```
myTable={}
print("---------table数组方式-------")
myTable[1]="baidu"
myTable[2]="guge"
myTable[3]="taobao"

print(myTable[1])
print("----------table键值对方式----------")
myTable2={}

myTable2['baidu']="www.baidu.com"
myTable2['360']="www.360.com"
print(myTable2['baidu'])
```

迭代器方式遍历table

```
for key,value in ipairs(表名) do
print(key,value)
end
//数组方式
for key,value in ipairs(myTable) do
        print(key,value)
```

```
end
//键值对方式
for key,value in pairs(myTable2) do
        print(key,value)
end
```

长度

table.getn

<3>table 长度

table.getn(表名)

返回 table 表的长度。

这个方式适合“数组模式”，不能用于“键值对模式”。

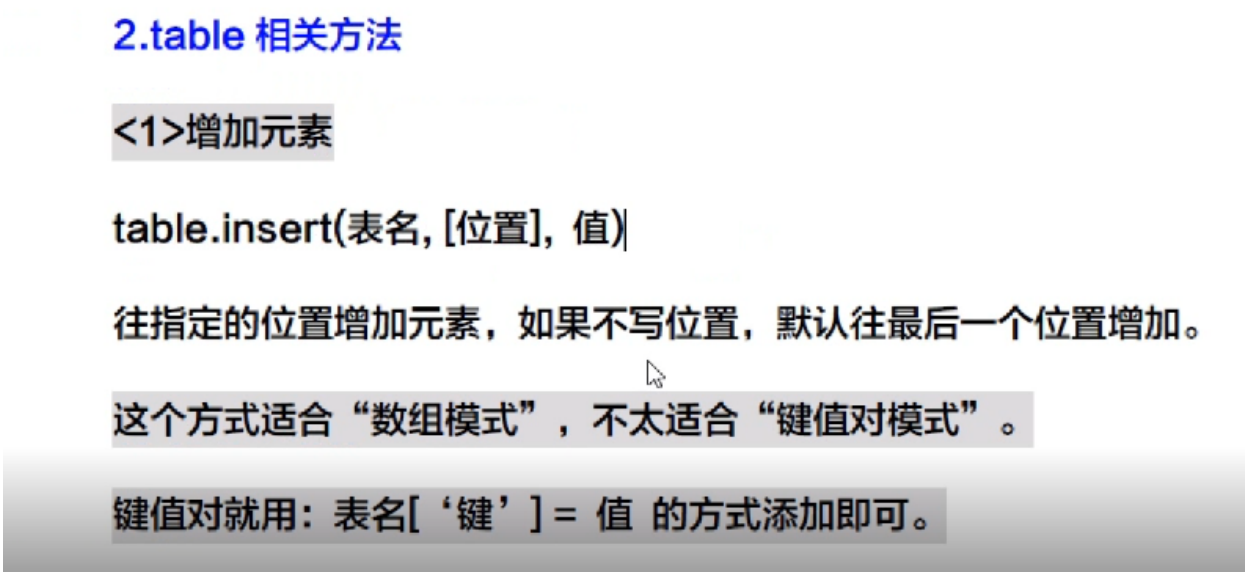
键值对就用：迭代器迭代，然后累加一个变量的方式获得长度。

```
index=0
for key,value in pairs(myTable2) do
        print(key,value)
        index=index+1
end
print(index)
```

