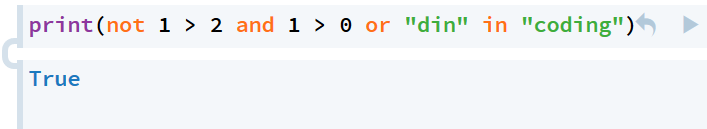
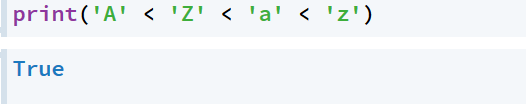
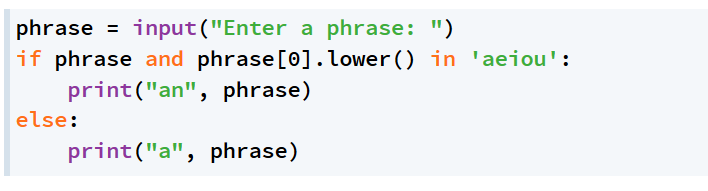
Worksheet 3 Conditionals



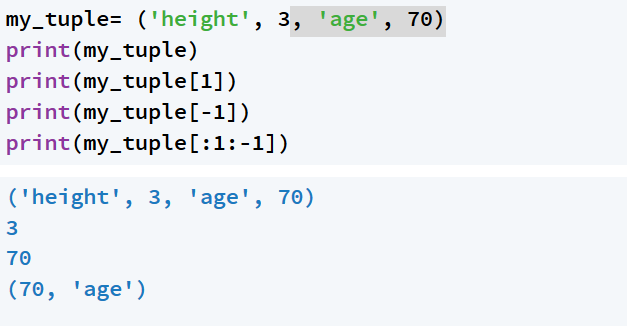
(including in) > not > and > or.

