**Python3 Notes**

**-2019**

Edit by ZN

**Python\_Notes**

无数据类型；

**String** ：单双引号，转义\，取消转义，前加r，元素不可更改；

**List** [] ：列表，方括号，元素可更改；

**Tuple** () ：元组，圆括号，元素不可更改，一个元素时在其后添加逗号；

[x:y]：x：索引（前），y：截断（后），元组同；

string、list和tuple都属于sequence（序列）；

**Set**：可以使用 { } 或者 set() 函数创建集合，注意：创建一个空集合必须用 set() 而不是 { }，因为 { } 是用来创建一个空字典。基本功能是进行成员关系测试和删除重复元素；

**Dictionary**（字典）:列表是有序的对象集合，字典是无序的对象集合。字典是一种映射类型，用"{ }"标识，是一个无序的键(key) : 值(value)对集合。键(key)必须使用不可变类型。在同一个字典中，键(key)必须是唯一的。构造函数 dict() 可以直接从键值对序列中构建字典；

单行注释以 # 开头，多行注释用三个单引号 ''' 或者三个双引号 """ 将注释括起来；

**运算符**：逻辑运算符：and，or，not；成员运算符in，not in；身份运算符用于比较两个对象的存储单元：is，is not；\*\*：幂运算符；//：向下取整除运算符；

**交互式/脚本式**编程；

支持三种数值类型：**整型（int）、浮点型（float）、复数（complex）**；

变量在使用前必须先"定义"（即赋予变量一个值），否则会出现错误；

在交互模式中，最后被输出的表达式结果被赋值给变量 \_；

**del**：删除元素；**len**：长度；嵌套列表；

元组元素值不可修改，但是可以对元组进行连接组合；

**字典键的特性**：不允许同一个键出现两次。创建时如果同一个键被赋值两次，后一个值会被记住。键必须不可变，所以可以用数字，字符串或元组充当，而用列表就不行；

**条件控制**：每个条件后面要使用冒号:，表示接下来是满足条件后要执行的语句块。使用缩进来划分语句块，相同缩进数的语句在一起组成一个语句块。在Python中没有switch–case语句。

**循环语句**：for/while，没有do...while，注意冒号和缩进，（无限循环在服务器上客户端的实时请求非常有用），while...else，for循环遍历（for...else），range()函数数字数列，如果从for或while循环中终止（break），任何对应的循环else块将不执行，pass（空语句占位）；

**迭代器**：**iter()**，**next()，StopIteration 异常用于标识迭代的完成，防止出现无限循环的情况；**

**生成器**：yield是一个返回迭代器的函数，只能用于迭代操作；

**函数**：在python中，类型属于对象，变量是没有类型的；可更改和不可更改对象；**参数**：必需参数，关键字参数，默认参数，不定长参数；加了星号\*的参数会以元组(tuple)的形式导入，存放所有未命名的变量参数；加了两个星号\*\*的参数会以字典的形式导入；如果单独出现星号\*后的参数必须用关键字传入；匿名函数：lambda表达式；

