**python基础**

**第一章**

**1 字面量**

字符串用“”包起来。

**2 注释**

单行注释：#，多行注释：“““这是一段注释”””。

**3 变量**

print语法：print ( 内容1，内容2，内容3，…，内容N )

**4 数据类型**

type()语句作用：得到数据的类型，并带有返回值，可用变量接收。

语法：type(被查看类型的数据)

不仅能查看<字面量>的类型，还能查看<变量>中存储的数据类型

**5 数据类型转换**

常见的转换语句：

1 int ( x )，将x转换为一个整数。

2 float ( x )，将x转换为一个浮点数。

3 str ( x )，将x转换为字符串。

同type语句一样，这三个语句，都带有返回值，可以直接输出，也可用变量存储。

注意：

任何数据都能转换成字符串类型。

只有数字型字符串才能转换为数字。

**6 标识符**

规范的命名。

**7 运算符**

补充：1 // 整除，2 \*\* 指数，复合赋值运算符：同c。

**8 字符串**

8.1字符串的三种定义方式

单引号，双引号，三引号。

作用：1 在字符串中包含双引号，使用单引号定义。2 在字符串中包含单引号，使用双引号定义。

**8.2 字符串拼接**

字符串字面量之间，通过+号完成拼接。

字符串字面量和字符串变量 或 字符串变量和字符串变量之间，通过+号完成拼接。

注意：无法和非字符串类型进行拼接。

**8.3 字符串格式化**

字符串拼接的缺点：1 变量过多时，拼接起来太麻烦。 2 字符串无法和数字或其它类型完成拼接。

字符串格式化本质也是字符串拼接。

c知识扩充，以及语法规则，见下：

name = "黑马程序员"

message = "学IT就来%s" % name

其中的%s，%表示：我要占位，s表示：将变量变成字符串放入占位的地方，所以，综合起来的意思就是：我先占个位置，等一会有个变量过来，我把它变成字符串放到占位的位置。

多个变量占位：变量要用括号括起来，用“，”连接，并按照占位的顺序填入。

name = "黑马程序员"

salary = 19801

message = "学IT就来%s,月薪高达%s." % (name,salary)

**8.4 格式化的精度控制**

数字精度控制：我们可以使用辅助符号“m.n”来控制数据的宽度和精度。

m，控制宽度，要求是数字，设置的宽度小于数字自身，不生效。

.n，控制小数点精度，要求是数字，会进行小数的四舍五入。

例如：

%5d：表示将整数的宽度控制在5位，如数字11，被设置5d，就会变成：[空格][空格][空格]11，用三个空格补足宽度。

%5.2f：表示将宽度控制为5，将小数点精度设置为2。

**8.5 字符串格式化 - 快速写法**

前面的格式化方法效率还不够高，更高效方法见下：

语法：f“内容 { 变量 } ”

注意：该方法不理会类型，不做精度控制，原样输出。

name="曹中泰"

year=2004

age=18

print(f"我的名字是{name},出生于{year},年龄为{age}.")

**8.6 对表达式进行格式化**

表达式：一条具有明确执行结果的代码语句。

使用：在无需使用变量进行数据存储的时候，可以直接格式化表达式，简化代码。

print("1\*2的结果是%d" % (1\*2))

**9 数据输入 ( input语句 )**

name = input("请告诉我你是谁 ")# input语句，可以在输入内容前，输出提示内容。

print(name)

input 语句返回值为str，可用类型转换，成为整型、浮点型。

name = int (input("请告诉我你是谁 "))# 将 str 转换为 int。

**第二章**

**1 布尔类型和比较运算符**

python中常用的有6种数据类型：数字，字符串，列表，元组，集合，字典。

定义变量存储布尔类型数据：变量名称 = 布尔类型字面量。

比较运算符(>，<，==)