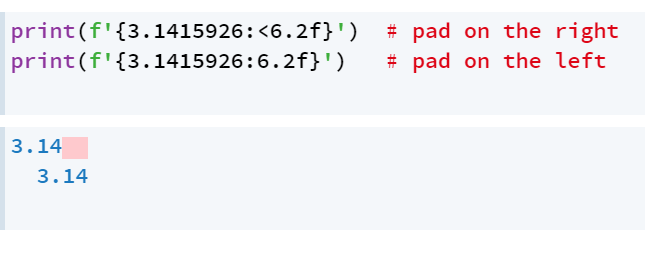


大写>小写



确认string 非空以及首字母是元音



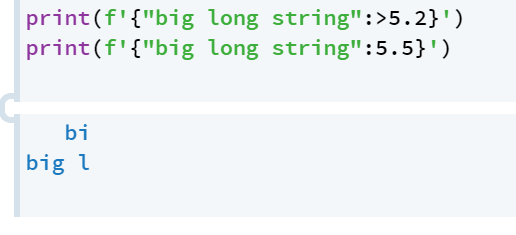


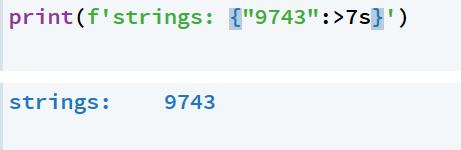
< 左对齐

* 右对齐

6.2f 总长6个字符 精度2位小数

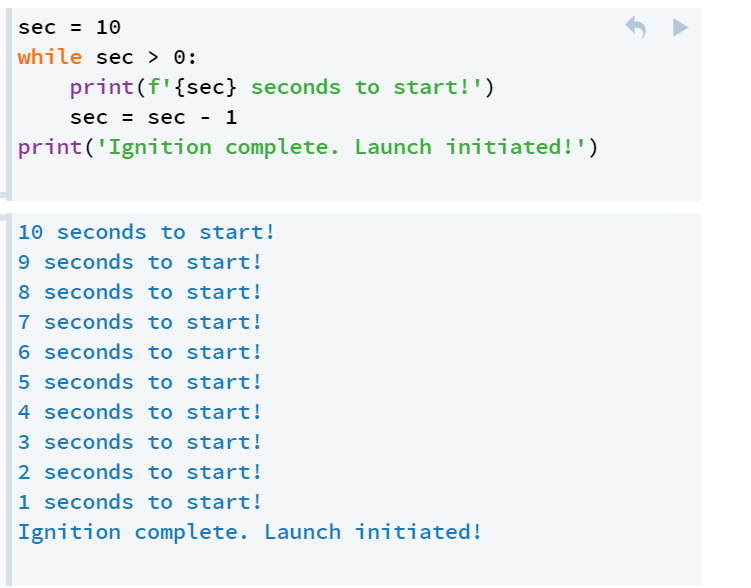
02d 字符长为2 位数不够用零补



* 5.2: print 字符串中两位 总共占5个字符 print2 个字符
* 
* >7s: 右对齐 总长

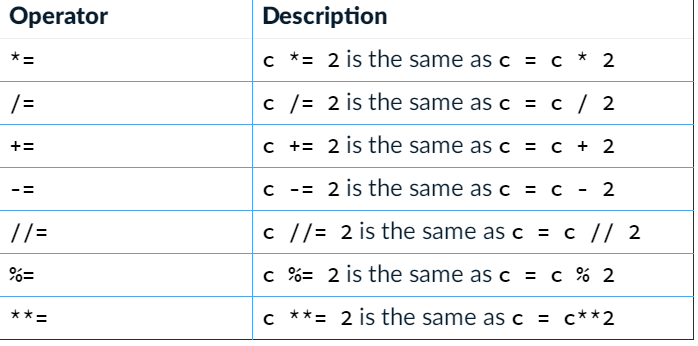
**Worksheet 8**

While loop

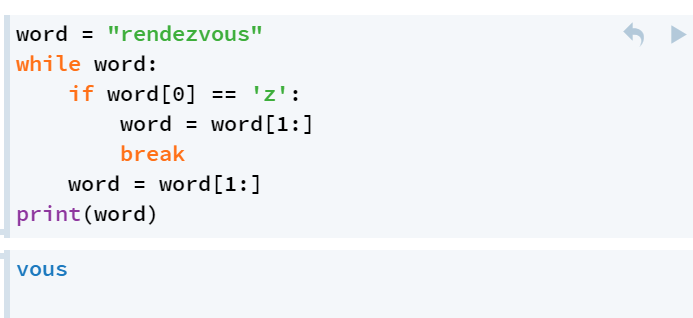


当sec > 0时 while loop evaluate它为TRUE

只要是True While loop 就一直执行下面的循环

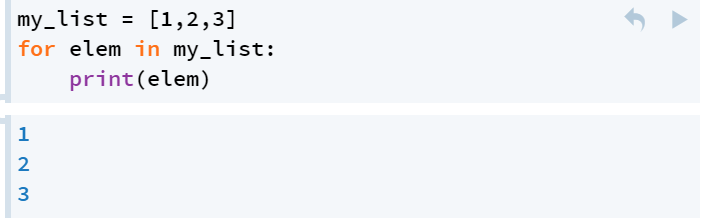


break



遇到break 自动跳出循环

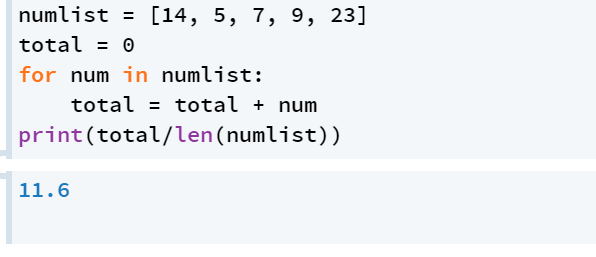
For loop





将列表中的元素依次赋值给variable（任意什么名字）

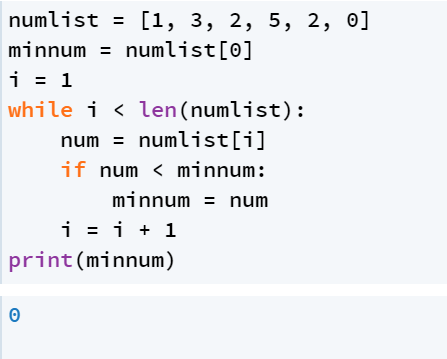
计算一个List中所有element 的和



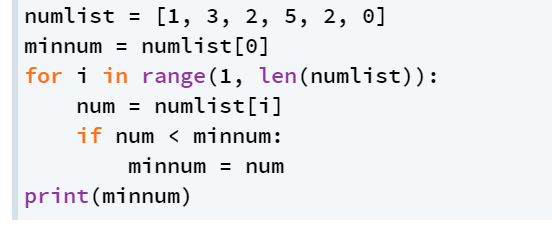
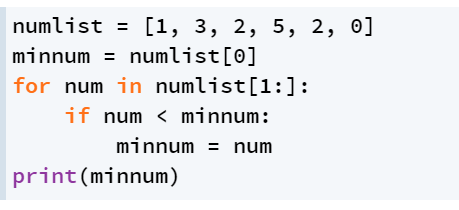
找出一个List中最小的element

