<https://www.cnblogs.com/wyb666/p/9857678.html>

class TestDict:

name = "wyb"

age = "21"

def \_\_init\_\_(self):

self.gender = 'male'

def keys(self): # 获取字典的键

s = ('name', 'age', 'gender')

return s

def \_\_getitem\_\_(self, item): # 获取键对应的值

return getattr(self, item) # getattr获取对象下某个属性的值

o = TestDict()

print(dict(o)) # 创建字典 -> 先调用对象下的keys方法再用o["xxx"]获取值([]本质上是调用对象下的\_\_getitem\_\_方法)

class Base(object):

def \_\_init\_\_(self):

self.fields = []

def \_\_getitem\_\_(self, item):

return getattr(self, item)

def keys(self):

return self.fields

def hide(self, \*keys): # 隐藏域

for key in keys:

self.fields.remove(key)

return self

def append(self, \*keys): # 添加域

for key in keys:

self.fields.append(key)

return self

class Sx(Base):

name = "wyb"

age = 21

def \_\_init\_\_(self):

super(Base, self).\_\_init\_\_()

self.gender = 'male'

# self.fields定义默认输出字段

self.fields = ["name", "age"]

o1 = Sx()

print(dict(o1))

o2 = Sx()

print(dict(o2.hide("age")))

o3 = Sx()

print(dict(o3.append("gender")))

# 输出结果:

# {'name': 'wyb', 'age': 21}

# {'name': 'wyb'}

# {'name': 'wyb', 'age': 21, 'gender': 'male'}