

— 选择飞天，重启未来 —

Week 8 Asm 知识梳理 – INFO1110

by
李指导

FEIT EDUCATION

飞天教育

 FEIT 小助手 【USYD】 
澳大利亚 悉尼



知识复习

Function

本质上，就是一个可以接收参数的，独立写出去的代码块

格式：

```
def function_name( parameter1,parameter2 ):  
    # code  
    # if you do not want to have any code,  
    # add a line like the below  
    pass
```

如果需要利用 **function** 的运算结果，则可以 **return** 运算结果

例子

```
def minus( number1,number2 ):  
    result = number1-number2  
    return result
```

自定义 **function** 的时候需要注意的点：

如果没有 **return** 值，最后会默认 **return None**

单独写一个 **return** 等同于没有 **return** 值

写完 **return** 后 **function** 将会直接结束

格式上，**def** 以及最后的冒号是必须的

Pass 指的是什么也不做，所以如果要写一个空 **function** 的话可以这么做

同文件中的两个 **function** 不能重名！

What is a function and what are the features of a function prototype?

Function is a block of code
that are written outside the main function
and
can be executed through other code invoking its function name.

Function prototype includes the function name and parameters that the function takes in.

What happens to the variables and calculations in the function when it finishes?

If the variables were created in this function and are never returned as a result of the function, they will be removed from the program's memory.

How does a function deal with errors that it cannot handle?

It will throw an exception at the line where the error occurs, or the error may be dealt by another try except block that has is containing the function.

Describe what happens to the values of the arguments when calling a function foo(3, someObject)

Both arguments will be passed into the function, and will be correspondingly modified by the function if any modifications were made.

x = 3

y = x

y += 1

Class

- a) 本质上就是其他 data type 的集合体
- b) 一个 class 必须要有自己的名字
- c) 可以写在其他文件中，但使用时需要 import 另一个 python 文件的名字
- d) Class 中其他类型的变量可以有无数个
- e) 每个变量需要有自己的名字，且只能出现一次

Class 实例

```
class Student:  
  
    #self denotes that the object is doing something to itself  
    def __init__(self):  
        # things to do when new object is made  
        #the class now will have a variable called age  
        #this age variable can be used everywhere in the class  
        #and this variable should have a unique name  
  
        self.age = 18  
  
    def print_age(self):  
        print(self.age)
```

实例中的 init 学名叫做 **constructor**，用来代表

“新创建一个本 class 的 object 时， object 需要初始化的值”

在以上例子中， init 的作用为：

1. 为这个属于 Student class 的 object 创建了一个名为 age 的 variable
2. 赋予了这个 age variable 一个最初的值为 18
3. 能够让 class 代码块内部随意使用这个 age variable

需要注意的是，每当新建一个某 class 的变量时，init function 会无条件触发，所以需要注意不要做一些过多的事情

Class 可以写在其他文件里，但是记得要正确使用 import 来使用在其他文件中的代码

Constructor 中也可以提供更多的 parameter，用来初始化 object

实例中的 class 中有一个 function，需要使用的话则需要手动触发

Class 使用实例：

```
boy = new Student()  
boy.print_age()
```

What is a [default] constructor?

If we did not explicitly make a __init__ function by ourselves, the default constructor is what is being used to initialize the instances of that class.

What are instance variables?

Instance variables are variables that belong to that certain class, and each instance of the class will have such variables along with them.

How are instance variables accessed?

Either through a dot (some_var.some_attribute) or a function (get_some_attribute()'s return value)

What is the meaning of self?

Self refers to the instance itself, meaning functions within the class having self as its first parameter may use the instance variables of the class.

Class 例題

1. Write a class that is called Computer, and give the class the following attributes and functions

1. A variable that represents the name of the computer
2. A list that contains the users registered in the computer
3. A function to print the number of users that the computer registered
4. A function to print the name of the computer

2. Write a class that is called Calculator, and give the class the following attributes

- a) A function which has two parameters that allows for addition
- b) A function which has two parameters that allows for subtraction
- c) A function which has two parameters that allows for multiplication
- d) A function which has two parameters that allows for division
- e) A function to print the number of calculations the object had made
- f) A function which prints the greatest value the calculator has ever calculated
- g) A function that returns the lowest value the calculator has ever calculated

Object

How do you create an Object?

