

# [笔记][LIKE-Python-2][03]

Python

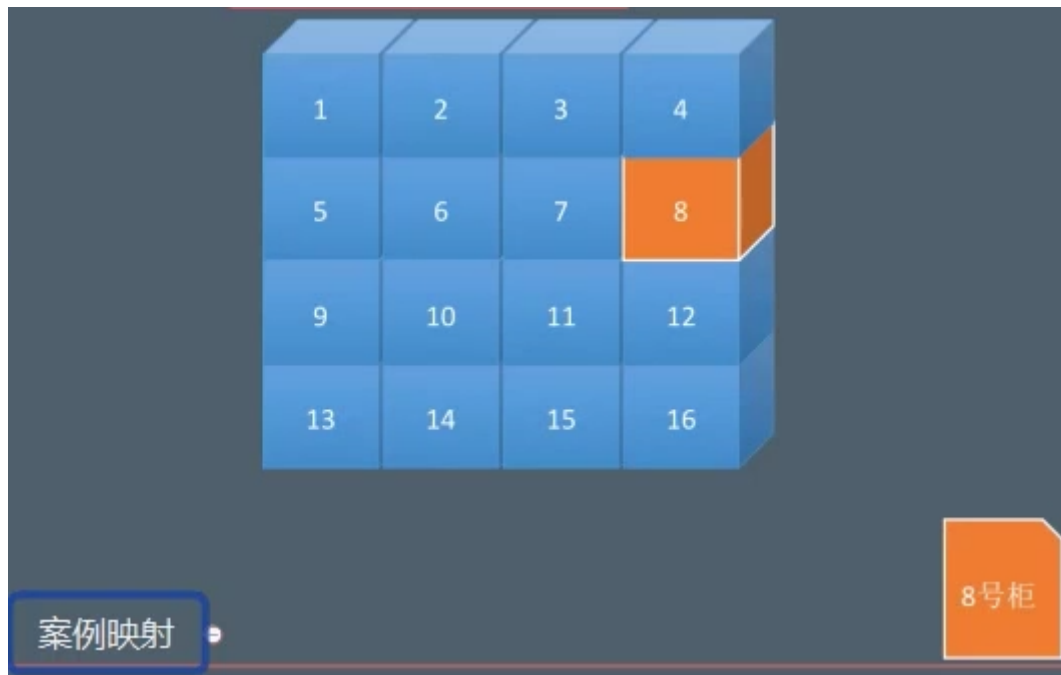
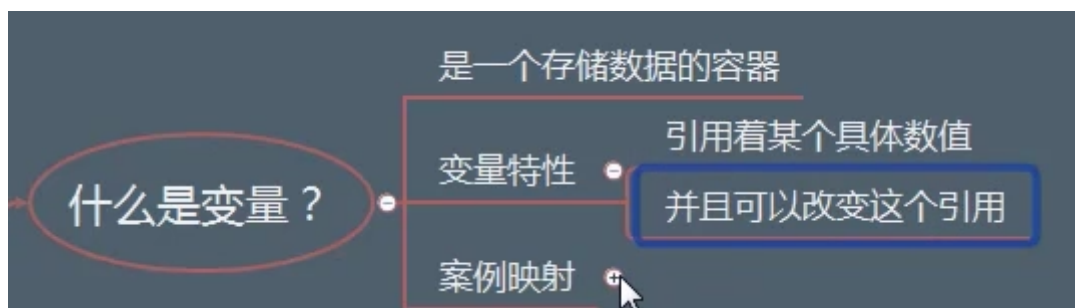
[笔记][LIKE-Python-2][03]

