函数

- 函数定义
  - 把功能进行定义和封装，在需要的时候随时拿过来直接调用。
  - 语法：def 函数名(): 函数体
- 函数的调用
  - 语法：函数名(参数)
- 函数的返回值 return
  - return的作用
    - 结束函数
    - 返回一个结果
  - return的几种情况
    - 没有return ， 返回None
    - 只有return ， 返回None
    - return + 一个返回值
      在调用方接收一个返回值
    - return + 多个返回值
      用逗号隔开，在调用方接收的是元组类型的数据
- 命名空间
  - 内置命名空间
    - Python解释器内部运行时的变量函数
  - 全局命名空间
    - 当程序运行时将变量与值的对应关系存放在一个空间 我们在python文件中直接声明的变量，函数
  - 局部命名空间
    - 在函数内部声明的变量函数
  - 取值顺序（就近原则）
    - 局部命名空间>全局命名空间>内置命名空间
- 作用域
  - L :local,局部作用域，即函数定义的变量
    E :enclosing,嵌套的父级函数的局部作用域，即包含此函数的上级函数的局部作用域但不是全局
    G :global， 全局变量,就是模块级别定义的变量。
    B :built-in，系统固定模块里面的变量。比如int等。
    搜索变量的优先级顺序：LEGB
  - 全局作用域
    - 全局命名空间+内置命名空间 可以通过globals()函数查看全局作用域的名字
  - 局部作用于
    - 局部命名空间 可以通过locals()查看当前作用域的名字
- 迭代器和可迭代对象
  - 可迭代对象（iterable）
    - 用dir()函数查看对象，如果对象中含有__iter__()函数就是可迭代对象，str，list，tuple，dict，set，range都是可迭代对象
  - 迭代器（iterator）
    - 用dir()函数查看对象，如果对象中含有__next__()和__iter__()函数就是迭代器， 文件句柄是一个迭代器
    - 迭代器特点
      - 节省内存
      - 惰性机制
      - 不能反复，只能向下执行
- 生成器
  - 生成器：含有yield 的函数就是一个生成器
    - 生成器函数被执行获取到的是生成器而不是函数的执行
    - 最后一个yield之后，如果再进行__next__()函数会报错
    - 生成器是可迭代的，生成器可与进行for循环
    - send()可以让生成器向下执行一次，并给上一个yield赋值，第一个不能用send最后一个send不要传值
- 推导式
  - 列表推导式
    - 语法：[结果 for 变量 in 可迭代对象 if 条件]
  - 生成器推导式
    - 语法：(结果 for 变量 in 可迭代对象 if 条件)
  - 字典推导式
    - 语法：{结果 for 变量 in 可迭代对象 if 条件}
  - 集合推导式
    - 特点：去重
    - 语法：{结果 for 变量 in 可迭代对象 if 条件}
- 参数
  - 静态参数
    - 实参（调用的位置）在调用函数时，调用方传给函数的值
      - 位置参数
      - 关键字参数
      - 混合参数：先位置参数后关键字参数
    - 形参（声明的位置）在函数声明的位置，声明出来的变量
      - 位置参数
      - 默认值参数
        - 如果默认值参数是一个可变的数据类型，在调用他的时候改变他，其他位置 也跟着改变
      - 混合参数：先位置参数然后默认值参数
  - 动态参数
    - 形参
      - 形参位置 *args
        - 可以传入任何位置参数
          可以接收所有的位置参数组成元组（Tuple）类型的数据
      - 形参位置**kwargs
        - 可以接收所有的关键字参数， 组成字典
    - 实参
      - 实参位置的*和**表示的是打散
      - 形参位置的*和**表示聚合
- 闭包
  - 闭包就是内层函数对外层函数的局部变量的应用
    - 闭包的优点
      - 常驻内存，提高效率
      - 安全
    - 判断是不是闭包
      - 使用__closure__函数，如果不是闭包返回None
- 函数名的应用
  - 1）函数名可以想变量一样使用
  - 2）函数名可以作为函数的参数
  - 3）函数名可以作为return返回值
  - 4）函数名可以作为容器类变量的元素
- global和nonlocal
  - global()函数：把全局变量拿到局部使用，把局部变量提升为全局变量
  - global()函数：在函数中对父级或更高级的变量拿过来进行修改，但是不能修改全局变量