**2 if语句的基本格式**

if判断语句：语法：

if 要判断的条件：

条件成立时，要做的事。

与c不同，python通过缩进来确定，代码块的归属关系。

**3 if else语句**

语法：

if 条件：

满足条件时要做的事。

else：

不满足条件时要做的事，同if一样，代码块，需要4个空格( tab键 )作为缩进。

**4 多条件判断语句**

语法：在python架构下，else换成 elif，最终以else结尾。if和elif之间是“或”的关系。

注意：1 多条件判断下，条件之间是**互斥**的，若if满足条件，elif不再发挥作用。

2 可以在条件判断中，直接写input语句，节省代码量。

**5 判断语句的嵌套**

同c++，嵌套的关键点，在于：空格缩进，通过空格缩进，来决定语句之间的：层次关系。

**6 while循环**

语法：while 条件：

条件满足，做的事情

同判断语句，都需要设置空格缩进。

print语句：输出自动换行，不换行语法：末端加end=’’。print ( )，print空内容，就是输出一个换行。

**7 for循环**

for循环是一种“轮询”机制，是对一批内容进行“逐个处理”。

for循环基础语法：for 临时变量 in 待处理数据集（序列）：

循环满足条件时执行的代码。

同while循环不同，for循环是无法定义循环条件的，只能从被处理的数据集中，依次取出内容进行处理，所以，理论上，python中的for循环无法构建无限循环。

注意：

1 for循环无法定义循环条件，只能被动取出数据处理。

2 循环内的语句，需要有空格缩机。

**8 range语句**

for循环中待处理数据集，称之为：序列类型。

序列类型指，其内容可以一个个依次取出的一种类型，包括：字符串、列表、元组。

range语句作用：获得一个简单的数字序列。

语法1：range ( num )

获取一个从0开始，到num结束的数字（整数）序列（不含num本身）

如：range ( 5 ) 取得的数据是：{ 0，1，2，3，4 }

语法2：range ( num1，num2 )

获得一个从num开始，到num2结束的数字序列（不含num2本身）

语法3：range ( num1，num2，step )

获得一个从num1开始，到num2结束的数字序列（不含num2本身），step表示数字之间的步长。

**9 break与continue**

略。

**第三章**

**1 初见函数**

len函数：统计数据、容器的长度。

**2 函数的定义**

语法：def 函数名（传入参数）：

函数体

return 返回值

注意：参数如不需要，可以省略；返回值如不需要，可以省略。

**3 函数的参数**

传入参数的功能：在函数进行计算的时候，接受外部（调用时）提供的数据。

参数之间使用逗号进行分割。

实参与形参：同c。

**4 函数的返回值**

语法：同c。

如果函数没有使用return语句返回数据，函数也有返回值。无返回值的函数，实际上就是返回了：None（空）这个字面量。

None的应用：在if判断中，None等同于False，一般用于在函数中主动返回None，配合if判断做相关处理。

python支持：“if not False：”这种表达。

**5 函数的说明文档**

实际上，就是给函数做注释。

**6 函数的嵌套调用**

所谓函数嵌套调用指的是一个函数里面又调用另外一个函数。

**7 变量的作用域**

变量作用域指的是变量的作用范围（变量在哪里可用，在哪里不可用）。

**局部变量与全局变量**

局部变量：定义在函数体内部的变量，即只在函数体内部生效。

全局变量：所谓全局变量，指的是在函数体内、外都能生效的变量。

global关键字：使用global关键字可以在函数内部声明变量为全局变量，在函数体内部定义的变量，都是局部变量，故用global关键字将其转换为全局变量。

**第四章**

**1 数据容器入门**

数据容器：一种可以容纳多份数据的数据类型。

数据容器根据特点的不同，如：

是否支持重复元素；是否可以修改；是否有序，等。

分为5类，分别是：列表，元组，字符串，集合，字典。

**2 列表（list）**

**2.1 列表的定义**

基本语法：变量名称 = [ 元素1，元素2，元素3… ]

