

# 数据分析基础与Jupyter

---

## 一，安装Jupyter

`pip install --upgrade jupyter`

使用Pip进行安装

`jupyter notebook`

开启服务

此时你已经到了jupyter页面

## 二，Python基础数字

普通加法运算

`1 + 1` 运行后得 2

`1 + 2` 运行后得 3

`3 + 1` 运行后得 4

普通乘法运算

`2 * 8` 运行后得 16

赋值

`a = 5` a的值为 5

`a * a` 运行后得 5

`a = a + 3` 运行后得8，3赋值给了a，然后`a3 + a5` 得8

`b = 3`

`a * b` 运行后得24

`b = a * b` 则b = 24

普通除法运算

`24 / 3` 其运行结果为 8.0

`24 // 3` 其运行结果为 8

`25 / 3` 其运行结果为 8.333333333333334

`25 // 3` 其运行结果为 8

于的运算

`3 % 2` 取于为 1

`b % a` 取于为 0

## 三，字符串

双引号，单引号，三引号的区别

`"jscsd"` 其输出结果为 `'jscsd'`

`'jscsd'` 其输出结果为 `'jscsd'`

`"""`

`jcsd`

`jcsd`

`jcsd`

`"""`

其输出结果为 `'\n jcsd\n jcsd\n jcsd\n'` (\n 在计算机中是代表换行的意思，在Java,Python等语言中都是用\n来定义换行的。)

len

len(a) 其已被赋值为 jscsd,其用Len计算总长度为 5

[]

通过索引获取字符串中字符，如将a赋值为jscsd,其a[0]，输出结果为 j

[:]

截取字符串中的一部分，如a[0:4]，其输出结果为 jscs

count()

主要计算在字符串中某一个字符串的数量如 b.count('s')，其输出结果为2

upper()

转换字符串中的大小写，如 c = b.upper()，此时b 的字符串保持原样为jscsd,而c字符串中的大小写为 JSCSD。

c.count('s')，此时使用c.count检索s的数量的时候显示为0，是因为在python语言中区分大小写的，毕竟是一门优雅高贵的语言

而此时需要修改为 c.count('S')，则正常显示为 S的数量为2

lower()

字符串可以转为大写，当然也可以使用lower()转换为小写，此时的c的字符串已经为小写的jscsd了。

如何判断大写和小写的字符串是否一致？

我们可以使用 b == c.lower() 来判断大小写的字符串内容是否一致，如果是则 True，不过不是则 False

<>

大于号和小于号可以用来判断大小，比如  $1 > 2$ ，其返回结果为False，因为1不可能大于2

而小于号可以  $2 < 1$ ，其返回结果也是False因为2不可能小于1。

而  $2 > 1$  则返回结果为True，因为2是大于1的

$1 < 2$  返回结果也是True，因为1的确是小于2

split

当你有特殊需要的时候，可以用split进行分割，比如我们拿域名为例，[www.jxfox.org.cn](http://www.jxfox.org.cn)如何分呢，可以使用split 进行分割

d = '[www.jxfox.org.cn](http://www.jxfox.org.cn)' 首先需要对d进行赋值

然后使用 d.split('.') 进行分割，以 . 来分，其输出结果为 ['www', 'jxfox', 'org', 'cn']

如果你不设置d.split("")的时候，他默认为 空格，比如你赋值h为 z s d k，此时他将会以空格为一格子进行分割，其输出结果为 ['z', 's', 'd', 'k']

isdigit

使用isdigit判断字符串是否是数字类型的可以使用aa.isdigit()，当前aa被赋值的内容为 123，侧最后显示结果为True

可以和ss作为对比，ss的赋值内容为zsdk，则使用ss.isdigit为Fales，因为他不是数字类型的，是字母类型的。

既然可以判断数字，也可以使用isalpha() 判断字母

赋值给ss为zsdk，使用ss.isalpha()判断则输出为True， 而b赋值的内容为123，aa.isalpha()输出结果为False，因为他不是字母。

int

int可以用于数字和字符串之间的转换，比如将aaa赋值为123456789，使用int(aaa)其输出结果为123456789，而普通的输出则为'12345689'.

str

str也可以用于字符和数字之间的转换比如str(aaa)，此时的aaa内容为123456789的字符串，可以使用str(aaa)转换为数字 此时输出结果为'12345689'，已经成为了数字

#### 四，列表

[]

定义一个列表还是非常简单的，毕竟是基础部分，我们可以使用 a = [] 定义一个空的列表

a = [1,2,3, 'jscsd'] 其输出结果为 [1, 2, 3, 'jscsd']

+ 的作用

我们首先搞出来两个列表 a = [1,2,3]，b = [4,5,6]，然后使用 + 号把他们合并起来 a + b输出结果为 [1, 2, 3, 4, 5, 6]

但是你如果想把这些输出结果放到另一个列表里面可以使用 =

比如 g = a + b，将a+d数据合并到g里面，其g内容为 [1, 2, 3, 4, 5, 6]

既然可以合并也可以增加，那么可以使用b.append(7)，在b列表里面增加一个7则输出结果为[4, 5, 6, 7]

append()

b.append(a) 这个关键自动意思就是把a套在b中，其输出结果为[4, 5, 6, 7, [1, 2, 3]]

使用这个关键字的唯一好处是，在b[0]的时候输出结果是4，而b[4]的时候输出的是a的列表[1, 2, 3]

extend

当你需要吧两个列表合并的话，可以使用他，比如

t = [456,789]，r = [852,147] 然后使用t.extend(r) 进行合并结果为[456, 789, 852, 147]

pop

pop可以快速的弹出你不想要的东西，比如h.pop(0)，则14给弹出了，h原有的列表是14,25,36，而pop后则剩下 [25, 36]

remove

remove也可以和pop实现一个操作，其案列为 h.remove(25)，其h原有的列表为 [25, 36] remove 后为 [36]

clear

clear这个关键字就非常的好，他可以清空你列表所有字段，比如清空 u 其列表信息为 [1,1,1,1,1,8,1,4,5,]

使用u.clear() 后变空 为[] 空了

#### 五，字典

字典就是关键词dict，在定义的时候参数是自定义的，比如

jx = {}，赋值可以使用 jx['f'] = 'x' 然后查看jx内容则发现输出的内容为一对出现的，列表中为单个的。

字典可以通过名称来取得对应的值，且值是可以重复的，但键 是唯一的。

jx.keys()

输出键名

jx.values()

输出所有值

`jx.items()`

输出成对出现的内容

Json

Json指的是一种字符串的格式。

`json.dumps(jx)`

导出，变为字符串格式

`type(json.dumps(jx))`

查看jx格式为str

`json_str = json.dumps(jx)`

导出为字符串格式

`json.loads(json_str)`

导出为字典格式