将numlist[1:] 中所有element 依次与 minnum 比较 如果小于minnum, update it

While loop



For loop

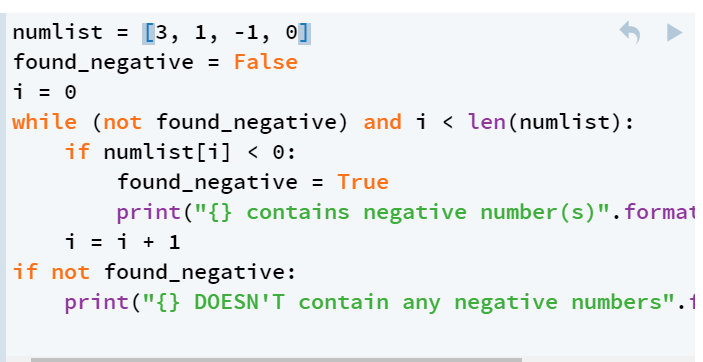
 

绝大部分情况for loop 更简洁， 但是当遇到不需要执行所有循环的情况下，

While loop 更简洁

Eg. Check列表中是否出现negative number

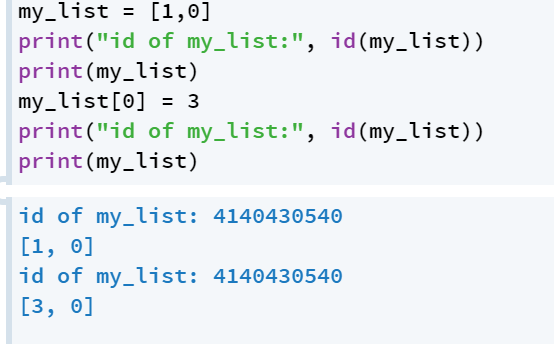
遇到negative 就跳出循环



**Worksheet 9**

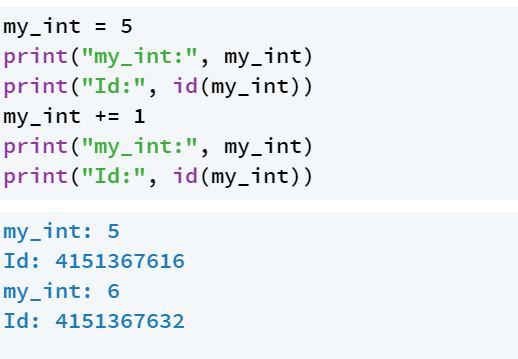
Mutability

Mutate : ID可以变 Value 不变



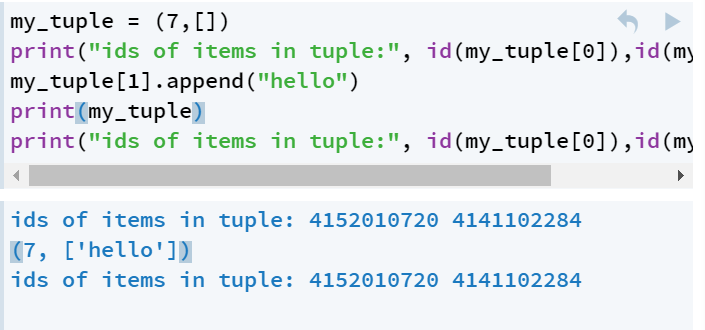
Immutable: 只要ID不变 Value 就不变

整数和tuple是immutable



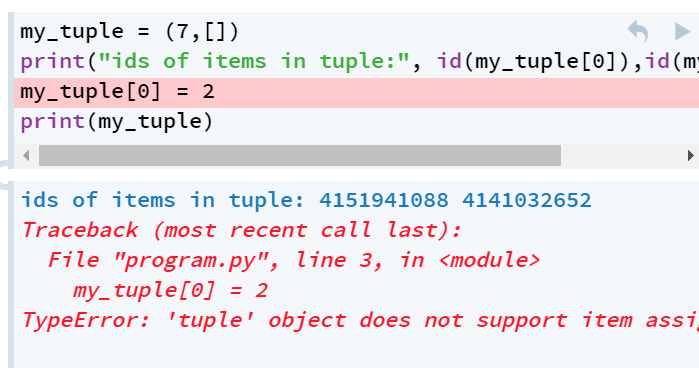
如果integer change 它的ID也一起change

Tuple里面如果consist of list( mutable 的值) List 里面的value 是可以变的



原因 当list 里的element 改变 list 里的ID不变 不影响tuple

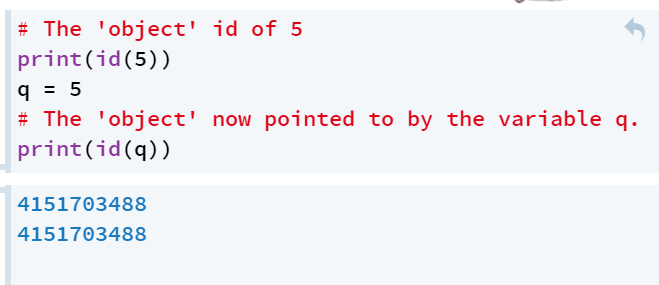
但不能直接改变tuple



Mutability and assignment

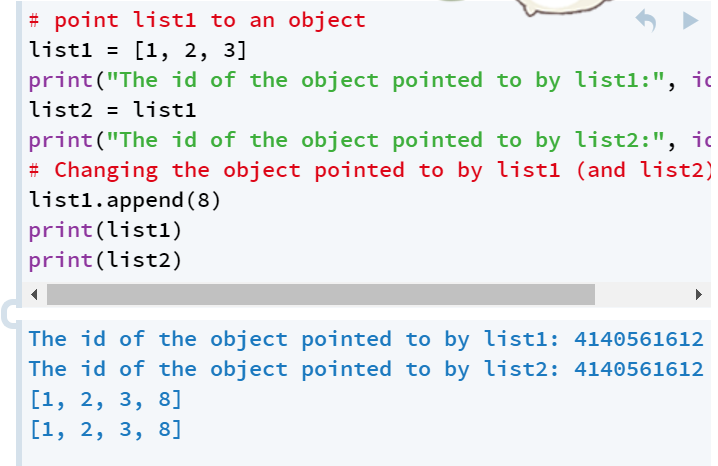
Assignment (=) 将variable point to an object

ID 不变

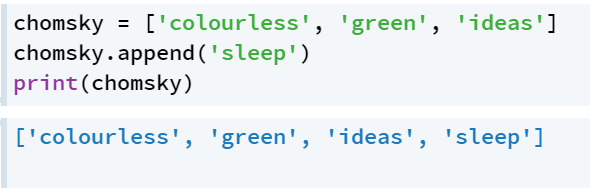


如果两个variable 同时指向一个object

如果一个variable 的值 改变其他也要变（因为只有一个object）

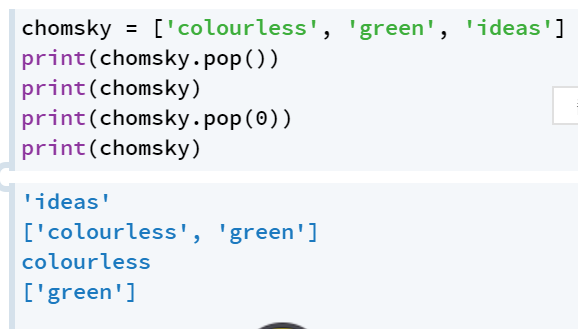


.**append()** 在末位加 element

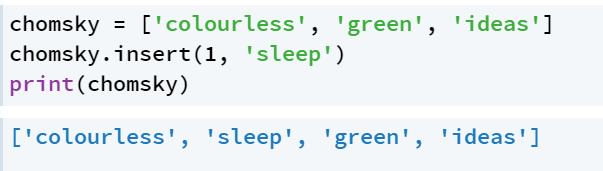


.pop() 默认：删去末位元素 将此元素store在.pop()中

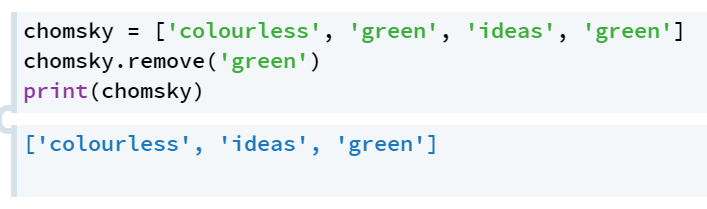
.pop(n) 删去第n 位元素



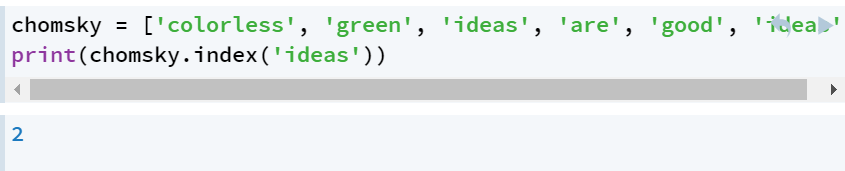
**.insert()** 在第n 位insert 某元素

****

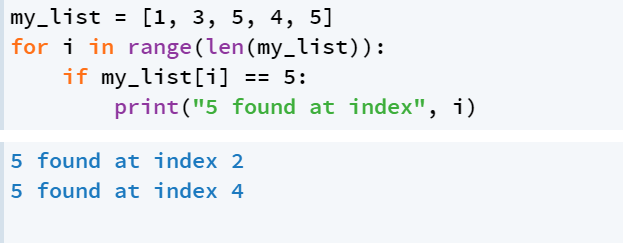
**.remove()** 删去某元素

****

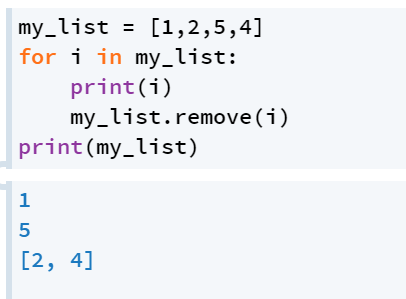
**.index()** 查找某元素的index

****

查找某一元素的index

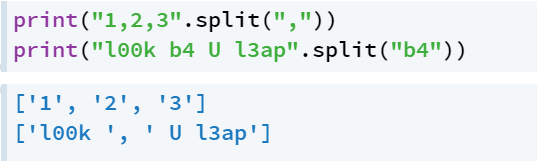
****

**.remove()** 将某元素删去

****

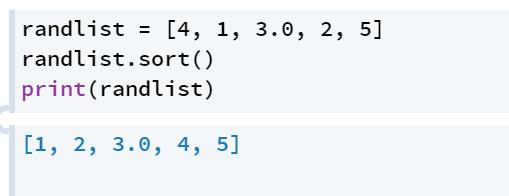
**.split()** 默认以whitespace 为分隔符（space tab 空行）