定义空列表：变量名称 = [ ]；变量名称 = list ( )，空列表中元素个数为0。

注意：列表可以一次存储多个数据，且可以为不同的数据类型（元素的数据类型没有任何限制），支持嵌套。

**2.2 列表的下表索引**

访问列表中的元素：正向：同数组；反向（从后向前）：-1，-2，-3，…，-N，不可超出范围。

如果列表是嵌套的列表，同样支持下标索引（形式同二维数组）。

**2.3 列表的常用操作（方法）**

在python中，如果讲函数定义为class的成员，那么函数会称之为：方法。

1 查找某元素的下标

功能：查找指定元素在列表的下标，如果找不到，报错ValueError。

语法：列表.index (元素)，index就是列表对象（变量）内置的方法（函数）。

2 列表的修改

语法：列表[下标] = 值。

3 插入元素

语法：列表.insert (下标，元素)，在指定的下标位置，插入指定的元素。

4 追加单个元素

语法：列表.append (元素)，在尾部追加一个元素。

5 追加多个元素

语法：列表.extend (其他数据容器)，将其它数据容器的**内容**取出，依次追加到列表尾部。

6 删除元素

语法1：del列表[下标]；**语法2：**列表.pop (下标)。

**语法2：**移除、取出元素，即可以用变量去接收取出的数据。

语法3：列表.remove (元素)，删除某元素在列表中第一个匹配项。

语法4：清空列表：列表.clear ()。

7 统计某元素在列表内的数量

语法：列表.count (元素)。

8 统计列表内元素个数

语法：len (列表)。

列表的特点：有顺序，善包容、能重复、可修改。

注意：双引号，单引号，交替出现，外单内双 / 外双内单。

**3 元组（tuple）**

**3.1 元组的定义**

元组是只读的list。

语法：定义元组使用小括号，且使用逗号隔开各个数据，数据可以是不同的数据类型。

定义单个元素的元组：name = (“hello”，)；name = (“hello”)，此时name是字符串类型。

访问元组中的元素：下标索引（同list）。

**3.2 元组的常用操作（方法）**

1 查找某元素的下标：index。

2 统计某元素在元组内的数量：count。

3 统计元组内元素个数：len。

注意：元组内的列表，列表中的元素可以修改。

**4 字符串（str）**

**4. 1 字符串的定义**

字符串是字符的容器，一个字符串可以存放任意数量的字符。

同元组一样，字符串是一个：无法修改的数据容器。

**4.2 字符串的常用操作（方法）**

0 index、count、len，同元组。

1 字符串的替换

语法：字符串.replace (字符串1，字符串2)

功能：将字符串内的全部：字符串1，替换为字符串2。

注意：不是修改字符串本身，而是得到了一个新字符串，本质上来说，是相当于，它会得到一个返回值。

2 字符串的分割

语法：字符串.split ( 分割符字符串 )

功能：按照指定的分割符字符串，将字符串划分为多个字符串，并存入列表对象中。

注意：字符串本身不变，而是得到了一个列表对象。

例如：通过空格分割字符串。

name = "hello world"

namelist = name.split(" ")

print(namelist)

3 字符串的规整操作（去前后空格）

语法：字符串.strip ( )

4 字符串的规整操作（去前后指定字符串）

语法：字符串.strip ( 字符串 )

注意，传入的是“12”其实就是：“1”和“2”都会移除，是按照单个字符。

**5 序列**

**5.1 序列的定义**

序列是指：内容连续、有序，可使用下标索引的一类数据容器，列表、元组、字符串，均可以视作序列。

**5.2 序列的切片**

切片：从一个序列中，取出一个子序列。

语法：序列 [ 起始下标：结束下标：步长 ]

