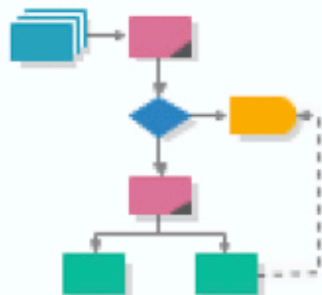


# 图解

# Python

可能是迄今为止

最易懂的Python视频教程



---

"""获取对象的信息之反射"""

"""

所谓"反射", 指的是以字符串的形式来操作 (包括: 增删改查) 对象的属性和方法。

用于"反射"的内置函数有以下四个 (参数name都是一个字符串):

1. `hasattr(object, name)`  
用于判断指定的对象object是否有参数name指定的属性或方法。
2. `getattr(object, name[, default])`  
用于获取指定的对象object中名为name的属性或方法。  
如果不指定参数default, 那么当object中不存在名为name的属性或方法时, 抛出AttributeError。  
如果指定了参数default, 那么当object中不存在名为name的属性或方法时, 就会返回default。  
`getattr(object, name)`等价于: `object.name`。
3. `setattr(object, name, value)`  
用于在指定的对象object中添加或修改名为参数name的属性或方法, 添加或修改后的值为value。  
`setattr(object, name, value)`等价于: `object.name = value`。
4. `delattr(object, name)`  
用于删除指定的对象object中名为参数name的属性或方法。  
`delattr(object, name)`等价于: `del object.name`。

注意: 只有在不知道对象信息的情况下, 才会去获取对象的信息。因此, 如果可以直接写: `object.name`, 就不要写为`getattr(object, 'name')`。

"""

---

```
class MyClass(object):
    def __init__(self):
        self.x = 1

    def do_sth(self):
        print("do_sth被调用")

mc = MyClass()

print(hasattr(mc, 'x'))          # True
print(hasattr(mc, 'do_sth'))    # True
print(hasattr(mc, 'y'))         # False

print(getattr(mc, 'x'))         # 1

f = getattr(mc, 'do_sth')
f() # do_sth被调用

# print(getattr(mc, 'y')) # AttributeError
print(getattr(mc, 'y', 2)) # 2

setattr(mc, 'z', 3)
print(getattr(mc, 'z')) # 3

setattr(mc, 'z', 4)
print(getattr(mc, 'z')) # 4

delattr(mc, 'z')
print(hasattr(mc, 'z')) # False
```