空号内添加以之为分隔符的内容

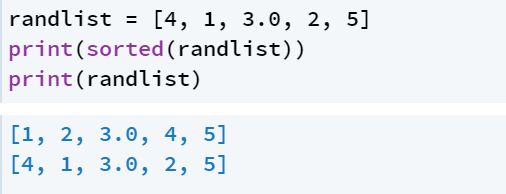
****

**.sort() VS .sorted()**

.sort 原来的list 顺序改变

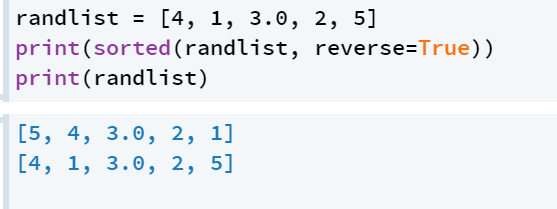


Sorted() 原来的list 顺序不改变



当处理倒序时

Sorted(item, reverse = True)

****

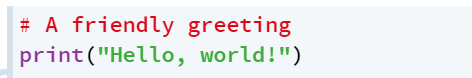
.sort()

item.sort(reverse = True)

**Worksheet 10**

Comment

After the **#** everything else on that line of code will be considered part of the comment and ignored by the computer.



**When to comment**

That is, you should group related lines of code together into one or more 'steps',

each taking a significant step towards solving the problem.

Then, every step should have a comment to help a reader identify and understand it. （将几行code组成一个step, 用comment 标注出来）

**What to say**

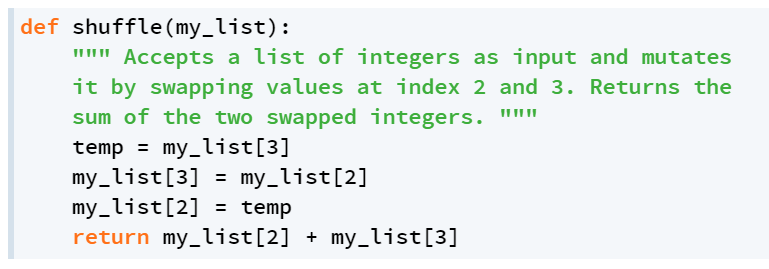
解释这么做的原因

Docstring ‘’’

(1) what the inputs to the function are;

(2) what the function does;

(3) what the return value is.



Name of variable

Meaningful, not generic. It makes the **purpose of the variable** and its value in the program, or the **action performed by the function** immediately clear. 有意义

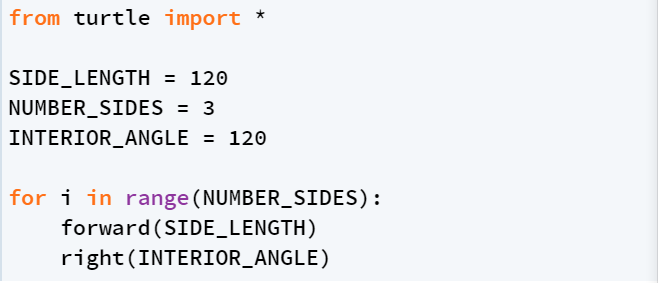
Obvious, not obscure. Avoid making your own abbreviations; the **meaning of the name** should be immediately clear. 清楚不含糊

Importantly, you **should not** use one-letter variable or function names, apart from conventions understood by all programmers  慎用one-letter variable

Magic numbers

 use variables and functions as a way to explain code where we otherwise would have no named variables.





将数字赋值给variable 是背后的含义明了易懂

* Line length must not exceed 79 characters.
* There should be no spaces between function/method names and arguments (also for slices and indexing) ie. write **myfunc(arg)** instead of **myfunc (arg)**
* Use blank lines between logically separate blocks of code (e.g. between functions and major sections associated with different comments).
* Avoid multiple statements on the same line
* Always start a new line after a statement containing **if, elif, else, while, for**, etc.
* Include spaces between arithmetic operators, i.e. write **2 \* 1** instead of **2\*1**
* To distinguish them from standard assignment, do not include spaces between (default) parameter variables in the function definition, i.e write **def my\_function(myvar=1):**instead of **def my\_function(myvar = 1)**
* Use 4 spaces for indentation. (Grok does this by default. If you're having problems, check your user settings.)
* Don't use tab characters in your code. (Again, Grok is your friend here, in that it will automatically translate any tab-based indentation into spaces.)

Commenting:

* Keep the indentation level of any comments the same as the line of code it is associated with.
* Do not state the obvious in comments.
* Do not use docstrings (**'''these green comments'''**) for anything other than function documentation. Use the inline comments (**# the red ones**) for everything else.

**Week 11 Dictionaries and sets**



A collection of keys and values

输入key 索引出value

Indexing dictionaries



1 一个key 只能在此dictionary 里出现一次

2 每个Key只有一个对应的value

Dictionary methods

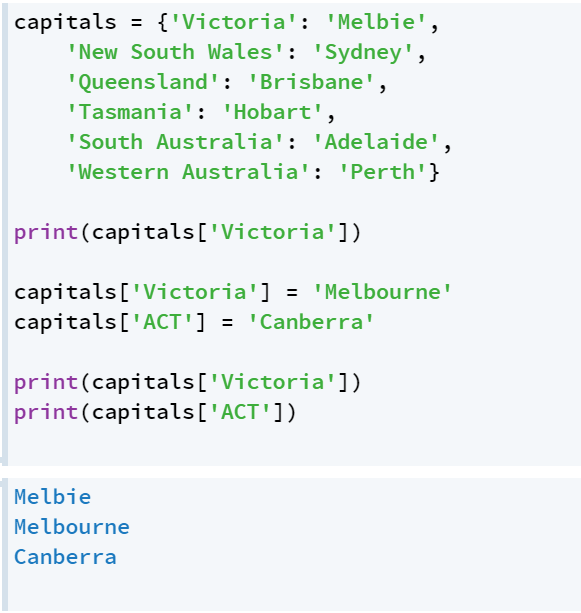
The **.get()** method takes a key as an argument and returns the value associated with it

The **.pop()** method takes a key as an argument and returns the value, also **deleting** that key-value pair.