004. Python变量

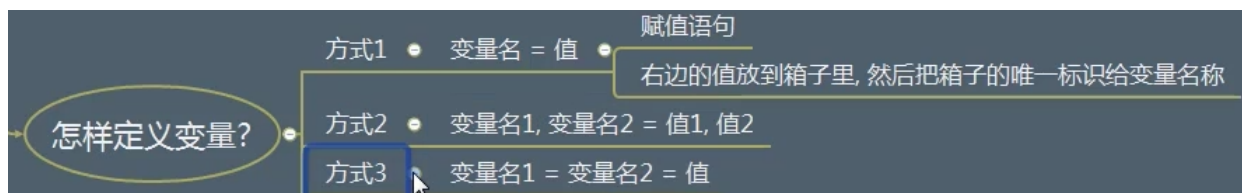
005. Python数据类型

006. Python类型转换

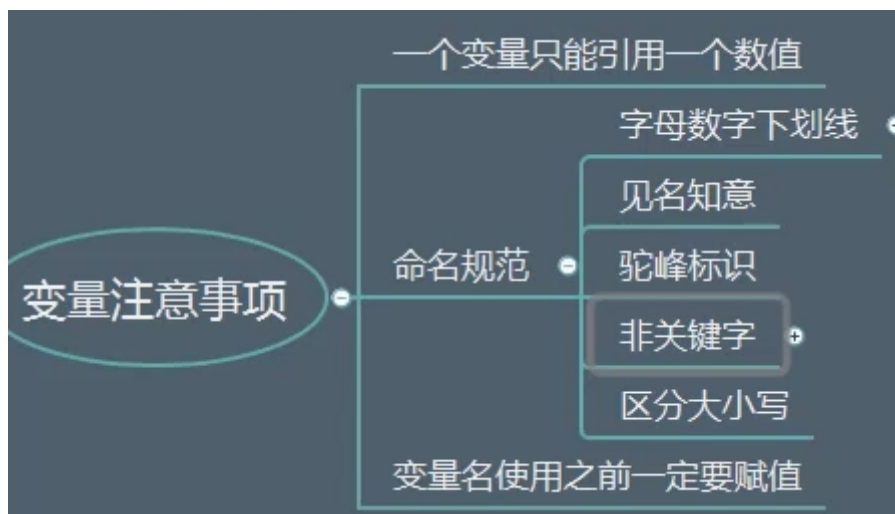
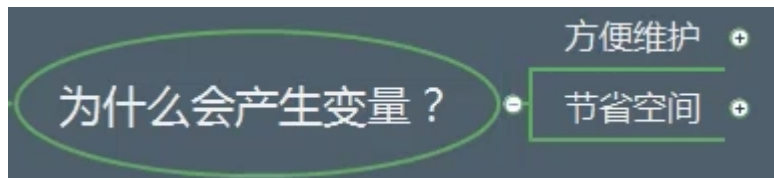
## 004. Python变量



