1 EAFP

It's Easier to Ask Forgiveness then Permission

```
def quack_and_fly(thing):
    try:
thing.quack()
thing.fly()
    except AttributeError as e:
print(e)
```

2 Python OOP

2.1

class		instance
varible	X	variable
selfsiz		selfli
class.$_{dict}$_ _		instance.$_{dict}$_ _
classname.var		self.var

2.2

isinstance(x, y) issubclass(x, y)

2.3 @classmethod instance metodlarında otomatik olarak ilk parametre instance olur. Bu decorator ile class gelir. def set$_{raiseamt}$(cls, amount):

cls.raise$_{amt}$ = amount

2.4 @staticmethod

Sınıfla ilgili ancak instance ya da class referansına ihtiyaç duymayan metodlar.

(ÖRNEK)

3 Property Decorators

```
def decorator_function(orginal_function):
    def wrapper_function():
print('Wrapper executed this before {}'.format(original_function.
return orginal_function()
    return wrapper_function


@decorator_function
def display():
    print('display function ran')
```

Yukarıdaki decorator tanımlaması aşağıdaki gibi generik hale getirilebilir. Aşağıda anlamadığım bir nokta var.

```
def decorator_function(orginal_function):
    def wrapper_function(*args, **kwargs):
print('Wrapper executed this before {}'.format(original_function.
return orginal_function(*args, **kwargs)
    return wrapper_function


@decorator_function
def display():
    print('display function ran')


@property def fullname(self):
```

```
    return '{} {}'.format(self.first, self.last)

@fullname.setter def fullname(self, name):
    first, last = name.split(' ')
    self.first = first self.last = last

 @fullname.deleter def fullname(self, name):
     print('Delete name!')
     self.first = None self.last = None del emp1.fullname
```

4 Functools

5 Context Managers

Resides in contextlib module

5.1 with (resource)

5.2 ignored (exception)

5.3 lock (threading)

6 Magic Methods (<u>dunder</u>)

Method	Explanation
`__init__`	
`__repr__`	
`__str__`	
`__add__`	
`__len__`	
`__iter__`	
`__next__`	
`__reversed__`	
`__getattr__`	
`__setattr__`	
`__delattr__`	
vs	

7 try/except

```
try:
    f = open('file.txt')
except FileNotFoundError as e:
    print(e)
except Exception as e:
    print(e)
else:
```

```
    print(f.read)
    f.close()
finally:
    print('Executing Finally...')
```

8   Collections

8.1   namedTuple

```
from collections import namedtuple
>>>Point = namedtuple('Point', ['x', 'y'], verbose=True)
>>>p = Point(11, y=22)
>>>p[0]+p[1]
  33
>>>p.x + p.y
  33
>>>x, y = p
>>>x + y
  33
```

8.2   deque (double ended queue)

```
from collections import deque
d = deque('ghi')
for elem in d:
    print(elem.upper())
G
H
I
d.append('j')
d.appendleft('R')
```

- append/appendleft

- pop/popleft

8.3   chainMap

Best way to merge dictionary without altering source dictionaries
from collections import ChainMap

EAFP It's Easier to Ask Forgiveness then Permission #+BEGIN_SRC python def quack_and_fly(thing): try: thing.quack() thing.fly() except AttributeError as e: print(e) #+END_SRC		