Finally, the **.clear()** method deletes the entire contents of a dictionary



Updating dictionary

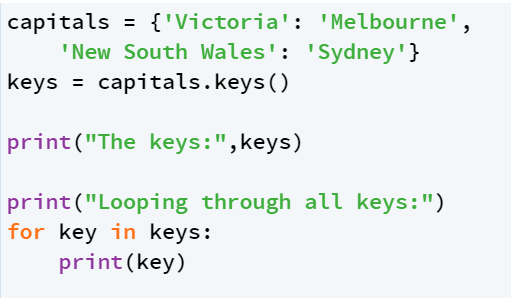


Testing Dictionary Membership



用if in 检测输入对象是否在dictionary中

Accessing all keys



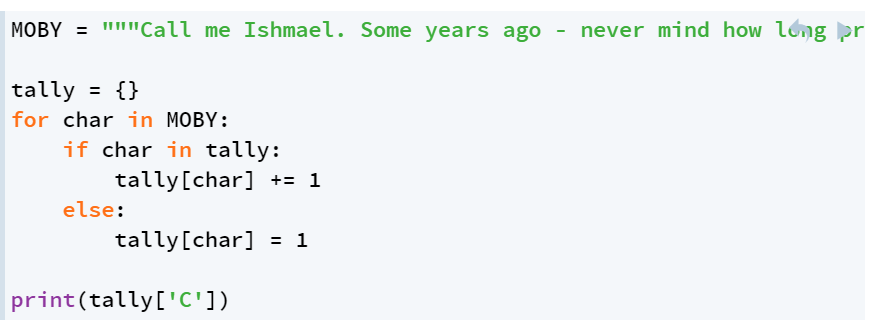
**.keys()** returns a special iterable collection called a **dict\_keys**

 supports iteration and the **in** operator like a **list**,

**not** support indexing(keys[0]是错的)

Counting Things with dictionaries

Count 一个dictionary 中某个character 出现的次数



作用： key 句子里每一个字母

Value 字母对应出现的次数

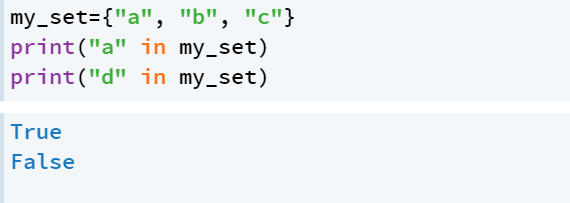


找出大写字母出现的次数

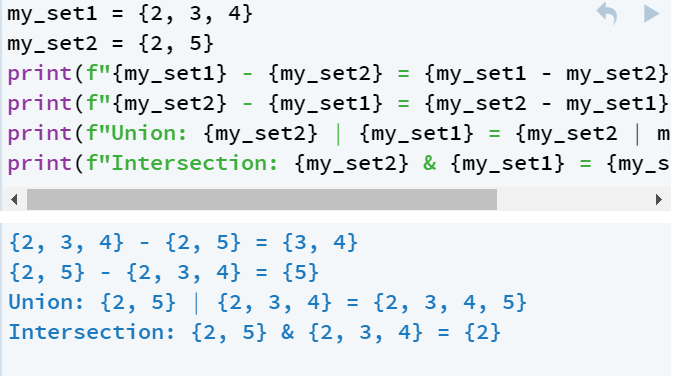
.set()

将dictionary 中的Key提出来， 去掉重复的key, 组成的List

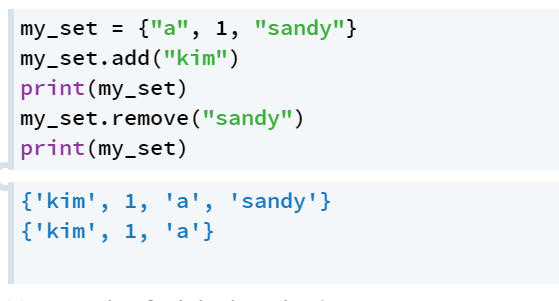
这种list 没有顺序（每一次调用顺序都不一样）



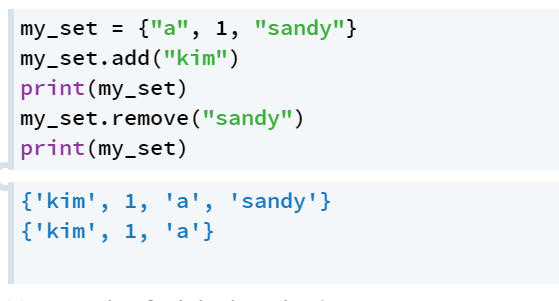
检查元素是否在set里



Set的运算 差 并 交



原set 加或者减另一个set



Calculate set 中元素的数量

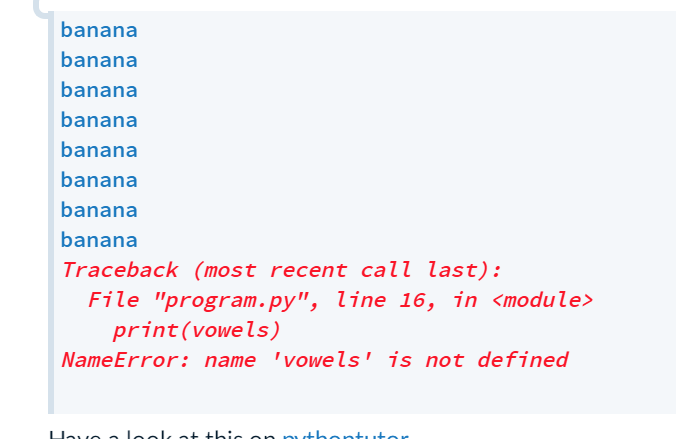
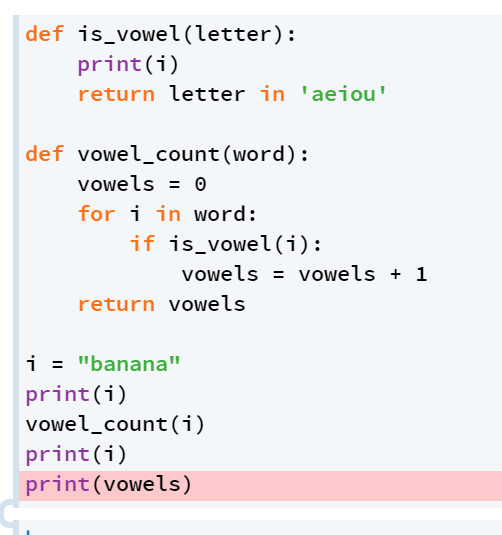
Worksheet 12

