



大数据成就未来



Python基础知识

张敏

18/1/2

目录

| | |
|---|--------------|
| 1 | Python固定语法 |
| 2 | Python基础数据类型 |
| 3 | 文件操作 |
| 4 | 函数 |



Python固定语法

编写第一个Python程序

- `str = 'hello world'`
- `file = open('/Users/zhangmin/Desktop/helloworld.txt','w')`
- `file.write(str)`
- `file.close()`
- Python 中的变量不需要声明。每个变量在使用前都必须赋值，变量赋值以后该变量才会被创建。
- 变量一般由字母、数字和下划线组成；通常第一个字符是字母或下划线 ‘_’ ；区分大小写。



Python固定语法

编写第一个Python程序

- 机器学习 = ['决策树','神经网络','聚类分析']
for 算法 in 机器学习:
 print(算法)
- Python 3 源码文件以 UTF-8 编码，所有字符串都是 unicode 字符串。
- 或在脚本首行指定编码方式：
• *# -*- coding: GB18030 -*-*
- 语法格式：代码块以Tab或空格缩进限制，而不以花括号” {}” 等分割代码



Python固定语法

编写第一个Python程序

- `A,b,c = 1,2,"python"`
- `a = b = c = 1` #多变量赋值
- 单行注释: `#我是注释`
- 多行注释: `"""多行
注释"""`

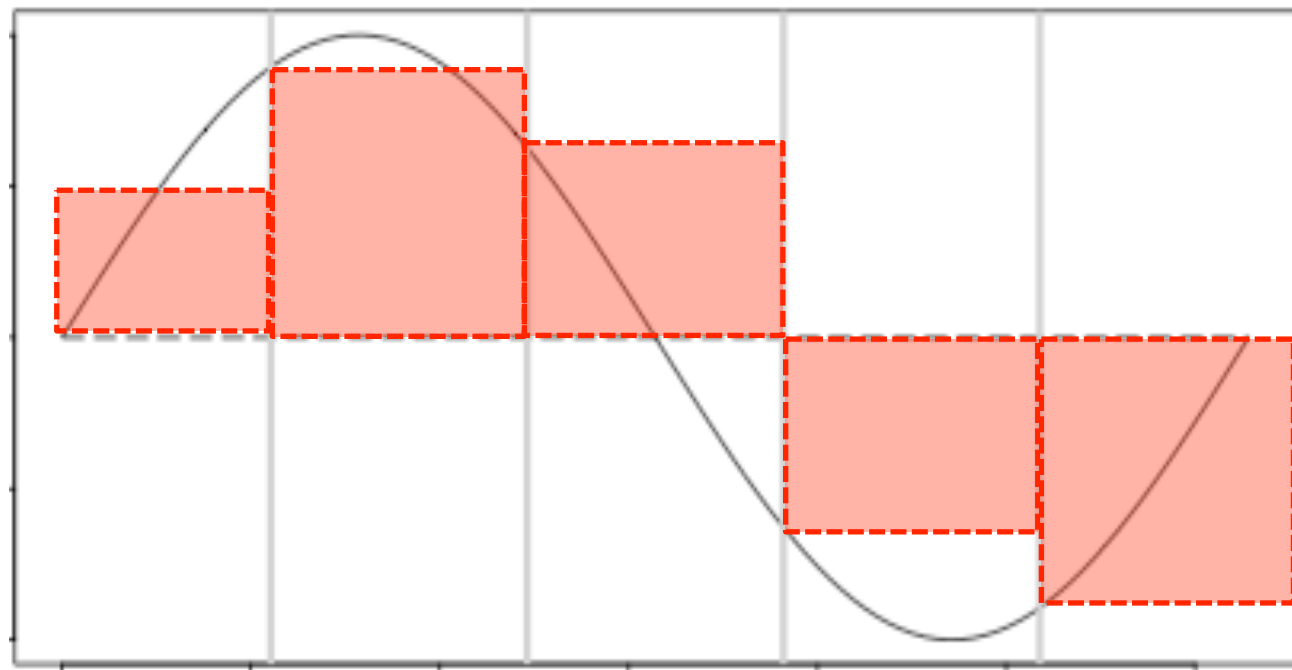
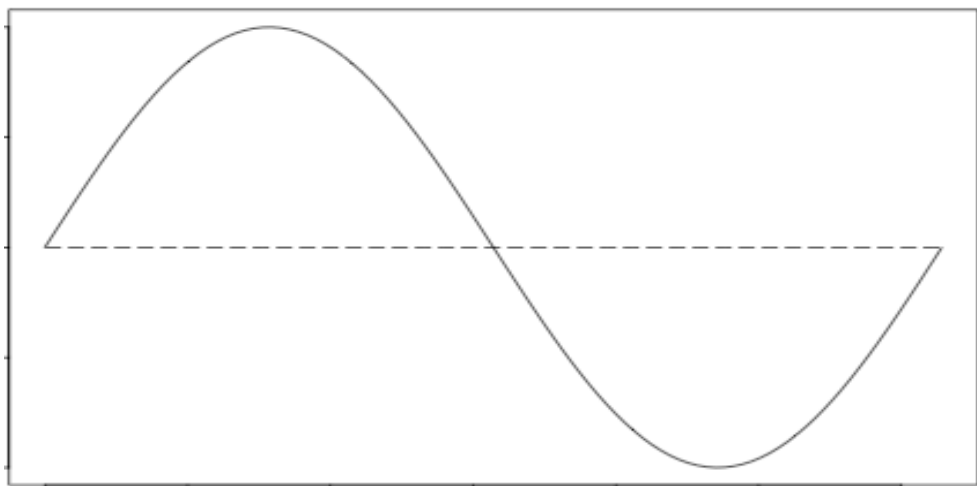