变量实际上**不是**用来存储数据的容器，它是对数据的一个**引用**，我们可以改变这个引用。变量就相当于储物柜的小票。



PyCharm 小技巧: Ctrl - R 进行替换



【!】注意: 其实驼峰标识用来命名变量不太对, Python 官方推荐的是 snake\_case 这样的变量名。

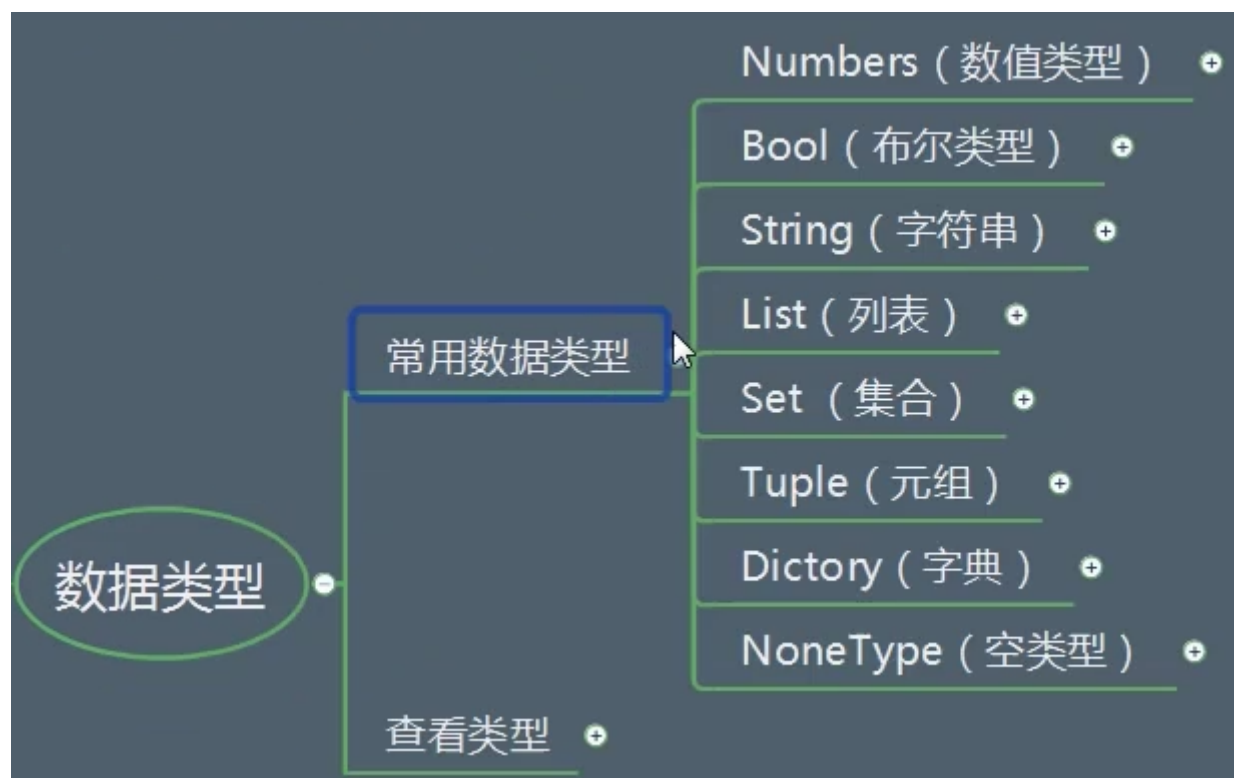
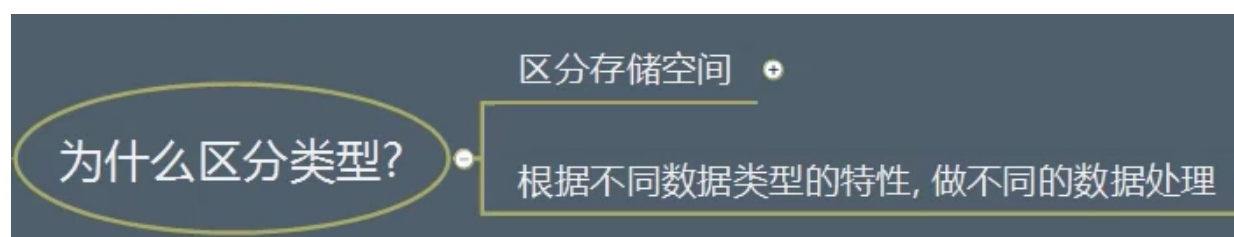
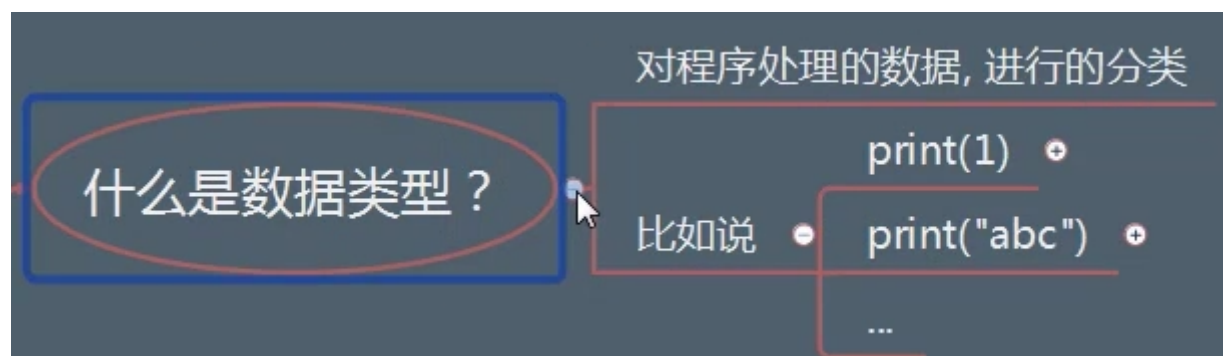
```
# 方式1
# 变量 = 值
a = 1
print(a)

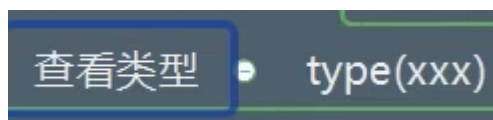
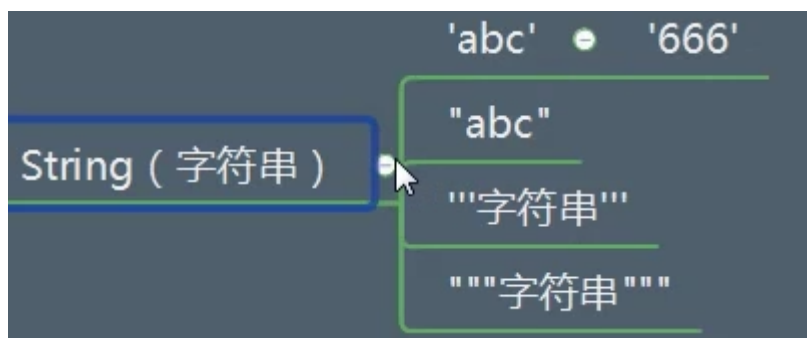
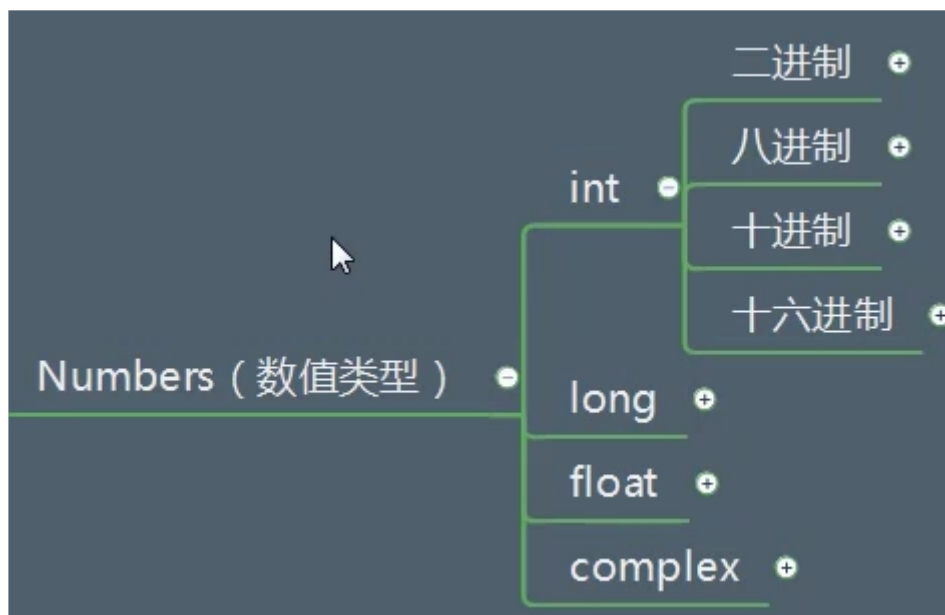
# 方式2
# 变量名1, 变量名2 = 值1, 值2
a, b = 1, 2
print(a, b)

# 方式3
# 变量名1 = 变量名2 = 值
a = b = 3
print(a, b)

# 查看关键字
import keyword
print(keyword.kwlist)
```