在函数内的variable are created in local **namespace**

**是不能accessible的**



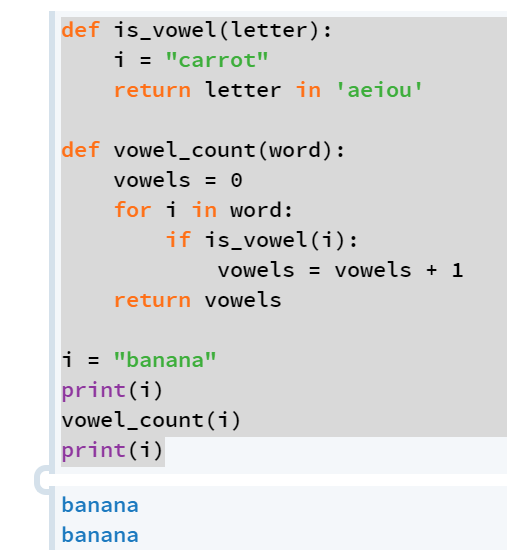
在上述函数中 函数里的i 和 vowels 无法被调用出来



Is\_vowel()是vowel\_count的子函数

当Python在function里遇到了一个variable(i) 他会先根据Local namespace（此处为print(i)） dereference it

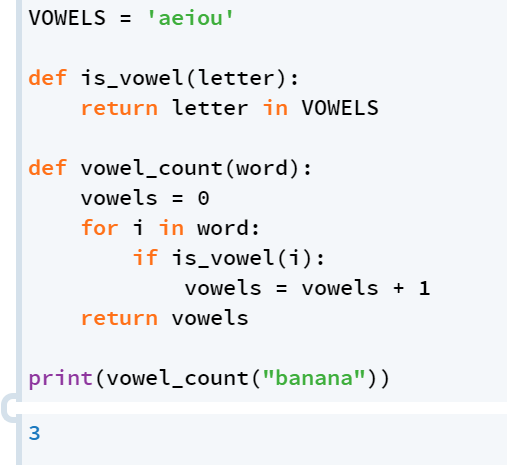
行不通再到globle namespace 里



不要将global variable 写在函数里

此处Print（i）不知道是“banana” 还是is\_vowel 里的carrot

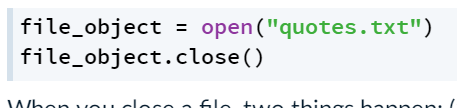
可以给function中可能要用到的parameter 加global variable(如此处的VOWELS)



Library里的Key不能是正数或者float?

Worksheet 14

Opening files



默认只能read 不能修改



“read” mode 同上

‘w’” write mode 如果文件存在 删除源文件 该文件代替源文件

如果文件不存在 新建一个文件

‘a’ append mode 保留源文件 将新的内容附在最后

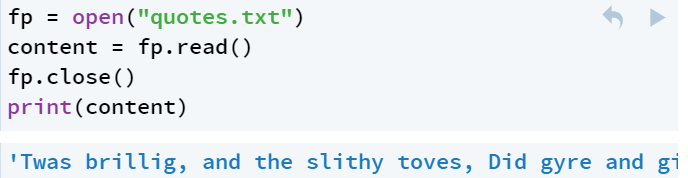
如果文件不存在 新建一个文件

Open() VS .close()

Open() 是函数 需要打开一个文件

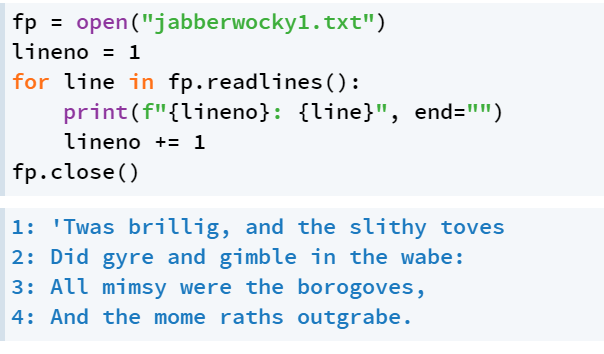
.close() 是method argument 为该文件

Reading files



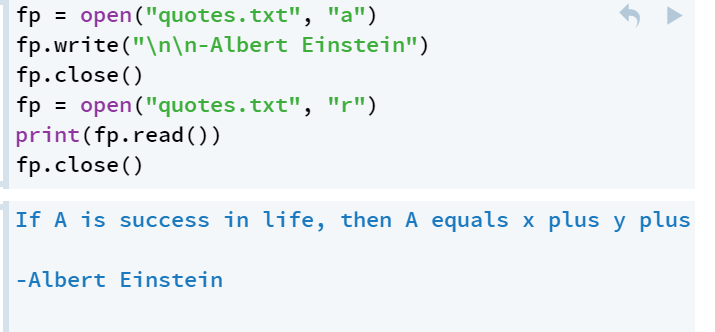
.read() 读文件中所有内容

Read Files a line at a time



.readlines()一行一行的读

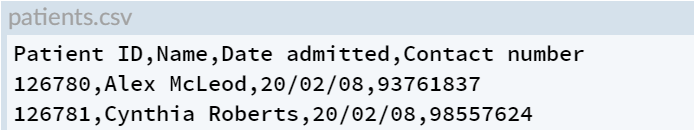
Appending to files



.write(“A string”)