Knowing the class of the object to be created, normally there are two ways.

1. A = 1.5, an object of type float is created
2. my_car = Car(...), the ... denotes all necessary parameters needed

How to use the constructor for an object

It is not necessary to use the constructor explicitly, the constructor is called automatically by default when the object is created.

Methods

Write a method to read the instance variable value and return it

Write a method to initialise certain instance variables as default values defined by the description

Design a method that would change the state of an object, instance variables, based on a given description.

```
1 class Adventurer:  
2     def __init__(self):  
3         self.inventory = []  
4         self.skill = 5  
5  
6     def get_inv(self):  
7         return self.inventory  
8  
9     def add_skill(self):  
10        self.skill += 1
```

Assignment 拆解

这个 assignment 是基于两个大的 class 的基础上，我们来进行一些“模拟手机”功能的填写。

在这份占分 35% 的作业当中，分数分布为：

20% - 通过 ed 上可以看到细节的 public test

10% - 通过 ed 上看不到细节的 hidden test

5% - 进行 report 的编写

其中，最后提交的文件中必须包括：

- dorrigo_mobile.py
- dorrigo_contact.py
- report.md

具体提交地点为在 Ed 上即可，目前默认多次 submit 并不会带来分数上的扣除。

针对两个.py 文件，我们可以逐个击破来分析各个 function 所需要我们做的事情：

Dorrigo Mobile

说明部分

我们在这次 assignment 当中需要模拟完成一个手机的操作系统，而其中我们需要包括的几项核心功能为：

能够在开机/关机状态之间切换

能够显示电池剩余电量

能够显示已联网/未联网状态

联网时的信号强度

每个手机应当有一个对应的 owner，owner 信息应当以 dorrigo_contact 的形式储存

而有关于 `owner` 的信息，我们的 `owner` 又应该包含以下信息：

`First name`

`Last name`

`Phone number`

一组可存储的联系人信息，初始时为空白（`None`）

以上信息的储存形式为我们另一个文件中 `dorrigo_contact` 的形式存在，而我们存储的每个 `dorrigo_contact` 除了要包含 `owner` 的信息之外，我们还需要包含和这个联系人最多二十条的聊天记录，也就是说我们不仅要储存联系人的信息，还要储存一系列 `string`。

在 `assignment` 的规划当中，我们的 `phone` 应该支持以下功能（支持的形式为有对应的 `method` 在 `class` 中被 `implement`）：

开机

充电

换电池

消耗 k 格电量

查找联系人

删除联系人

添加联系人

除此之外，还有一些“common sense”类的逻辑我们需要去额外注意：

如果手机是关着的，那说明他没有联网

如果手机没有联网，那么手机就没有信号强度

电池的电量范围为 0~100，100 为满格，0 为空格

信号强度的范围为 0~5，5 为最佳强度，0 为无强度

代码拆解

```
from dorrigo_contact import dorrigo_contact
```

from import.... 这一行的意思是能够让我们使用其他文件中的 python 代码，所以如果去掉了这一行我们就无法使用另一个文件当中的 class 和 method 等任何资源

```
class dorrigo_mobile:

    def __init__(self, max_contacts):
        """
        Every phone manufactured has the following attributes

        the phone is off
        the phone has battery life 25
        the phone is not connected
        the phone has signal strength 0
        Each of the contacts stored in the array contacts has a
None value

        the owner first name "Dorrigo"
        the owner last name is "Incorporated"
        the owner phone number is "180076237867"
        the owner chat message should have only one message
        "Thank you for choosing Dorrigo products"
        """

```

开头一行我们声明我们将要开始对 class 进行填充，同时 class 名为 dorrigo_mobile

回忆我们之前讲过的，

__init__(self,...)

这个 `method` 的意思是为我们 `class` 的初始化提供一个起始点，也就是说：

我们新建一个这个 `class` 的 `object` 时，`__init__` 会为了这个新建的 `object` 自动执行一遍

在 `__init__` 中我们可以声明这个 `class` 中可以被使用到的 `attribute`，并进行初始化

我们可以通过 `self` 以外接进来的参数，来对我们的 `class` 进行针对性的设置

在这里，我们如果翻看 `assignment` 的要求，在 `dorrigo_mobile` 中，我们需要添加的参数包括：

开机/关机状态

电池剩余电量

已联网/未联网状态

联网时的信号强度

对应的 `owner`, `owner` 的信息以 `dorrigo_contact` 的形式储存

这里大家可以自行思考一下，除了 `owner` 已经说明了对应的类型以外，各个 `attribute` 最好以什么数据类型来储存

```
def get_copy_of_owner_contact(self):
    """returns a copy of the owner contact details. Returns a
datatype dorrigo_contact.
    return None if the phone is off
    """
```