目录



Python基础数据类型

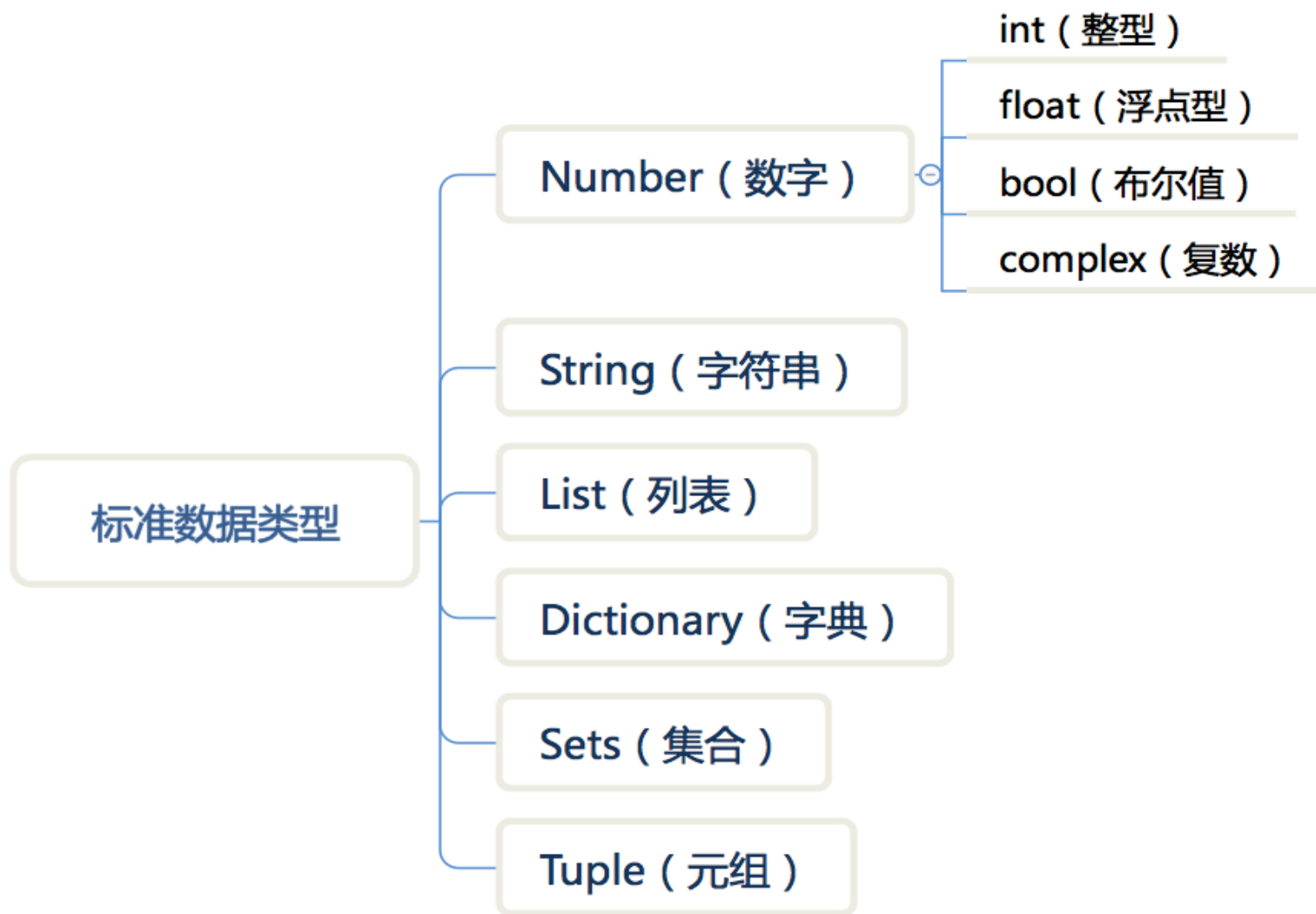
练习1:求 $y=\sin(x)$ 从0到 2π ，与x轴围成的面积



如何用Python中的数据类型 / 结构表示和存放相应数据？



Python基础数据类型



Python基础数据类型

Number (数字)

- `var1 = 1`
- `var2 = 10.3`
- `del var1`
- `x,y = True,False` #布尔值
- 数值不可变，
- 整型无长短之分
- 取绝对值：`abs(-1)`
- 数学函数：`math`包，如`math.sin(1.5)`
- 随机函数：`random`包，如`random.choice(range(10))`
-



Python基础数据类型

| 函数 | 返回值（描述） |
|--|---|
| <u>abs(x)</u> | 返回数字的绝对值，如abs(-10) 返回 10 |
| <u>ceil(x)</u> | 返回数字的上入整数，如math.ceil(4.1) 返回 5 |
| <u>exp(x)</u> | 返回e的x次幂(e^x),如math.exp(1) 返回2.718281828459045 |
| <u>fabs(x)</u> | 返回数字的绝对值，如math.fabs(-10) 返回10.0 |
| <u>floor(x)</u> | 返回数字的下舍整数，如math.floor(4.9)返回 4 |
| <u>log(x)</u> | 如math.log(math.e)返回1.0,math.log(100,10)返回2.0 |
| <u>log10(x)</u> | 返回以10为基数的x的对数，如math.log10(100)返回 2.0 |
| <u>max(x1, x2,...)</u> | 返回给定参数的最大值，参数可以为序列。 |
| <u>min(x1, x2,...)</u> | 返回给定参数的最小值，参数可以为序列。 |
| <u>modf(x)</u> | 返回x的整数部分与小数部分，两部分的数值符号与x相同，整数部分以浮点型表示。 |
| <u>pow(x, y)</u> | $x**y$ 运算后的值。 |
| <u>round(x [,n])</u> | 返回浮点数x的四舍五入值，如给出n值，则代表舍入到小数点后的位数。 |
| <u>sqrt(x)</u> | 返回数字x的平方根，返回类型为实数，如math.sqrt(4)返回 2+0j |



Python基础数据类型

列表 (list)