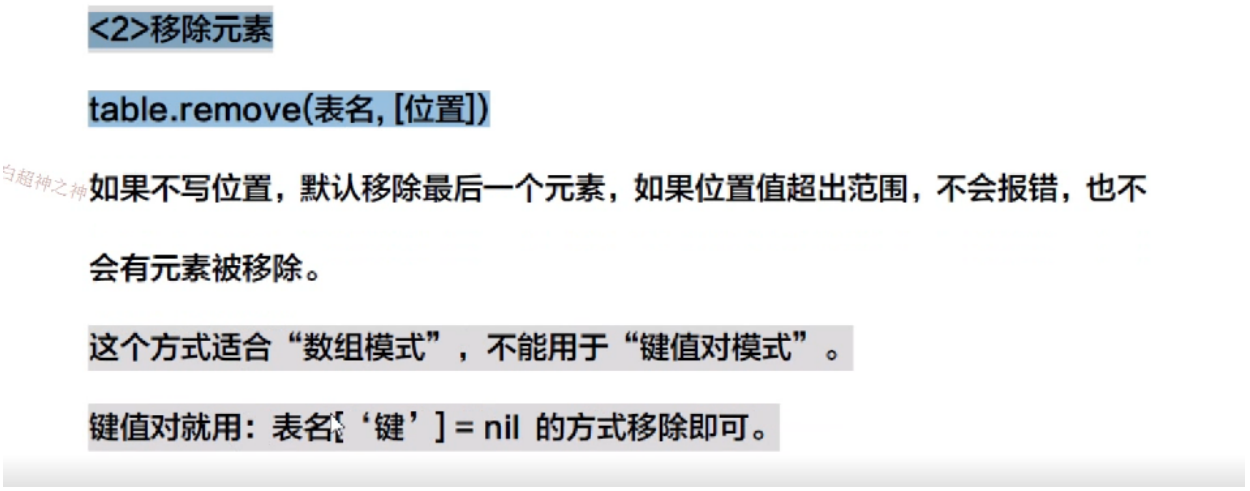
增加元素

table.insert

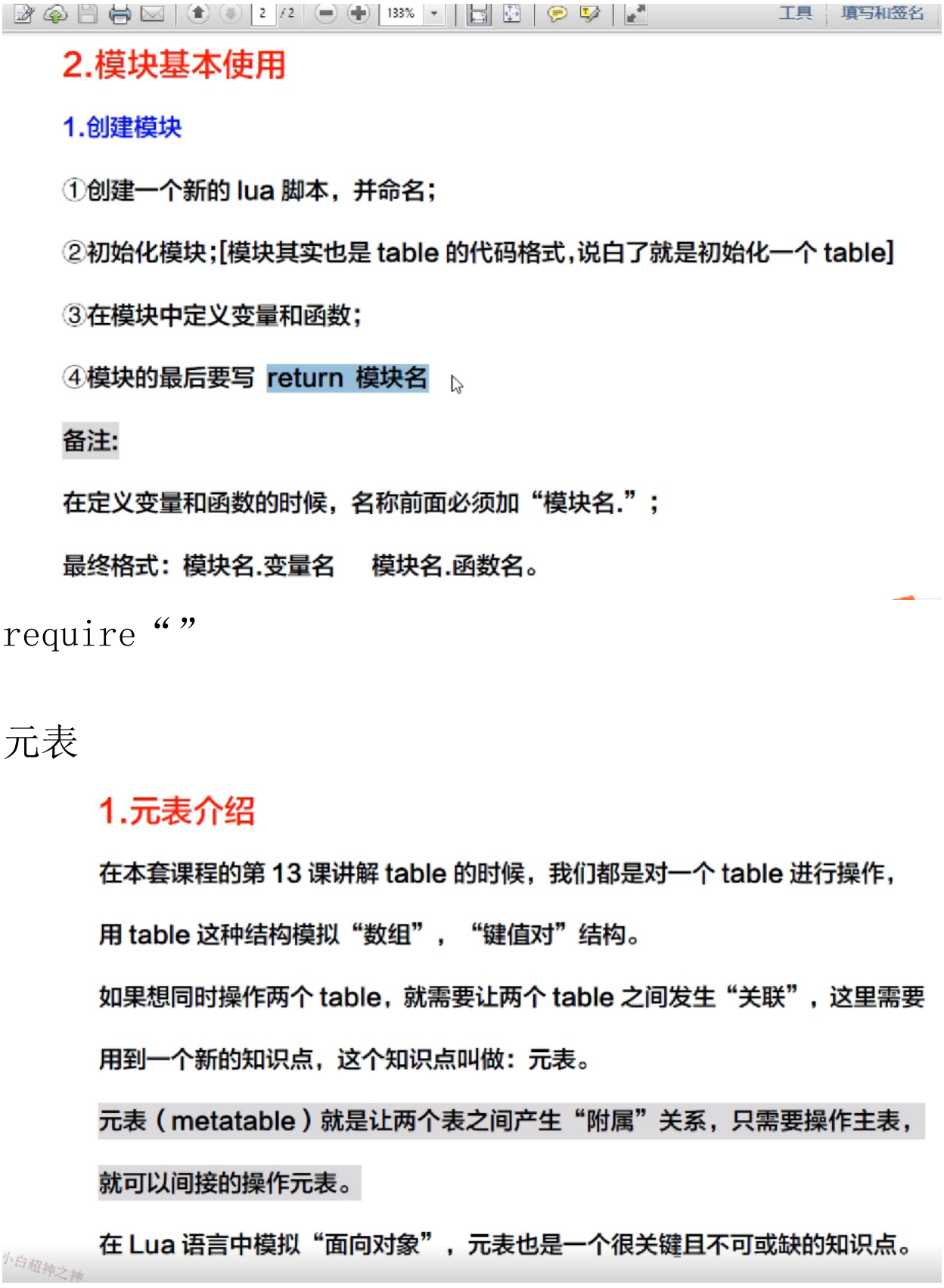
table.insert(myTable,1,""yahoo")



移除元素



模块



require“”

元表


```
tableA={}
tableB={}
--tableB是tableA的元表
setmetatable(tableA,tableB)
--判断tableA是否有元素
```

```
print(getmetatable(tableA))

tableC={name="Jack",age=10}
tableD={gender="男",adress="USA"}
setmetatable(tableC,tableD)
tableD.__index=tableD--设置元表的indx索引
print(tableC.name)
print(tableC.gender)
```