dict

--初始化

```
>>> mydict = dict(name1 = '...')
>>> mydict
{'name1': '...'}

>>> s = [('Tom', 5), ('Jone', 2), ('Susan', 4), ('Tom', 4), ('Tom', 1)]
>>> dict(s)
{'Jone': 2, 'Susan': 4, 'Tom': 1}
>>> mydict['fef']=1
>>> mydict
{'name1': '...', 'fef': 1}
```

--字典访问

```
mydict.keys()
mydict.values()
mydict.items()
```

返回可迭代对象

--in

```
>>>key in mydict
```

--mydict.clear()

清空字典

--mydict.get(key[,string])

尝试获取mydict中key的键值，失败返回string

--mydict.setdefault(key, default=None)

如果键不已经存在于字典中，将会添加键并将值设为默认值

```
>>>print ("Value : %s" %  dict.setdefault('Age', None))
```

--mydict.pop(key)

```
>>> mydict.pop()
Traceback (most recent call last):
  File "<pyshell#58>", line 1, in <module>
    mydict.pop()
TypeError: pop expected at least 1 arguments, got 0
>>> mydict.pop('fef')
1
```

--mydict.popitem()

随机弹出键对值

```
>>> mydict.popitem()
('name1', '...')
```

--mydict.update(anotherdict)

类似于list中的extend

--dict.fromkeys(keys[,values])

FE:

```
>>> fromkeys((1,2,3),('number','one'))
Traceback (most recent call last):
```

```
  File "<pyshell#53>", line 1, in <module>
    fromkeys((1,2,3),('number','one'))
NameError: name 'fromkeys' is not defined
>>> dict.fromkeys((1,2,3),('number','one'))
{1: ('number', 'one'), 2: ('number', 'one'), 3: ('number', 'one')}
>>> dict.fromkeys((1,2,3),'number')
{1: 'number', 2: 'number', 3: 'number'}
>>> dict.fromkeys((1,2,3))
{1: None, 2: None, 3: None}
```