如果我们没有通电的话，那么我们这里直接 `return None`

反之，我们直接 `return` 我们存储在手机当中的 `owner dorrigo_contact` 即可

```
def add_contact(self, contact):
    """ only works if phone is on
    will add the contact in the array only if there is space
    and does not exist
    The method will find an element that is None and set it
    to be the contact.
    returns True if successful
    """

```

我们的通讯录当中如果还有空位（具体有多少个位置在最开始新建 `object` 的时候我们已经接进来了参数）那么我们就 在我们的通讯录当中添加上这个给进来的 `contact`

在几个情况下，我们拒绝添加：

手机关机

没有空间

已经在 `contact` 中

如果添加成功，我们返回 `True`，不然我们返回 `None`

```
def remove_contact(self, contact):
    """ only works if phone is on
    find the dorrigo_contact object and set the array element
    to None
    return True on successful remove
    """

```

在手机关机的情况下，我们直接中止这个 `method` 的运行，不进行任何操作

在开机的情况下，我们如果在通讯录中能够找到参数所给进来的 `contact`，那么我们就从我们的通讯录当中移除对应的 `contact`，并把该项设置为 `None`

```
def get_number_of_contacts(self):
    """ only works if phone is on
    returns the number of contacts, or -1 if phone is off
    """
```

在手机关机的情况下，我们直接中止这个 method 的运行，返回-1

反之的话我们返回目前已经存储的非 None 联系人的数量

```
def search_contact(self, name):
    """ only works if phone is on
    returns a list of all contacts that match first name OR
last name
    if phone is off, or no results, None is returned
    """
```

在手机关机的情况下，我们直接中止这个 method 的运行，返回 None

如果我们的通讯录当中，没有任何一个联系人的姓名和给进来的参数，name，相同的话，那么我们 return None

不然的话，我们返回所有姓或者名为 name 的 dorrigo_contact，以一个 list 作为整体返回

```
def is_phone_on(self):
    """ returns True if phone is on
    """
```

返回我们的手机开机状态，无需其他操作

```
def set_phone_on(self, on):
    """
        set the on status based on the boolean input
        when phone turns on, it costs 5 battery for startup.
network is initially disconnected
        when phone turns off it costs 0 battery, network is
disconnected
        always return True if turning off
        return False if do not have enough battery level to
turn on
        return True otherwise
    """

```

将我们的手机设置为开机/不开机，具体状态取决于给进来的参数，`on` 的值

开机需要消耗 5 的电量，并且开机时联网状态为“未联网”

我们在关机时不需要消耗任何电量，并且同样会设置为未联网模式

如果我们没有足够的电量去开机的话，那么我们返回 `False`

其他任何的情况下我们都去返回 `True`

```
def get_battery_life(self):
    """
        Return the battery life level as an integer. if the
phone is off, zero is returned.
    """

```

这里返回我们的剩余电量

如果我们是关机的状态，那么我们返回 0

```
def change_battery(self, new_battery_level):
    """ Change battery of phone.
    On success. The phone is off and new battery level adjusted
    and returns True
    If new_battery_level is outside manufacturer specification
    of [0,100], then
        no changes occur and returns False.
    """

```

这里我们需要模拟一个“换电池”的行为

如果更换电池成功，我们默认我们的手机开关机状态为关机，并且将电池状态调整为参数中 `new_battery_level` 所给定的值

如果值不在 0~100 的范围之内，那么我们返回 `False` 并且默认这次更换没有进行

反之的话我们进行所有操作，并返回 `True`

```
def is_connected_network(self):
    """ only works if phone is on.
    returns True if the phone is connected to the network
    """