（语法）lambda [arg1 [,arg2,.....argn]]:expression

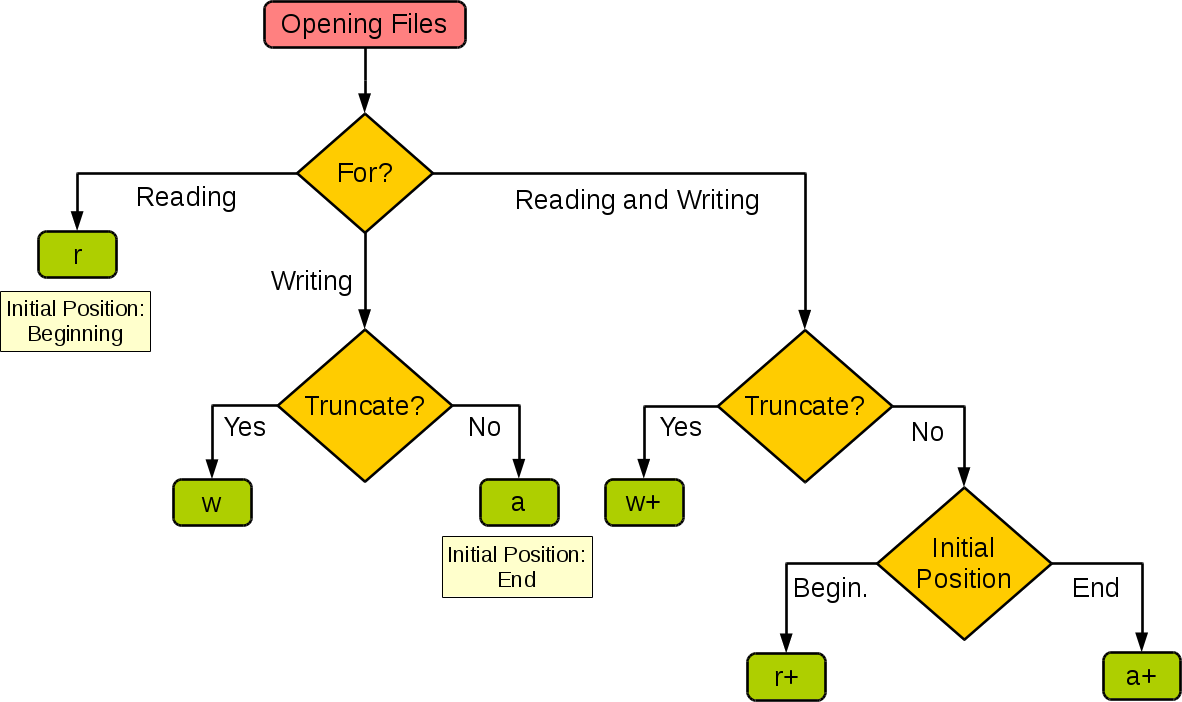
**变量作用域**：只有模块（module），类（class）以及函数（def、lambda）才会引入新的作用域；

**global和nonlocal关键字**：内部作用域修改外部作用域的变量；

**数据结构**：**列表list**，将列表当做堆栈（append，pop）/队列（append，popleft）使用；**列表推导式**；**嵌套列表**；**del**语句（可以按照索引删除元素，可以删除实体变量）；**元组和序列**；**集合**（无序不重复{a,b,c,...},set()）；**字典**（{键值对}，dict函数直接构建）；遍历技巧（字典遍历用items()同时解读键值对，序列遍历用enumerate()同时得到索引和值，zip()同时遍历更多序列，reverse()反向遍历，sorted()返回已排序的序列（不修改原值））；

**模块**：包含所有定义的函数和变量的文件；**import**：一个模块只会被导入一次；**from...import**，**~~from...import\*~~**，**\_\_name\_\_**属性（使该程序块仅在该模块自身运行时执行，说明：每个模块都有一个\_\_name\_\_属性，当其值是'\_\_main\_\_'时，表明该模块自身在运行，否则是被引入），**dir()**函数（dir(模块)可以找到模块内定义的所有名称，dir()会罗列当前定义的所有名称），**标准模块库**，**包**（是一种管理Python模块命名空间的形式，目录只有包含一个叫做 \_\_init\_\_.py 的文件才会被认作是一个包）；

**输入和输出**：**str()**：函数返回一个用户易读的表达形式，**repr()**：产生一个解释器易读的表达形式（可以转义字符串中的特殊字符）。str.format() 函数来格式化输出值，**rjust()**右对齐（**ljust()**，**center()**）,**zfill()**（左侧填充0），读取键盘输入（**input()**），读和写文件（**open(filename, mode)**，filename：包含了你要访问的文件名称的字符串值，mode：决定了打开文件的模式：只读，写入，追加等。如下图所示）；



