

# PYTHON

## 编程基础

# 字典的创建和初始化

# 创建字典



- 与集合类似，字典（dictionary）也是由若干无序的元素组成。
- 但与集合不同的是，字典是一种映射类型，字典中的每个元素都是键（key）:值（value）对的形式。
- 字典中每个元素键的取值必须唯一（即集合中不能包含键相同的元素）且必须是可哈希类型的数据，但对于每个元素值的取值则没有任何限制。

# 创建字典



- 字典的主要应用是做数据的快速检索。实际使用字典时，将要查询的数据作为键，将其他数据作为值。例如，在进行学生信息管理时，经常要根据学号进行学生信息的查询，此时就可以将学号作为键、而将其他信息作为值。
- 可以使用一对大括号 `{}` 或 `dict` 函数创建字典，如果要创建空字典可以使用 `{}` 或 `dict()`。

# 初始化字典元素



**fromkeys**方法的语法格式为：

```
d.fromkeys(seq[, value])
```

其中，d是一个已创建的字典对象；seq是一个包含了字典所有键名的序列；value是一个可选参数，其指定了各元素的初始值，默认情况下所有元素的值都被赋为None。

# 初始化字典元素



例：fromkeys使用示例。

```
d1={}.fromkeys(['sno','name','major'])
```

```
d2=dict().fromkeys(['sno','name','major'], 'Unknown')
```

```
print('d1的值为：', d1)
```

```
print('d2的值为：', d2)
```

d1的值为： {'sno': None, 'name': None, 'major': None}

d2的值为： {'sno': 'Unknown', 'name': 'Unknown', 'major': 'Unknown'}

# 初始化字典元素



提示：如果使用的字典对象中原来已经有其他元素，则调用`fromkeys`方法后原有的元素都会被清除。例如，对于下面的代码：

```
d1=dict(age=18)
print('d1的值为：', d1)
d2=d1.fromkeys(['sno','name','major'])
print('d2的值为：', d2)
```

## 初始化字典元素



程序执行完毕后，将在屏幕上输出：

d1的值为：`{'age': 18}`

d2的值为：`{'sno': None, 'name': None, 'major': None}`

可以看到，虽然在调用`fromkeys`方法前d1对象中已经有了一个键为`age`的元素，但该元素在d2中并不存在。