更改原文件

The CSV format

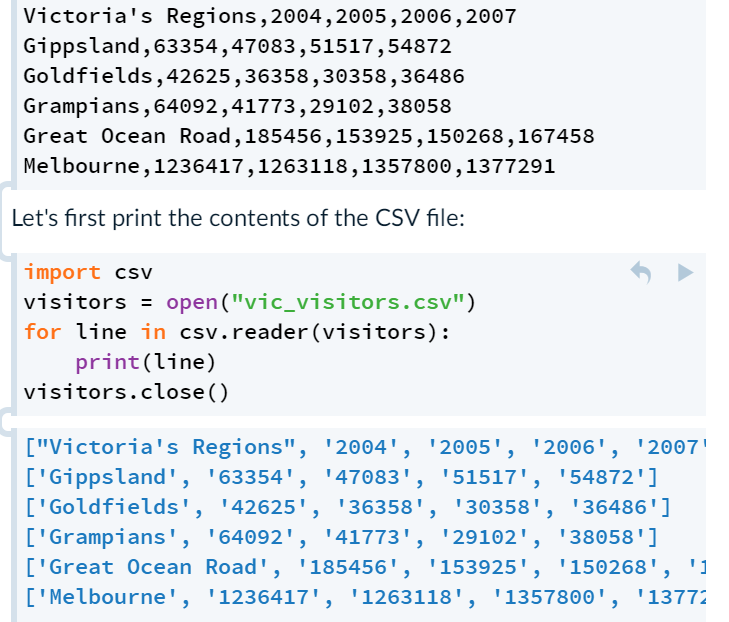


每一行叫 a row / a record

每一列中的值 field values

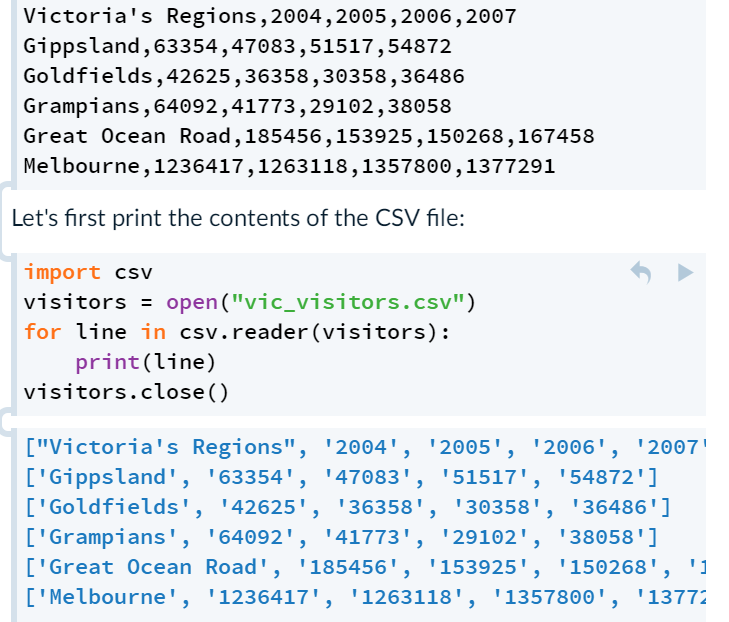
第一行 header row

The CSV library



.reader() iterate through a CSV file

自动将一行转化为string 的list



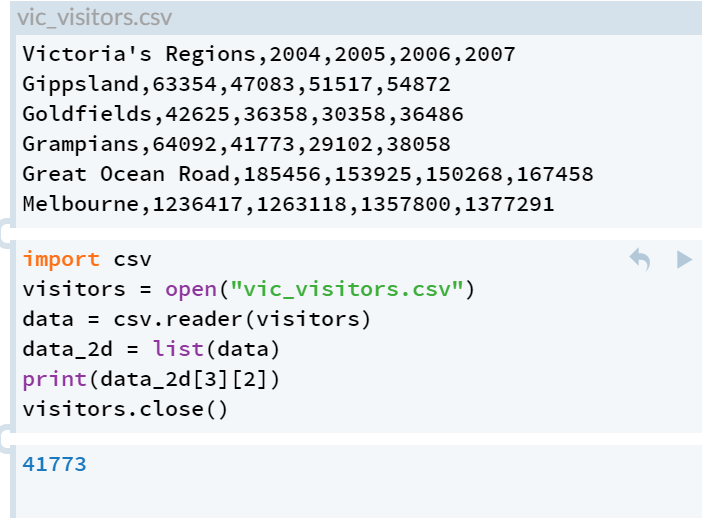
Column Headings



,next()

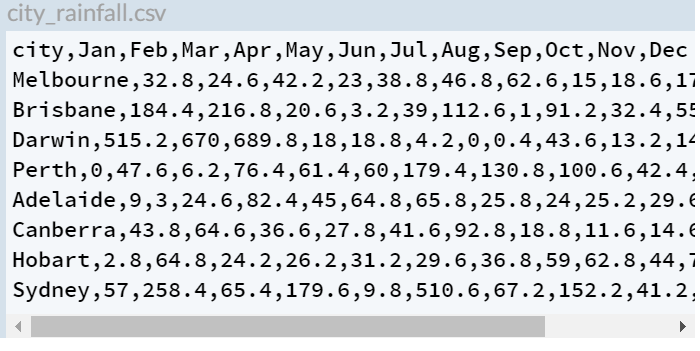
h causes the iterable to return its first element, and iterate onto the next row of data.

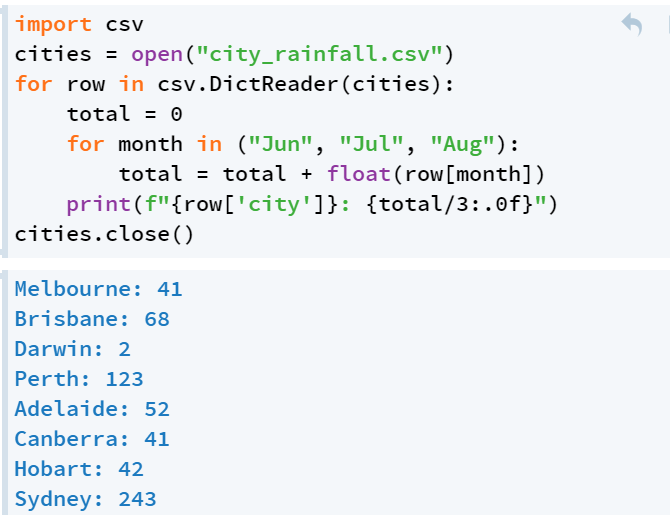
CSV file as 2D Data



用list() 将他转化为二维列表 以便调取具体的某一行某一列的数

Naming the Elements of a row



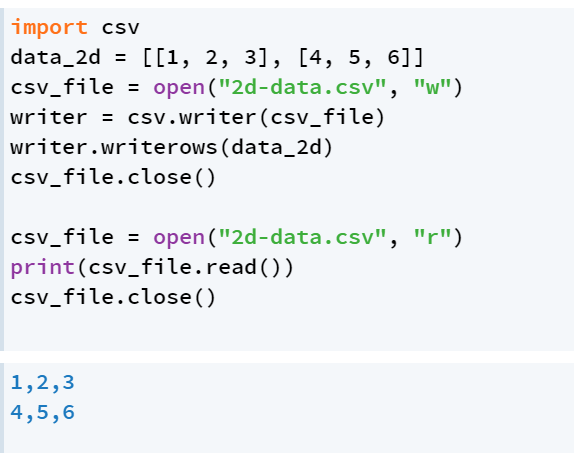


.DictReader() 每一行都被转化为一个dictionary

如{City: Melbourne; Jan: 32; Feb: 8…..}

可以直接通过column label 调用

Writing a CSV File



csv.writer() 打开一个文件 将他转变为csv writer 文件

.writerows()将列表转化为字符串 （如“1, 2, 3”）

Week 15

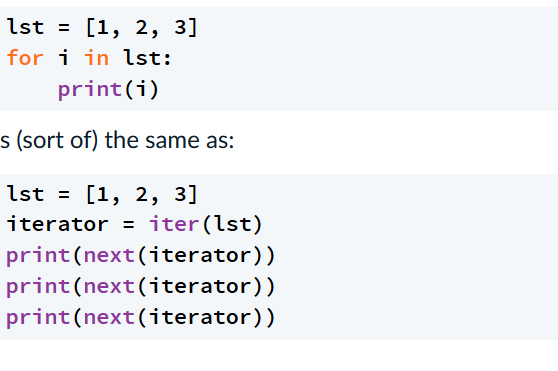
Iterators

Iterator object

keeps track of where it is up to in the object, implements a method that returns the next element

