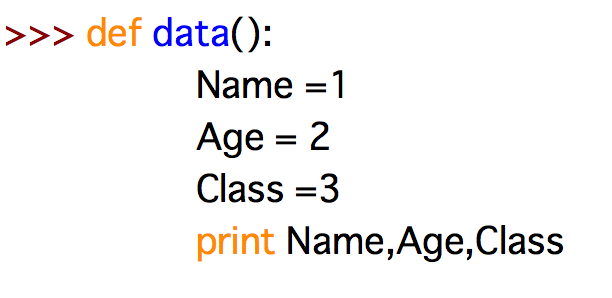
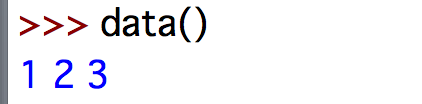
第七课，函数

1.函数：

什么是函数，是一个取了名字的一堆代码的集合。比如我们想完成一个人的资料。name=1，age =2，class=3，把这三句代码放入一个data的函数中，那么当我们调用data的时候，这三条代码就都会出来了。举个例子：我们在shell中，输入这样一段代码。



然后我们输出直接调用data（）这个函数



那我们看到的是输出了123出来，正是我们在函数中写的代码。那么，我们可以从里看出。如果我们在一个程序中，要使用三次这段代码，那不用函数的情况我们要写三遍Name，Age，Class，如果用了函数，我们可以方便的只写一次，然后我们去调用三次就可以了。

2.怎么来声明一个函数？

从上边的延时来看，我们清楚地可以知道def是声明一个函数的关键字。没错。 在我们想要创建一个函数的时候， def是必须得，然后我们要加一个空格，然后书写函数的明细。接着，我们再写一个括号，然后是冒号，格式：def 函数名（）：

注意：这里我们声明了一个函数之后，我们需要写一些函数的内容，比如刚才的name age，class和print，这些都是函数的内容，我们可以称他们是函数体。那么完成我们需要的功能的部分就是函数体了。

3.今天的代码：

7\_1defhouse.py

在这个程序中，def house（）是一个函数，他完成了我们上节课房子的搭建功能。然后我们，只要在运行程序是调用这个函数就可以了。那么我有一个问题：如果我不调用函数，那么在我们运行程序文件的时候，这个函数会运行么？答案肯定是不运行，如果我们不调用函数，这个函数我们写出来是不会被运行的。这关系到了。代码运行的先后顺序，将会导致程序的不同结果。

7\_2Buildstreet.py

在这个程序中，我们学习一个新的函数。randint（）这是random库里的一个函数，它的作用是随机一个数。稍后我们介绍一下这个函数的用法。 现在我们先分析一下这个程序文件，首先，我们可以看出一个wool（）函数，里边定义了一个变量c，他得值是一个随机数。

然后，我们可以看到， mc.setBlocks(x+1,y+1,z+1,x+5,y-1,z+5,block.WOOL.id,c)

这是要做一个4\*2\*4的羊毛毯，那么这段代码中的c在id后边。他决定了羊毛毯的款式，也就是颜色。那么刚刚我们已经说到了。c是随机数，也就是说，我们将随机一个颜色。接下来，我们看到，再循环中，我们调用了，wool并且，给x重新复制，使得x每循环一次增加5，使10次循环出来的羊毛毯紧挨着排列。

4.随机数：

刚刚我们说到了。random这个函数库。它里边的代码都是关于随机数的。那么我们目前能使用到的也就是random.randint()，这个函数，他可以随即一个范围内的数出来，这个范围我们就可以直接在括号里指出来，比如在我们刚刚的代码中。0，15这就是说我们要给C赋值一个0-15的一个随机数。

第八课，与方块交互

1.什么是交互？

其实在我们的印象中，交互，就是交流，互通，通信，

在我们程序中，交互的意思是指，对象与对象之间的交流信息，互相通信的意思，比如，对象A想和对象B做一个比较。对象A想要获取对象B的某个信息，或者对象A想把某个信息传递给对象B。这都是交互的体现。

2. getBlock（）

我们之前在创建一个房子的时候，已经学习过setBlock（）方法了。我们是用他，可以在某个坐标点上创建一个属于我们自己的方块。 那么，这个getBlock（）函数，可以检测到某个坐标点上的方块是很么属性的。比如玩家脚下的坐标。pos.x,pos.y-1,pos.z

