```
asdfasdf
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-201-c534d63b7f80> in <module>()
      3
      4
----> 5 asdfasdf
      6
      7

NameError: name 'asdfasdf' is not defined
```

In [172]:
```
f(10,0)
```

```
Cannot divide with 0
```

In [218]:
```
def is_adult(age):
    if age<0 or age>120:
        #print ('invalid age')
        #return
        raise Exception('Invalid age')
    return age>=18
```

In [221]:
```
is_adult(-10)
```

```
---------------------------------------------------------------------------
Exception                                 Traceback (most recent call last)
<ipython-input-221-fd663797513f> in <module>()
----> 1 is_adult(-10)

<ipython-input-218-d70a556f26a4> in is_adult(age)
      3         #print ('invalid age')
      4         #return
----> 5         raise Exception('Invalid age')
      6     return age>=18

Exception: Invalid age
```

In [222]:
```
try:
    is_adult(888)
except Exception as e:
    print (e)
```

```
Invalid age
```

In [223]:
```
try:
    raise Exception('LATHOS!')
except Exception as e:
    print (e)
```

```
LATHOS!
```

In [225]:
```
try:
    a=1/0
except Exception as e:
    print (e)
else:
    print ('Division is ok!')
```

```
division by zero
```

In [226]:
```
try:
    a=1/2
except Exception as e:
    print (e)
else:
    print ('Division is ok!')
```

```
Division is ok!
```

In [229]:
```
try:
    a=1/2
except Exception as e:
    print (e)
else:
    print ('Division is ok!')
finally:
    print ('PANTA MPAINW EDW!')
```

```
Division is ok!
PANTA MPAINW EDW!
```

In [231]:
```
try:
    a=1/0
except Exception as e:
    print (e)
else:
    print ('Division is ok!')
finally:
    # Clean up code
    print ('PANTA MPAINW EDW!')
```

```
division by zero
PANTA MPAINW EDW!
```

In [232]:
```
raise Exception('adsasdsdf')
```

```
---------------------------------------------------------------------------
Exception                                 Traceback (most recent call last)
<ipython-input-232-2ca76ea3560d> in <module>()
----> 1 raise Exception('adsasdsdf')

Exception: adsasdsdf
```

In [234]:
```
raise FileNotFoundError('Den to brika')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-234-0475ec9b0bf1> in <module>()
----> 1 raise FileNotFoundError('Den to brika')

FileNotFoundError: Den to brika
```

In [235]:
```
class MyFaboulousException(Exception):
    pass
```

In [236]:
```
raise MyFaboulousException('something went wrong')
```

```
---------------------------------------------------------------------------
MyFaboulousException                      Traceback (most recent call last)
<ipython-input-236-a4a2cce99d0e> in <module>()
----> 1 raise MyFaboulousException('something went wrong')

MyFaboulousException: something went wrong
```

Serialization

In [261]:
```
a = [
    1,
    {'a': 1, 'b': 2},
    {'c': [4,5,6,7,8]}
]
```

In [253]:
```
type(a)
```

Out[253]:
```
list
```

In [239]:
```
import json
```

In [241]:
```
s = json.dumps(a)
```

In [244]:
```
type(s)
```

Out[244]:
```
str
```

In [245]:
```
with open('t.json', 'w') as f:
    f.write(s)
```

In [247]:
```
!cat t.json
```

```
[1, {"a": 1, "b": 2}, {"c": [4, 5, 6, 7, 8]}]
```

In [248]:
```
n = json.loads(s)
```

In [251]:
```
type(n)
```

Out[251]:
```
list
```

In [254]:
```
a= '''
zcxvz
dvzx
cv
zxcv
zxc
vzx
cvz
xcv
zxc
vzx
cv
zxcv
zxc
vzx
cv
xcv
xczcvzxcvzxcvzxcv
'''
```

In [256]:
```
print (a)
```

```
zcxvz
dvzx
cv
zxcv
zxc
vzx
cvz
xcv
zxc
vzx
cv
zxcv
zxc
vzx
cv
xcv
xczcvzxcvzxcvzxcv
```

In [265]:
```
print (json.dumps(a, indent=2))
```

```
[
  1,
  {
    "a": 1,
    "b": 2
  },
  {
    "c": [
      4,
      5,
      6,
      7,
      8
    ]
  }
]
```

In [266]:
```
!cat t.json
```

```
[1, {"a": 1, "b": 2}, {"c": [4, 5, 6, 7, 8]}]
```

In [267]:
```
with open('t.json') as f:
    s = f.read()
d = json.loads(s)
d
```

Out[267]:
```
[1, {'a': 1, 'b': 2}, {'c': [4, 5, 6, 7, 8]}]
```

In [268]:
```
with open('t.json') as f:
    d = json.load(f)
```

In [270]:
```
json.dumps([1,2, alex])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-270-f536ee83b1a6> in <module>()
----> 1 json.dumps([1,2, alex])

~/anaconda3/lib/python3.7/json/__init__.py in dumps(obj, skipkeys, ensure_asci
i, check_circular, allow_nan, cls, indent, separators, default, sort_keys, **k
w)
    229         cls is None and indent is None and separators is None and
    230         default is None and not sort_keys and not kw):
--> 231         return _default_encoder.encode(obj)
    232     if cls is None:
    233         cls = JSONEncoder

~/anaconda3/lib/python3.7/json/encoder.py in encode(self, o)
    197         # exceptions aren't as detailed.  The list call should be roug
hly
    198         # equivalent to the PySequence_Fast that ''.join() would do.
--> 199         chunks = self.iterencode(o, _one_shot=True)
    200         if not isinstance(chunks, (list, tuple)):
    201             chunks = list(chunks)

~/anaconda3/lib/python3.7/json/encoder.py in iterencode(self, o, _one_shot)
    255                 self.key_separator, self.item_separator, self.sort_key
s,
    256                 self.skipkeys, _one_shot)
--> 257         return _iterencode(o, 0)
    258
    259 def _make_iterencode(markers, _default, _encoder, _indent, _floatstr,

~/anaconda3/lib/python3.7/json/encoder.py in default(self, o)
    177
    178         """
--> 179         raise TypeError(f'Object of type {o.__class__.__name__} '
    180                         f'is not JSON serializable')
    181

TypeError: Object of type Student is not JSON serializable
```

In [271]:
```
import pickle
```

In [273]:
```
s = pickle.dumps([1,2, alex])
```

In [274]:
```
d = pickle.loads(s)
```

In [275]:
```
d
```

Out[275]:
```
[1, 2, <__main__.Student at 0x102b58588>]
```

In [277]:
```
d[2].age
```

Out[277]:
```
50
```

In [278]:
```
def Human():


    def setter(self, age):
        self._age = age
```

In [ ]:
```

```