表示从序列中，从指定位置开始，依次取出元素，到指定位置结束，得到一个新序列。

细节：

起始下标表示从何处开始，可以留空（省略），留空视作从头（zero）开始。

结束下标（不含）表示何处结束，可以留空（省略），留空视作截取到结尾。

步长表示，依次取元素的间隔。

步长1表示，一个个取元素，步长默认为1，可以省略。

步长2表示，每次跳过一个元素取。

步长-1表示，反向取（注意，起始下标和结束下标也要反向标记）。

注意：1 此操作不会影响序列本身，而是会得到一个新的序列。

2 可以进行连续切片。

**6 集合（set）**

**6.1 集合定义**

集合的使用场景：无重复元素。

集合特点：能装万物，不支持元素的重复，内部无序。

基本语法：用{ }表示，其他同元组、列表，空集合定义：变量名称 = set ( )。

**6.2 集合的常用操作**

由于无序，集合不支持：下标索引访问。

1 添加元素

语法：集合.add ( 元素 )。将指定元素，添加到集合内，直接对集合操作，无返回值。

2 移除元素

语法：集合.remove ( 元素 )，将指定元素，从集合内移除，直接对集合操作，无返回值。

3 随机取出一个元素

语法：集合.pop ( )，功能：从集合中随机取出一个元素。

结果：会得到一个元素的结果，同时，集合本身被修改，元素被移除。

4 清空集合

语法：集合.clear ( )。

5 取出2个集合的差集

语法：集合1.difference ( 集合2 )，功能：取出集合1和集合2的差集（集合1有而集合2没有的）。

结果：得到一个新集合，集合1和集合2不变。

6 消除2个集合的差集

语法：集合1.difference\_update ( 集合2 )，功能：对比集合1和集合2，在集合1内，删除和集合2相同的元素。

结果：集合1被修改，集合2不变。

**6.3 2个集合合并**

语法：集合1.union ( 集合2 )，功能：将集合1和集合2组合成新集合。

结果：得到新集合，集合1和集合2不变。

**6.4 统计集合元素数量**

语法：len ( 集合 )，有返回值，使用时，要用变量接收。

**6.5 集合的遍历**

集合不支持下标索引，不能用while循环，可以用for循环。

**7 字典**

**7.1 字典的定义**

作用：通过key找到value，从而达到一一对应的目的，就像查字典一样。

语法：同样使用{ }，不过存储的元素是一个个的：键值对，见下。

{ key：value，

key：value，

}

定义空字典：字典名 = { }，字典名 = dict ( )。

字典中的key不可以重复，字典同集合一样，不可以使用下标索引。

通过key值来访问value：字典名[ key ]。

字典的key和value可以是任意数据类型（key不可以为字典）。

**7.2 字典的嵌套**

洋葱，访问：剥洋葱。

**7.3 字典的常用操作**

1 新增元素

语法：字典 [ key ] = value，结果：字典被修改，新增了元素。

2 更新元素

语法：字典 [ key ] = value，结果：字典被修改，元素被更新。

3 删除（取出）元素

语法：字典.pop ( key )，结果：获得指定key的value（有返回值），同时字典被修改，指定key的数据被删除。

4 清空元素

语法：字典.clear ( )。

5 获取全部的key

语法：字典.keys( )，结果：得到字典中全部key。

6 遍历字典

循环：语法：for key in dict：，遍历dict中所有的key。

7 统计字典内的元素数量

语法：len ( )，有返回值。



**8 数据容器的通用操作**

1 遍历，都支持for循环遍历。

2 统计，len，max（得到最大元素），min（得到最小元素），均有返回值。

3 转换，类型转换：list( 容器名 )，str( 容器名 )，tuple( 容器名 )，set( 容器名 )。

注意：字符串转列表：得到单个字符。字典转列表：得到key，去掉value。字典转字符串：均保留。字典转集合：得到key，去掉value。，转集合：去重复，得无序。

4 排序，语法：sorted ( 容器，reverse（默认值False）= True )，将给定容器进行排序，正序 / 逆序排序结果变为列表对象。

**9 字符串比较**

规则：按位比较，一位一位进行比对，只要有一位大，整体就大。

**第五章**

**1 函数多返回值**

函数内语法：return 1，2 / 函数外语法：x，y = test ( )。

注意：支持不同类型的数据return，变量之间用逗号隔开。

**2 函数多种传参方式**

**3 函数作为参数传递**

参数类型：

1 数字、字符串、字典、列表、元组等，数据作为参数传入。

2 函数本身作为参数传入。

语法：函数名1 ( 函数名2 )

**4 lambda匿名函数**

无名称的匿名函数，只可临时使用一次。

**5 模块 ( Module )**

每个python文件都可以作为一个模块，模块的名字就是文件的名字，语法：import module，好处不必多言。