3. 8\_1safeFeet.py

我们本次代码的任务是检测我们脚下是否安全，那么我们已经学习了函数的应用， 所以我们要创建一个函数， safeFeet（）

我们用这个函数来检测脚下的方块。是否安全。 b = mc.getBlock(pos.x,pos.y-1,pos.z)，我们用b对象获取到了玩家脚下的方块。如果这个方块是空气或者水，那么我们就处在不安全的地带。那么我们进行判断， 将我们脚下的这个方块b来和block.AIR.id与block.WATER\_STATIONARY.id与block.WATER\_FLOWING.id来对比。如果这三个对比都不为真，那么我们将else 打印安全。如果有一个成立，说明我们处于危险地带，打印危险。然后， 我们在循环中调用一下safeFeet（）这个方法

第九课，隐形桥梁

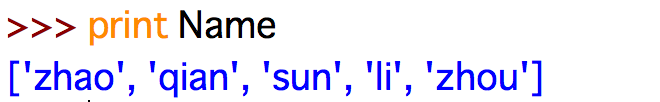
1.List：

什么是列表？打个比方。我们在学校做操的时候，每个学生都要按照队列来排队。那么，这就是一个列表，把每一个同学放在了一个表里，那么我们来看一下，怎么做一个列表。 首先我们需要一个变量。Name，然后把一个list赋值给这个Nmae。

在图片中我们可以看出Name是列表的名字，[ ]中的‘zhao’，‘qian’

是列表中的元素。元素和元素之间我们用逗号隔开

接下来我们打印一下这个列表：

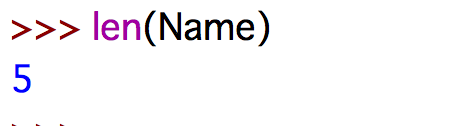


我们可以清晰的看到，将整个离别输出了出来。

2.常用的list操作：

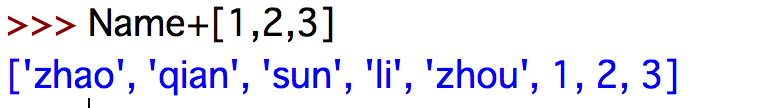
接下来我们来学习一下，列表的常用操作：

①



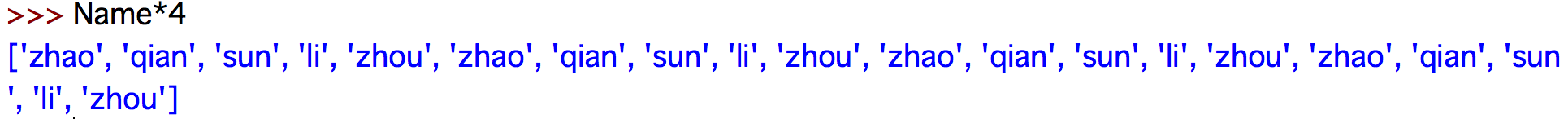
返回列表的长度，也可以直接用len(['zhao','qian','sun'])来使用，返回值是3

②



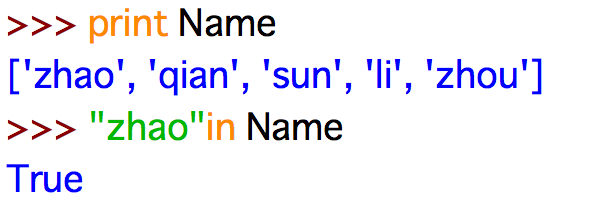
合并列表。也可以用[1,2,3]+['zhao','qian','sun']合并结果是[1,2,3，'zhao','qian','sun']

