2019-8-6

* 1. **变量**
     1. >>> A=[1,2,3,4]

>>> print(A)

[1, 2, 3, 4]

>>> A\*3

[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]

>>> B=[8,6,4]

>>> A\*4+B

[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 8, 6, 4]

* + 1. 变量名区分大小写
    2. 变量的命名以数字，字母，下划线，但不能以数字开头
    3. 系统保留关键字不能用作变量名
    4. 变量没有类型限制
    5. Int str tuple(不可改变) 值类型 list 引用类型
    6. 函数：id() 获取变量的内存地址
    7. 函数：hex() 获取16进制的数据
       1. >>> a=[12,56,88]

>>> id(a)

50499536

>>> hex(id(a))

'0x3028fd0'

* 1. 运算符
     1. 算数运算符
        1. 加 +
        2. 减 –
        3. 乘 \*
        4. 除 /
        5. 求整 //
        6. 求余 %
        7. 乘方 \*\*
     2. 赋值运算符
        1. 赋值 =
        2. +=
        3. -=

…

2019-8-7

* + 1. 比较运算符
       1. 单个字符串比较的是他们的阿斯克码值
       2. 集合和集合，列表和列表都可以比较，一一对应比较
          1. >>> ord('d')

100

>>> [1,23,1]>[4,5,6]

False

>>> (7,5,6)<(8,5,4)

True

* + 1. 逻辑运算符
       1. **and** 需要都要判断，若两个值都存在返回最后判断的值，若有一个值为空在则返回空表示的那个值，若都表示为空则返回第一个
          1. [] and [4,5]

[]

>>> 'c' and []

[]

>>> [] and '2'

[]

>>> 'd' and ''

''

>>> [] and ''

[]

* + - 1. or 若两个变量都要值则返回第一个，若一个值为空一个有值则返回有值的，若都为空则返回第二个空对象
         1. >>> [] or 'd'

'd'

>>> [] or ''

''

>>> 5 or 4

5

* + - 1. not

>>> not 4

False

>>> not []

True

>>> not [1,5]

False

* + - 1. 只要数据不为空其他任意变量都为true,否则为false
    1. 成员运算符
       1. in
       2. not in
          1. >>> a=1

>>> a in [1,2,3]

True

>>> b= '5'

>>> b not in (4,'5',4)

False

>>> b not in {4,'5',4}

* + - * 1. False
      1. 比较字典的时候，比较的是字典的key值
         1. >>> c=5

>>> c in {'2':4,5:4}

* + - * 1. True
    1. 身份运算符
       1. is
          1. >>> 1 is 2

False

>>> a=4

>>> b=4

>>> a is b

True

>>> 1 == 1.0

True

>>> a is 1.0

False

* + - * 1. 思考

>>> a ={1,2,3}

>>> b ={3,2,1}

>>> c =(1,2,3)

>>> d =(3,2,1)

>>> a==b

True //集合是无序的只要值相同就为true

>>> a is b

False //比较的是地址

>>> c ==d

False //元组是有序的，只要顺序不相同就为false

>>> c is d

False //比较的是地址

* + - 1. **isinstance()** 函数方法比较变量的类型
         1. >>> a= 5

>>> type(a) == int

True

>>> isinstance(a,str)

False

>>> isinstance(a,int)

True

>>> isinstance(a,(int,float,str))

True //只要该类型存在在元组里，则为true

* + 1. 位运算符
       1. & 按位与 二进制比较 都为1则为1,只要有0则为0
       2. | 按位或 二进制比较 只要有1则为1，都为0则为0
       3. ^ 按位异或
       4. ~ 按位取反
       5. << 左移动
       6. >> 右移动
       7. 把比较的变量都转换为二进制进行比较