

**Blockly教学案例**

**-第五版-**



2020-10-5

目录

[教学实验对象 1](#_Toc52775192)

[受众： 1](#_Toc52775193)

[教学目的： 1](#_Toc52775194)

[教学知识点： 1](#_Toc52775195)

[涉及知识点： 1](#_Toc52775196)

[教学内容： 2](#_Toc52775197)

[Plan1 2](#_Toc52775198)

[Plan2、Plan2-after1、Plan2-after2、Plan2-after3、Plan2-after4： 4](#_Toc52775199)

[Plan3 8](#_Toc52775200)

[Summary： 9](#_Toc52775201)

[教学安排 9](#_Toc52775202)

[循环： 9](#_Toc52775203)

[列表 9](#_Toc52775204)

[字典： 10](#_Toc52775205)

[函数 11](#_Toc52775206)

[总结与交流 11](#_Toc52775207)

[版本管理 11](#_Toc52775208)

## 教学实验对象

受众：没有python编写经验的学员

教学目的：通过Blockly + python 对应， 使得受众更好的了解和掌握Python

### 教学知识点：

循环初步，包括while、for-each

字典初步，包括字典的建立、插入/更新、弹出、获得所有键、字典遍历

可修改对象和不可修改对象

### 涉及知识点：

变量、循环、字典、列表、可修改对象参数与不可修改对象、函数、包管理

## 教学内容：

Plan1：感性理解算法流程，复习循环的用法

Plan2：了解字典

Plan2-after1了解字典的创建

Plan2-after2了解字典的取值

Plan2-after3了解字典的插入与修改

Plan2-after4了解字典的删除

Plan3：完成本次实验中字典部分的内容

summary：进行测试

### Plan1

**提供了一个生活场景**：总所周知，安排教学是一个需要细细考量的活。一个知识点的教导安排要关注到先修知识。显然我们需要安排的教学计划力求教导这个知识点前，这个知识点需要的先修知识点都教过啦。如果我们能够找到一个教学计划满足上面条件，我们称这个计划是优秀的。

**设计了一个问题**:现在给定一张知识点清单和知识点间的关系(详细见本题用例)，请编写程序找到一个优秀的清单。

**本题设计目标**：了解循环，掌握python的循环语法，包括while和for-each两种。

**本题提供的提示**：

我们通常会怎么做呢？

1.开开会，了解一下目的，搜集下资料

2.把要教导的知识点整理一下

3.对知识点进行划分，分成可以立即教导的和不能立即教导的

4.看看当前有没有马上可以教的

5.把能教的所有都安排上

6.对剩下的知识点进行划分，分成能可以立即教和不能立即教的

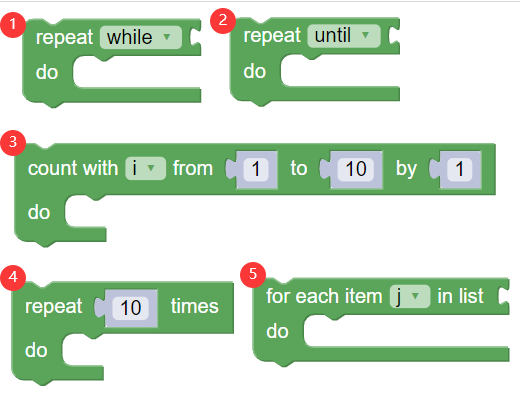
....不断重复这个操作直到没有可以立即教的

**具体知识点-循环的教学**：

在上面这个做法中，我们提到了"不断重复"和"能教导的所有"这个词，这两个词的出现往往意味着出现了循环操作。

循环语句允许我们执行一个语句或语句组多次。

基本上，所有的主流语言都支持循环，我们通过blockly来了解循环的共性：



blockly看上去为我们提供了很多的循环：

第一种为当条件满足时重复执行，第二种为反复执行直到条件不满足，判断依据都是条件

第三种，第四种，第五种都是提前计算重复次数。

事实上，我们完全可以使用前两种方式替换后三种,而第一种和第二种之间可以相互转化。

主流的语言都支持循环语法，但为了帮助我们更好的阅读和编写代码，有些语言提供了更多的语法支持，但正如上面的描述，其实一种就够了。

Python 提供了 while 循环和 for 循环

Python 编程中 while 语句用于循环执行程序，即在某条件下，循环执行某段程序，以处理需要重复处理的相同任务。

Python 里 while 的基本形式:

while 判断条件(condition)：

执行语句(statements)

Python for循环可以遍历任何序列的项目，如一个列表或者一个字符串。

Python 里 for 的基本形式:

for iterating\_var in sequence:

statements(s)

先看help目录下面的知识点.xmind

先走一遍流程

深复制，完全两个字典。

"收集要教导的知识点，查找相关处理工具"

9个知识点，2个工具

1）def plan\_divide\_dict(di) :

tmp = [i for i in di if di[i]["pre"] == 0]

res = dict([[i , di.pop(i)] for i in tmp])