```

我们直接返回我们是否联网的状态，如果关机的话则 `return false`

```
def disconnect_network(self):
    """ only works if phone is on.
    when disconnecting, the signal strength becomes zero
    always returns True
    """

```

只在开机的情况下进行任何改动，否则 return false

如果是开机的状态，那我们改变我们的信号强度为 0

```
def connect_network(self):
    """ only works if phone is on.
    Connect to network
    if already connected do nothing and return True
    if connecting:
        1) signal strength is set to 1 if it was 0
        2) signal strength will be the previous value if it is
not zero
        3) it will cost 2 battery life to do so
    returns the network connected status
    """

```

只在开机的情况下进行任何改动，否则 return false

如果我们已经联网，则不进行任何改动并 return true

如果之前的网络强度是 0，那现在设置为 1

连接网络需要耗费 2 格电

完成改动后，返回我们目前的信号强度

```
def get_signal_strength(self):  
    """ only works if phone is on.  
    returns a value in range [1,5] if connected to network  
    otherwise returns 0  
    """
```

只在开机的情况下进行任何改动，否则 return 0

其他情况下返回目前的信号强度

```
def set_signal_strength(self, new_strength):  
    """ only works if phone is on.  
    sets the signal strength and may change the network  
    connection status to on or off  
    signal of 0 disconnects network  
    signal [1,5] can connect to network if not already  
    connected  
    if the signal is set outside the range [0,5], nothing will  
    occur and will return False  
    returns True on success  
    """
```

只在开机的情况下进行任何改动，否则 return false

我们根据给进来的参数 new_strength 来进行设置，并且有可能将网络进行链接或断开

```
def charge_phone(self):
    """ each charge increases battery level by 10
    the phone has overcharge protection and cannot exceed 100
    returns True if the phone was charged by 10, otherwise
False
    """

```

我们进行充电的操作，增加 10 格的电量

如果充电过后点亮超过 100，那么我们 return False，除此之外 return True

```
def use_phone(self, k):
    """ Use the phone which costs k units of battery life.
    if the activity exceeds the battery life, the battery
automatically
    becomes zero and the phone turns off.
    returns True on successful use (not partial)

```

我们对手机进行 k 格电的消耗工作

如果我们消耗的电量使得电量低于 0，那么我们的电量自动变为 0 并关机

如果成功的使用了所有个 k 格电量，return True

否则 return False

悉大最强期末复习

□ 理工科

课程	Tutor	时长	原价	Week 7-8 预报名价
MATH1021	熊老师	6小时	240	169
MATH1023	涂老师	8小时	320	229
MATH1005	Daniel老师	7小时	280	199
MATH1004	Yvette老师	6.5小时	260	179
MATH1064	Nicole老师	4小时	160	119
MATH2070	Sarah老师	6小时	240	169
INFO1110	李指导	5小时	200	149
INFO1112	Shirley老师	6小时	240	169
INFO1113	Shirley老师	6小时	240	169
ELEC1601	Jeff老师	6小时	240	169
DATA1001	Cyan老师	5小时	200	149
DATA1002	Jeff老师	6小时	240	169
DATA2002	李指导	4小时	160	119
COMP2022	黄老师	6小时	240	169
ISYS2120	金老师	5小时	200	149
CHEM1111	Abby老师	6小时	240	169
GEOL1501	陈老师	6小时	240	179
CIVL1810	Winnie老师	6小时	240	179
CIVL2410	Kenny老师	8小时	320	239
CIVL2611	Fiona指导	8小时	320	239
CIVL2812	Tian老师	6小时	240	179
CIVL3811	程老师	6小时	240	179
CIVL3206	G老师	6小时	240	179

□ 文商科

课程	Tutor	时长	原价	Week 7-8 预报名价
BUSS1020	Wendy老师	6小时	220	159
BUSS1030	Coco老师	4小时	150	109
ECON1001	Lia老师	6小时	240	169
ECON1002	Lia老师	7.5小时	300	209
ECON1003	Nicole老师	7小时	250	169
ECMT1010	Nicole老师	7小时	250	169
PHIL1012	李老师	6小时	220	159
JPNS1611	蒋老师	4小时	150	109



转发海报集10赞，每科仅需预付30刀

即锁定期末课程最低价，近百元优惠

多门连报，每多一科立减20刀

多人同行，每人每科再减10刀