eval(str) 执行字符串的表达式

exec

compile

dir()

**一、函数反射**

\_\_name\_\_ 返回函数名

\_\_module\_\_ 返回函数所在模块

callable(obj) 判断是否可调用

函数、方法、lambda、类、实现\_\_call\_\_方法的实例返回True

**二、对象反射**

\_\_class\_\_ 返回对象的类

\_\_dict\_\_ 返回所有属性

\_\_dir\_\_() 列出所有方法和变量

hasattr(obj, attr) 判断属性是否存在

getattr(obj, attr) 访问属性 等同obj.attr

setattr(obj, attr) 设置属性 等同obj.attr =

delattr(obj, attr) 删除属性 等同del obj.attr

isinstance(obj, class) 判断是指定类实例

属性拦截

\_\_getattribute\_\_(self, name) 属性被访问时调用 (点式调用、getattr、hasattr)

缺省实现返回\_\_dict\_\_对应值，不存在抛AttributeError

\_\_getattr\_\_(self, name) \_\_getattribute\_\_方法抛出AttributeError时被调用

\_\_setattr\_\_(self, name, value) 设置属性时调用(点式调用、setattr)

\_\_delattr\_\_(self, name) 删除属性时调用(del obj.attr、delattr)

\_\_getitem\_\_(self, name) 属性[]访问时被调用

\_\_setitem\_\_(self, name, value) 属性[]设置时被调用

\_\_delitem\_\_(self, name) 属性[]删除是被调用

descriptor属性拦截：定义如下方法的类实例作为属性时，叫descriptor属性

\_\_get\_\_(self, instance, owner) 访问descriptor型的类属性时被调用

instance descriptor所在实例 (以实例调用类属性时，为None)

owner descriptor所在类

\_\_set\_\_(self, instance, value) 设置descriptor型的类属性时被调用

instance descriptor所在实例 (以实例调用类属性时，为None)

value descriptor所在类

\_\_delete\_\_(self, instance) 删除descriptor型的类属性时被调用

**三、类的反射**

\_\_name\_\_ 返回类的名称

\_\_bases\_\_ 返回类的所有父类

\_\_dict\_\_ 返回类的所有属性

\_\_module\_\_ 返回该类所在模块名

**四、模块反射**

\_\_name\_\_ 返回模块的名称，被执行脚本为'\_\_main\_\_'

\_\_dict\_\_ 返回模块所有变量

\_\_file\_\_ 返回模块所在路径

\_\_dir\_\_() 列出所有方法和变量

globals() 返回模块的所有全局变量

locals() 返回模块的局部变量(在方法中执行)