#print(res)

return res

抽取出能交，等于0放入res di里面所有等于0的去掉。

Res di

2）def plan\_deal(it, di) :

for j in it["nxt"]:

di[j]["pre"] = di[j]["pre"] - 1

后继节点减去1

enable\_dict 随便定义

这里只有while循环

enable\_dict = plan\_divide\_dict(remain\_dict)

可以加断点，print

Enable\_dict remain\_dict

Enable\_dict 是否为空



为什么还要两个参数，因为这里是字典，字典有个key,value。遍历字典，只有key。所以还是要两个参数。



还有个变量 remain\_dict,没给的。我们封装了。

另外还有个很重要的，Dict，字典。

先讲Plan\_dict

Dict1

key是常量，不允许列表当个key。key不能重复



if({123:{"123":123},"123":{123:"123"}} != {})

嵌套字典，字典还可以嵌套列表

Dict2

字典两种获取方式，a[key],a.get(key)

a = {1:2}

c = a.get(2) //none

b = a[2] //出错

打开控制台 f12

a = {2:[1]}

b = a[2] //地址赋予

b.append(2)

print(a)

print(id(a[2]))

print(id(b))

比较

a = {2:[1]}

b = a[2]

b.append(2)

import copy

c = copy.deepcopy(a)

b = a

print(id(a))

print(id(b))

print(id(c))



插入也就是修改。只要key相同，就是修改。

Dic3 跳过去

Dic4删除



第一种删除

a = {1:2,4:"hello",7:[2,3,4]}

del a[1]

print(a)

第二种删除

a = {1:2,4:"hello",7:[2,3,4]}

b = a.pop(1)

print(a)

print(b)

第三种

del a

Plan2

Xmind 机器无法识别图

如何表示这个图

用什么数据结构来表示

通用的数据结构:例如邻接矩阵，邻接表等等。

用到什么信息。选择什么数据结构。

每个点(空出来时间），后面的点（具体），前面多少个点（不需要具体哪个点）

后面的点 next 是个列表 []

前面多少个点 数字

合起来都是当前知识点的信息（key,value)

所有知识点的信息就是很多(key,value)对

学了字典就开始拼接

Plan2

多了个工具

用户只是提供每个点的next点是什么

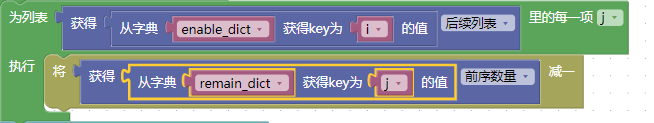
不能提供prev的个数，这个要程序自己算

前向统计的过程

可以处理其他链接图，不仅仅是这个测试案例

Dic3

少了减枝，我们把减枝拆分了出来



就是以前的减枝

不能用factory来写的逻辑块



cj\_dict\_block.js + cj\_dict\_stub.js

const minusImage =

开始。

### Plan2、Plan2-after1、Plan2-after2、Plan2-after3、Plan2-after4：

**本题的题目背景和Plan1是一样的。**

**本题设计目标**：了解字典dict、列表list。

**本题提供的提示**：

1. 知识点一： 作为[知识点二、知识点三]的先修知识，自身先修知识共计0个

2. 知识点二： 作为[知识点六]的先修知识，自身先修知识共计1个

3. 知识点三： 作为[知识点五、知识点八]的先修知识，自身先修知识共计1个

...

**具体知识点-字典的教学**：

了解字典

这是不是很像我们用的字典：条目和其信息？

并不是所有语言都直接支持字典，但是大多数语言直接或通过库的形式支持了字典，例如：

python有dict、C++的STL里有类似字典的map，java里java.util下有类似字典的map。

在计算机编程中，字典是一种可变容器模型，且可存储任意类型对象。

**Blockly的字典块**

Blockly并没有直接提供字典块，为了方面，笔者自定义了一些字典块方便使用。

目前字典提供了创建，插入、弹出、获得所有的key。

字典还有很多操作，不过目前我们先了解这些。

**python的字典操作**

创建字典

字典由键和对应值成对组成。字典也被称作关联数组或哈希表

基本语法：

变量 = dict 或 变量 = {} 或 变量 = {key : value , key2 = value2, ...}

其中key必须为不可变对象

例子：

a = dict()

b = {}

c = {1 : 2}

d = {1 : "2"}

e = {1 : ["2"]}

f = {1 : {1 : 2}}

g = {"1" : 2}

h = {1 : 2, "1" : "2"}

i = dict([[1, 2], ["1", "2"]])

其中

1. a : 创建一个空字典
2. b : 创建一个空字典
3. c : 创建一个key为数字1，value为数字2的字典
4. d : 创建一个key为数字1，value为数字2的字典
5. e : 创建一个key为数字1，value为链表["2"]的字典
6. f : 创建一个key为数字1，value为字典{1 : 2} 的字典
7. g : 创建一个key为字符1，value为数字2的字典
8. h ：创建一个key1为数字1，value1为数字2，key2为字符1，value2为字符2的字典
9. i : 通过列表创建一个字典

