|  |  |  |  |
| --- | --- | --- | --- |
| Name | Def. | Usage | Tips |
| data types | Number（整数，浮点）  String  Boolean | %d; %g，0.2f  %s  True/False | 容易写错 |
| if else | 判断+执行 | if True/False:  statement a  else:  statement b | 冒号/缩进 |
| !=, == ; =  \* , \*\*  / , // , % | 判断, 赋值  乘，平方  除，整，余 |  | 判断与赋值易混  余数可用于判断奇偶 |
| return | 返回主机供其他模块调用 | 返回  return  调用  def main():  variable = function1()  function2(variable) |  |
| 数据结构 | List  (item，index) | 创建  lst = [ ]  显示  list \ list[index]  增减/修改  lst.append() / lst += [item]  lst[index]=’item’  lst.remove(item)/ del(lst[index])  切片  lst[a,b]  排序  lst.reverse()  lst.sort() | 长度 len(lst)  index从0开始  切片 [a,b) |
| Tuple | 创建  tup = ()  赋值  a,b,c,d = (1,2,3,4)  查找  xx in tup  切片  tup[a:b] | 无修改\插入\删除  index从0开始  切片 [a,b) |
| Dictionary | 创建  dict= {}  增减/修改  dict ['key'] = 'value';  del dict['key']  显示  dict [‘key’]  dict.keys()/dict.values()  查找  dict.has\_key('key') | 长度 len(lst) |
|  | 遍历 | 应用示范  for index, item in enumerate(lst/tup):  for key in dict.keys(): |  |
| try  except | 试错 | raise Exception(e)  try:  except Exception, e:  print e | 对于自定义错误先Raise再try |
| library | 库 | import | 查询使用  [http://devdocs.io](http://devdocs.io/) |
| class |  | 定义+初始化  class Class1():  def\_\_init\_\_(self, tribute1, tribute2):  self.tribute1 = tribute1  self.tribute2 = tribute2  def function(self):  调用  def main():  self=Class(tribute1, tribute2)  self.function(tribute\_x) | 类的首字母大写  面向对象效率高 |