## 005. Python数据类型





```
print(6 + 6)
print('6' + '6')

# 查看数据类型
print(type(6)) # <class 'int'>
# 所有的东西都是对象

result = type('6')
print(result) # <class 'str'>

num = 10
print(type(num)) # <class 'int'>
# 不是判定变量名本身的数据类型
# 而是判断变量所引用的真实数据的类型
```

# 006. Python类型转换

数据类型转换

概念

场景?

方式

转换图

例如

函数	说明
<code>int(x, [,base])</code>	将X转换成为一个整数
<code>float(x)</code>	将X转换成为一个浮点数
<code>str(x)</code>	将X转换成为一个字符串
<code>repr(x)</code>	将X转换成为一个表达式字符串
<code>chr(x)</code>	将X转换成为一个字符
<code>unichr(x)</code>	将X转换成为一个Unicode字符
<code>ord(x)</code>	将X转换为对应的整数值
<code>hex(x)</code>	将X转换成为一个十六进制字符串
<code>oct(x)</code>	将X转换成为一个八进制字符串
<code>eval(str)</code>	计算字符串中的有效表达式, 并返回对象
<code>tuple(s)</code>	将序列S转换为一个元组
<code>list(s)</code>	将序列S转换为一个列表

面试补充

动态类型/静态类型

强类型/弱类型

结论

静态类型

动态类型

类型是编译的时候确定的,后期无法修改

类型是运行时进行判定的,可以动态修改

面试补充

强类型/弱类型

结论

强类型

弱类型

类型比较强势,不轻易随着环境的变化而变化

类型比较柔弱,不同的环境下,很容易被改变

Python是属于,强类型的,动态类型的语言

```
# Python 是个强类型的语言
# 不同数据类型不能运算
num = '6'
print(type(int(num)))
print(4 + int(num))
print(str(4) + num)

# 为什么要进行类型转换?
# 因为有时候无法控制数据的类型
score = input('请输入一个数字: ')
print(type(score))
print(int(score) + 6)

# 不能随意转换类型
num = '123'
result = int(num)
print(result)
# num = '123a' 这样就不能进行 int(num) 转换
```

完成于 201810271008