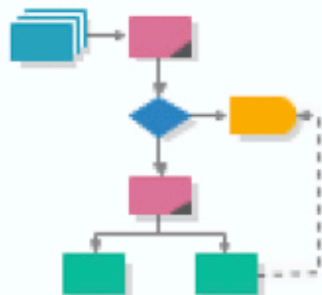


图解

Python

可能是迄今为止

最易懂的Python视频教程



```
"""类对象的特殊方法之__len__()"""
```

```
.....
```

内置函数`len()`用于返回对象的长度。

```
.....
```

```
print(len([1, 2, 3, 4, 5])) # 5
print(len('abcde'))         # 5
print(len({'a': 1, 'b': 2, 'c': 3})) # 3
```

```
.....
```

在上面的例子中，内置函数`len()`的实参都是内置类对象的实例对象，例如：

`[1, 2, 3, 4, 5]`是内置类对象`list`的一个实例对象；

`{'a': 1, 'b': 2, 'c': 3}`是内置类对象`dict`的一个实例对象。

内置函数`len()`的实参在默认情况下不能是自定义类对象的实例对象。

```
.....
```

```
# class MyClass(object):
#     pass
```

```
# print(len(MyClass())) # TypeError: object of type 'MyClass' has no len()
```

.....

如果想让内置函数`len()`的实参可以是自定义类对象的实例对象，必须在自定义类对象中实现特殊方法`__len__()`。这样，调用内置函数`len()`时，在其内部会自动调用实参所对应类对象的特殊方法`__len__()`。之所以难内置函数`len()`的实参可以是上述内置类对象的实例对象，是因为上述的内置类对象中都实现了特殊方法`__len__()`。

.....

```
class MyClass(object):  
    def __len__(self):  
        return 18
```

```
print(len(MyClass()))    # 18
```