person = {'name': 'John Doe', 'age': 23}

print("My name is {name} and I am {age} years old.".format(\*\*person))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import requests

r = requests.get('https://xkcd.com/353/')

print(help(r))  # help() returns detailed explanation of the object

print(dir(r))   # dir() returns list of methods & attrs associated to object

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

my\_set = {10, 20, 10, 20, 30, 40, 50, 40}

print(my\_set)

# {40, 10, 50, 20, 30}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

~~range vs xrange~~

~~xrange behaves more like a generator than what range does.~~

~~xrange yields one result at a time.~~

~~range put all result in memory.~~

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

~~Iteritems() gives us all result at a time.~~

~~items() put all result in memory.~~

~~For key, val in my\_dict.items():~~

~~Print(key, val)~~

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

class Person:

    def introduce(self):

        print('My name is John')

class English(Person):

    def introduce(self):

        super(English, self).introduce()

        print('... and I am 23 years old.')

p1 = English()

p1.introduce()

# My name is John

# ... and I am 23 years old.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

class Person:

    def \_\_init\_\_(self, name, age):

        self.fullname = name

        self.age = age

    def introduce(self):

        print(f'My name is {self.fullname}')

class English(Person):

    def \_\_init\_\_(self, name, age, position):

        super(English, self).\_\_init\_\_(name, age)

        self.position = position

    def introduce(self):

        super(English, self).introduce()

        print(f'... I am {self.age} years old and I am {self.position}.')

p1 = English('John Doe', 23, 'CEO')

p1.introduce()

# My name is John Doe

# ... I am 23 years old and I am CEO.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_