③



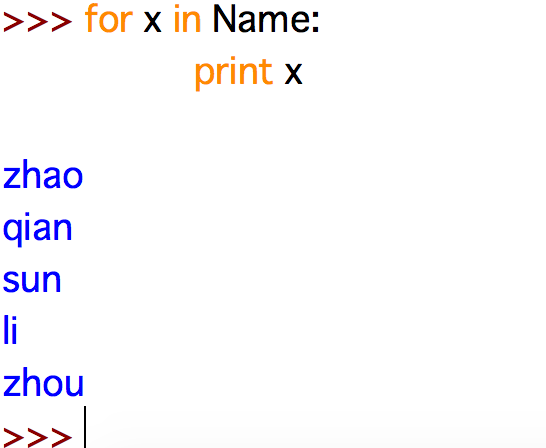
将一个列表重复4次合并

④

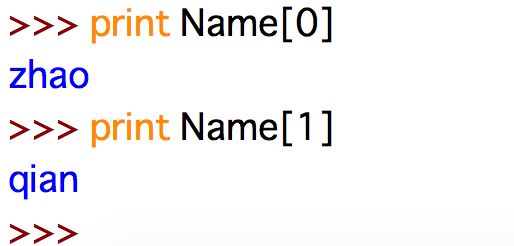


判断元素是否在列表中，

⑤



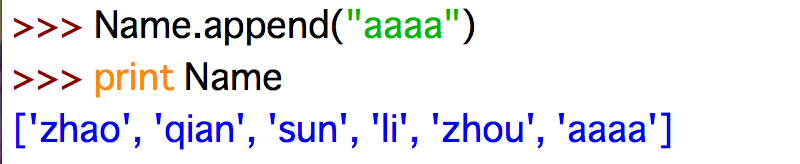
迭代列表元素

⑥

通过索引print列表单个值

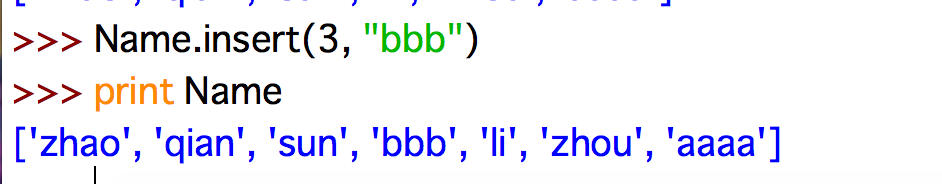
3.常用的一下列表的方法：

①



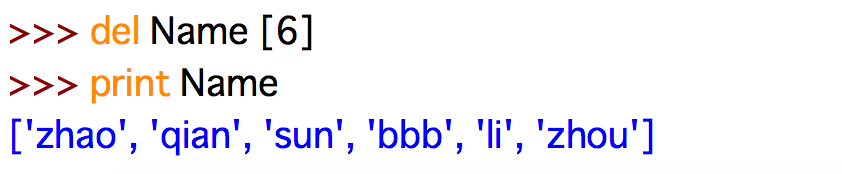
在列表末尾添加一个元素

②



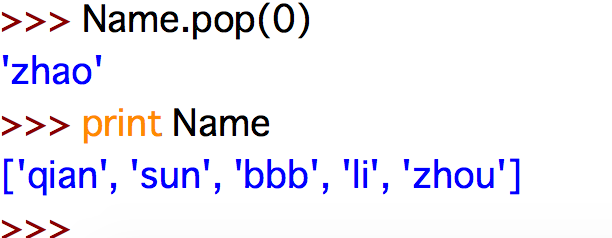
将元素插入到列表，3是索引位置（第四个位置）“bbb”是元素

③

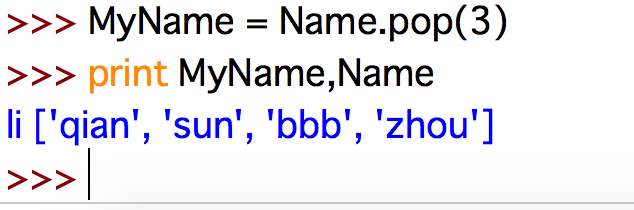


通过del和列表索引删除索引值所在的元素

④



和del很像，也是通过索引来删除列表元素，但pop（）会返回被删除的元素，举个例子：



我将Name中的li这个元素删除并赋值给了MyName。

接下来我们看一下我们今天的代码

4. 9\_1YCBridge.py：

今天的程序是要再玩家脚下做一个魔法桥梁，当我们脚下腾空的时候，要做一个玻璃方块在我们脚下。

首先我们声明了一个函数，buildBridge（）在函数里我们将要实现房子的搭建工作。让我们来看一下buildBridge（）函数内部的结构。

首先我们获取玩家坐标，检测玩家脚下的坐标是否为空气，或者水，如果是，那么我们就创造一个玻璃块，并且把玩家当前的坐标存入coordinate，形成一个列表。并且把这个列表存入到bridge中。elif这个关键字我们可以这样翻译：如果不是就判断。判断玩家脚下的方块是否为玻璃块。这是我们要判断一下bridge的长度。如果大于零，说明bridge中有之前存入的坐标，那么我们要把这个坐标删除，并且重新存入corrdinate中，然后我们在这里创造一个空气块，坐标为corrdinate中的012这三个元素。讲到这里，我们已经可以看出，我们的目的是，在我们移动的过程中， 我们要删除之前遗留的玻璃块，使得，只有我们脚下一个玻璃块。