文件对象的方法：**read()**，**readline()**，**readlines()**，**write()**（返回写入的字符数，写入非字符串需进行转换），**tell()**，**seek()**，**close()**，**with...as**关键字（处理异常），**pickle**模块（实现基本的数据序列和反序列化）；

**File（文件）方法**：**open()**，（**close()**，**flush()**等函数）；

**OS文件/目录方法**：丰富函数；

**错误和异常**：异常处理（**try...except...else...finally**）：可有多个**except**，一个**except**可以同时处理多个异常（元组形式），抛出异常（**raise**，参数指定抛出的异常），用户自定义异常（异常类继承自**Exception**类），定义清理行为（**finally**：无论如何都会执行的清理行为），预定义的清理行为（如**with...as**）；

**面向对象**：类，**\_\_init\_\_()**（构造方法）在类实例化时自动调用；**self**代表类的实例，而非类（类的方法与普通的函数区别—必须有一个额外的第一个参数名称，按照惯例它的名称是self）；**类的方法**：**def**关键字定义；**继承**：class DerivedClassName(BaseClassName1)基类必须与派生类定义在一个作用域内；**多继承**：class DerivedClassName(Base1, Base2, Base3)；**方法重写**：在子类重写父类方法，**super()**函数是用于调用父类(超类)的一个方法；**类属性与方法**：私有属性：属性名前两个下划线’\_\_’，类的方法必须包含参数self，类的私有方法也是两个下划线开头；**类的专有方法**：如下所示；**运算符重载**；

\_\_init\_\_ : 构造函数，在生成对象时调用

\_\_del\_\_ : 析构函数，释放对象时使用

\_\_repr\_\_ : 打印，转换

\_\_setitem\_\_ : 按照索引赋值

\_\_getitem\_\_ : 按照索引获取值

\_\_len\_\_ : 获得长度

\_\_cmp\_\_ : 比较运算

\_\_call\_\_ : 函数调用

\_\_add\_\_ : 加运算

\_\_sub\_\_ : 减运算

\_\_mul\_\_ : 乘运算

\_\_truediv\_\_ : 除运算

\_\_mod\_\_ : 求余运算

\_\_pow\_\_ : 乘方

**标准库概览**：系统操作接口（**os**模块（dir()，help()），shutil模块（copyfile()，move()）），文件通配符（glob模块），命令行参数（sys模块（argv）），错误输出重定向和程序终止（stderr，sys.exit()），字符串正则匹配（re模块），数学（math模块，random模块），访问互联网（简单的（urllib.request，smtplib）），日期和时间（datetime模块），数据压缩（zlib，gzip，bz2，zipfile，tarfile），性能度量（timeit()，:mod:profile和pstats模块提供大代码块的时间度量工具），测试模块（doctest模块，unittest模块）；

**Python3高级教程**

**正则表达式：**re.match函数

**多线程**：（\_thread，threading模块），产生新线程，语法：\_thread.start\_new\_thread ( function, args[, kwargs] )，args必须是tuple类型。

threading.currentThread()：返回当前的线程变量。

threading.enumerate()：返回一个包含正在运行的线程的list。

threading.activeCount()：返回正在运行的线程数量。

Thread类提供了以下方法：

run()：用以表示线程活动的方法。

start()：启动线程活动。

join([time])：等待至线程中止。这阻塞调用线程直至线程的join()方法被调用中止-正常退出或者抛出未处理的异常-或者是可选的超时发生。

isAlive()：返回线程是否活动的。

getName()：返回线程名。

setName()：设置线程名。

**线程同步**：使用Thread对象的Lock和Rlock可以实现简单的线程同步（对于每次只允许一个线程操作的数据，将其操作放到acquire和release方法之间）；**线程优先级队列**（Queue模块）；

**XML解析**：

**JSON**：json.dumps()：对数据进行编码，json.loads()：对数据进行解码。