获取字典的值

可以通过key获取字典的值

基本语法:

字典[键] 键必须存在，否者会抛出异常 或者 字典.get(键) 键不存在时返回None 或者 字典.get(键, 默认值) 键不存在时返回默认值

例子：

a = dict()

b = {}

c = {1 : 2}

d = {1 : "2"}

e = {1 : ["2"]}

f = {1 : {1 : 2}}

g = {"1" : 2}

h = {1 : 2, "1" : "2"}

i = dict([[1, 2], ["1", "2"]])

vc = c[1]

e = e[1]

vg = g["1"]

vd = d[2] //key为2并不存在，抛出异常并终止运行

vd = d.get(2) //key为2并不存在，vd为None

vd = d.get(2, "No Key") //key为2并不存在，vd为"No Key"

由于python有可变对象和不可变对象之分，当字典的键保存的值为可变对象时，获取的是其值的引用

例子：

vc = c[1]

vc = 1

print(vc)

ve = e[1]

ve.append(1)

print(ve)

观察其不同

不可变对象的复制操作：Alice 有一个文档，复印了一份给Bob

Bob对文档的修改不会影响到Alice 的文档

可变对象的复制操作：Alice 有一个在线文档，共享链接给Bob

Bob对文档的修改会Alice的文档，因为看上去是两个的文档其实是一个文档

在字典中插入或修改对象

基本语法:

字典[键]=新值 如果键不存在即为插入，如果键存在即为修改

例子：

a = {}

a[1] = 2

a["1"] = 2

var1 = {"1" : 2}

a["dict"] = var1

a["self"] = a

观察字典A

在字典中删除键值对

基本语法:

del 字典[键] 或者 字典.pop(键) 采用字典.pop(键)拥有返回值

例子：

a = {1:2, "1":"2"}

del a[1]

b = a.pop("1")

观察字典A和B

### Plan3

**本题的题目背景和Plan1是一样的。**

**本题设计目标**：用于检测学员是否掌握字典，并且尝试通过等价来引出函数，为下一次教学做铺垫。

**本题提供的提示**：

这个块的语义很简单，就是从当前可教导的xxx中，安排教导这个x知识点，并更新其余知识点

我们首先要获得这个xxx列表，也就是enable\_dict[i]["nxt"]

然后使用循环去遍历这个列表，并且把列表中每一项在remain\_dict的pre减一。

观察源代码，了解两种写法的异同，并讨论目前这种写法和原先代码写法的优劣。

### Summary：

总结和提问环节，主要是测试学员掌握情况

## 教学安排

鉴于学员多数为近似零基础，内容不会讲的很深

上午：

### 循环：

关于循环的感性认知与思想理解（约10分钟）

关于循环的Blockly自主尝试（约30分钟）

关于Python的循环语法讲解（约10分钟）

关于变量在循环语句中的应用（约10分钟）

关于Python的循环代码编写（约30分钟）

容差时间：【-45，？】

### **列表**

根据学员对于循环的掌握程度以灵活处理本内容，在最坏情况下可以仅完成列表感性认知约列表

关于列表的感性认知（5分钟）

关于列表的创建（5分钟）

变量：关于Python可修改对象与不可修改对象（20分钟）

列表元素的获取（5分钟）

列表元素的修改（5分钟）

容错【？，？】

下午：

### 字典：

关于字典的感性认知（5分钟）

关于字典的创建讲解（5分钟）

关于字典的创建自主尝试（20分钟）

关于字典的值获取讲解（5分钟）

关于Python可修改对象与不可修改对象的教学（15分钟）

关于字典值获取的自主尝试（15分钟）

关于字典插入与修改对象讲解（5分钟）

关于字典插入与修改对象的自主尝试（15分钟）

关于字典对象的删除讲解（5分钟）

关于字典对象的删除的自主尝试（15分钟）

code.deepcode的使用（1分钟）

容错【-20，？】

Plan3作为过渡部分，检测对于字典与列表的理解。学有余力的同学可以初步了解函数。

### **函数**

根据学员接收程度灵活处理本部分，在最坏情况下正部分略去

了解函数思想（？分钟）

了解接口（5分钟）

了解函数的定义（5分钟）

了解函数的可变参数和不可变参数（5分钟）

### 总结与交流

关于本次实验的想法交流

## 版本管理

|  |  |  |
| --- | --- | --- |
| 版本 | 内容 | 作者 |
| 第一版 |  |  |
| 第二版 | 完成了循环初步、列表初步、字典初步、输入与输出、函数初步 | 黄佳俊 |
| 第三版 | 完善了字典、删除了输入 | 黄佳俊 |
| 第四版 | 做了一些语言表述的修正和界面的改变 | 黄佳俊、罗云花 |