- `all_in_list = [`
- `1,`
- `'a word' ,`
- `print(1),`
- `True,`
- `[1,2],`
- `]`
- 1、列表中每个元素都是可变的；
- 2、列表中的元素是有序的，每个元素对应一个位置；
- 3、列表可以容纳Python中的任何对象。



Python基础数据类型

列表的增删改查

- `fruit = ['pineapple','pear']`
- `fruit.insert(1,'grape')`
- `fruit.insert(-1,'apple')`
- `fruit.remove('grape')`
- `fruit.remove(fruit[0])`
- `del fruit[0:2]`

| | | | | | | |
|----|-----|------|----|----|----|----|
| a | 101 | True | L | 0 | n | F |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| -7 | -6 | -5 | -4 | -3 | -2 | -1 |



Python基础数据类型

列表的方法

| 名称 | 函数说明 |
|-------------------------|-----------------------------------|
| list.append(obj) | 在列表末尾添加新的对象 |
| list.count(obj) | 统计某个元素在列表中出现的次数 |
| list.extend(seq) | 在列表末尾一次性追加另一个序列中的多个值（用新列表扩展原来的列表） |
| list.index(obj) | 从列表中找出某个值第一个匹配项的索引位置 |
| list.insert(index, obj) | 将对象插入列表 |
| list.pop(obj=list[-1]) | 移除列表中的一个元素（默认最后一个元素），并且返回该元素的值 |
| list.remove(obj) | 移除列表中某个值的第一个匹配项 |



Python基础数据类型

for循环



```
List = [2,'a','列表',3.4]
```

```
for i in List:
```

```
    print(i)
```

```
for i in range(10):
```

```
    print(i)
```



Python基础数据类型

列表推导式

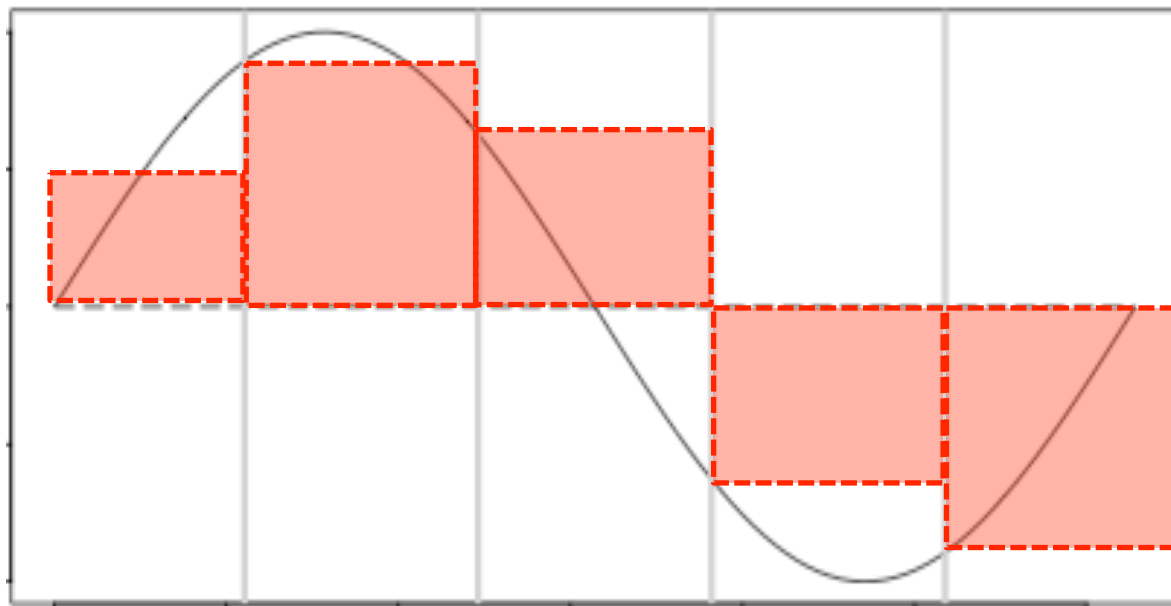
- `a = []`
- `for i in range(1,11):`
- `a.append(i)`
- `b = [i for i in range(1,11)]`
- `a = [i**2 for i in range(1,10)]`
- `c = [j+1 for j in range(1,10)]`