iter()

make an iterator object out of all the sequences



next()

iterate once over the iterator object

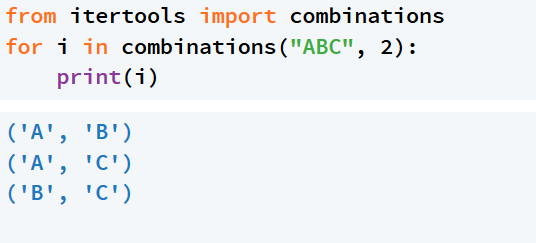
permutations(iterable, r)

iterable 中r 个元素所组成的排列Anr



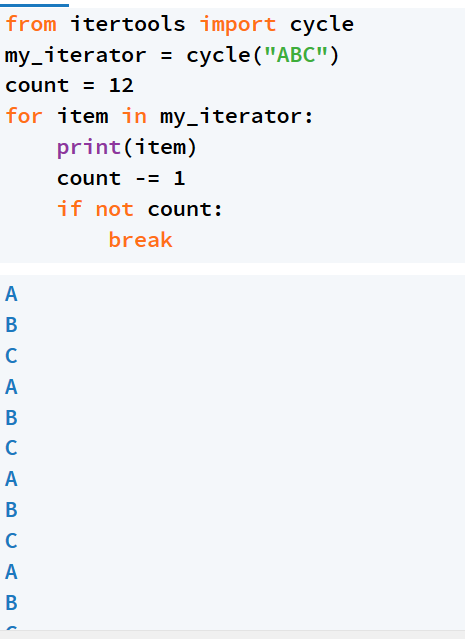
combinations(iterable, r)

.从iterable 中抽取r个数



Cycle()

循环重复iterable





Counter = itertools.count(100, step = 10)

Print(next(count))

Worksheet 6

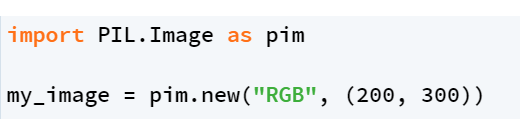
导入



We will use Pillow to manipulate digital images and practice writing more complex programs in Python

**pim.open(filename)** opens an image file called **filename** (a string)\

**pim.new(mode, size, bg\_colour)** creates a new blank image with colour-mode **mode** with dimensions **size**.



**image.size**: a tuple representation of the size  (200, 300)

**my\_image.width** : width 200

**my\_image.height** height 300

Color mode

* **"1"**  for black and white images
* **"L"**  for greyscale images
* **"RGB"**  for RGB (Red, Green, Blue) colour images

**my\_image.mode** ： evaluate to the string representing the mode of this image.

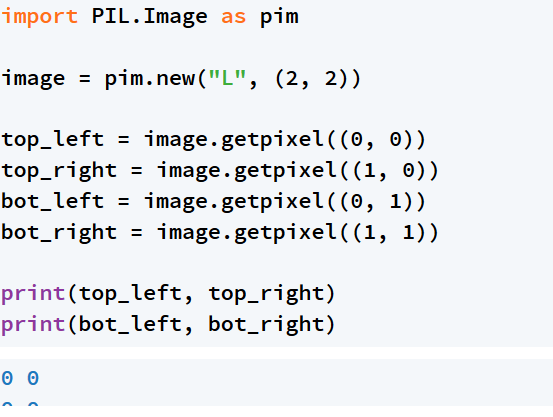
Accessing pixels

We can get the colour value of any pixel in an Image object's grid by using a special method called .getpixel(x\_coordinate, y\_coordinate)

Black/ White: black 0, white 1

Grey: 0-255

RGB 3-tuple of values(0-255)



Create image 时 grey 默认为黑色

Change image

Image.putpixel(2\_tuple\_address, color\_value)



Saving image

.save()



Manipulating image



调整颜色亮度

1 打开文件

2 创建一个一样size的图片

3 x, y循环

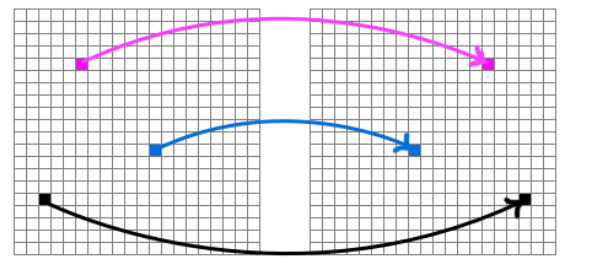
4 提取出原来的pixel

5 修改pixel

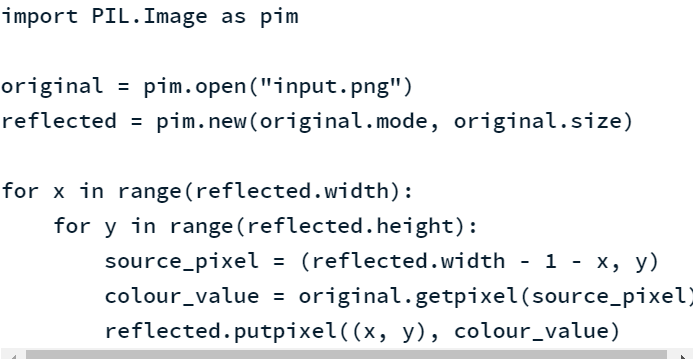
6 更改新建的pixel

Average colors

将原图做轴对称变化



New\_x = width – 1 - original\_x



1 复制一个一样大小的图片

2 找到与（x, y）对应的 source pixel

3 找到相应的color

4 更改(x, y)上的color

Week 7

Exception handling

x = “Not a number”

while type(x) != int:

try:

x = int(input(“Please enter a number: “))

except ValueError:

print(“Oops! Try again!”)

如果程序运行正常， 跳过

如果遇到此错误，执行以下内容

There are four steps to drawing a bar chart with **matplotlib**:

1. Generating some "axes" to plot relative to.
2. Plotting bars on the axes.
3. Labelling the axes.
4. Displaying the axes.

[https://docs.python.org/3/library/stdtypes.html#mapping-types-dict](https://docs.python.org/3/library/stdtypes.html" \l "mapping-types-dict)

<https://docs.python.org/3.7/library/stdtypes.html#sequence-types-list-tuple-range>