Python基础数据类型

练习1:求 $y=\sin(x)$ 从0到 2π ，与x轴围成的面积

1. 将各小矩形的高度存放至一列表中
2. 将各高度乘以宽度，得各矩形面积
3. 求和



Python基础数据类型

Python常用操作符

- 算术操作符
- 赋值操作符
- 比较操作符
- 逻辑操作符

算术操作符一般会返回一个数，而比较和逻辑操作符会返回布尔值True或False。



Python基础数据类型

算术操作符

| 操作符 | 描述 | 实例 |
|-----|---------------------|-------------------|
| + | 加法-返回两操作数相加的结果 | 3+2返回5 |
| - | 减法-返回左操作数减去右操作数的结果 | 3-2返回1 |
| * | 乘法-返回两操作数相乘的结果 | 3*2返回6 |
| / | 除法-返回右操作数除左操作数的结果 | 3/2返回1但3.0/2返回1.5 |
| % | 模-返回右操作数对左操作数取模的结果 | 5%3返回2 |
| ** | 指数-执行对操作指数的计算 | 3**2返回9 |
| // | 取商-返回右操作数对左操作数取商的结果 | 3.0//2返回1.0 |



Python基础数据类型

赋值操作符

| 操作符 | 描述 | 例子 |
|-----|---------------------------------------|---------------------------|
| = | 简单的赋值运算符，赋值从右侧操作数左侧操作数 | c=a+b将 a和b相加的值赋值给 c |
| += | 加法AND赋值操作符，它增加了右操作数左操作数和结果赋给左操作数 | c += a 相当于 c = c + a |
| -= | 减法AND赋值操作符，它减去右边的操作数从左边操作数，并将结果赋给左操作数 | c -= a 相当于 c = c - a |
| *= | 乘法AND赋值操作符，它乘以右边的操作数与左操作数，并将结果赋给左操作数 | c *= a 相当于 c = c * a |
| /= | 除法AND赋值操作符，它把左操作数与正确的操作数，并将结果赋给左操作数 | c /= a 相当于c = c / a |
| %= | 模量AND赋值操作符，它需要使用两个操作数的模量和分配结果左操作数 | c %= a相当于c = c % a |
| **= | 指数AND赋值运算符，执行指数（功率）计算操作符和赋值给左操作数 | c **= a 相当于 c = c ** a |
| //= | 取商，并分配一个值，执行取商并将结果赋值给左操作数 | c //= a 相当于 c = c // a |



Python基础数据类型

比较操作符

| 操作符 | 描述 | 实例 |
|-----|----------------------------------|-------------|
| == | 如果两个操作数的值相等则返回True，否则返回False | 3==2返回False |
| != | 如果两个操作数的值不等则返回True，否则返回False | 3!=2返回True |
| <> | 与!=效果相同(3.x版本中无此功能) | 3<>2返回True |
| > | 如果左操作数大于右操作数则返回True，否则返回False | 3>2返回True |
| < | 如果左操作数小于右操作数则返回True，否则返回False | 3<2返回False |
| >= | 如果左操作数大于或等于右操作数则返回True，否则返回False | 3>=3返回True |
| <= | 如果左操作数小于或等于右操作数则返回True，否则返回False | 2<=2返回True |

Python基础数据类型

逻辑操作符

| 操作符 | 描述 | 实例 |
|-----|-----------------------------------|-----------------------|
| and | 逻辑与运算符。当且仅当两个操作数为真则返回真，否则返回假。 | True and False返回False |
| or | 逻辑或运算符。当且仅当有两个操作数至少一个为真则返回真，否则返回假 | True or False返回True |
| not | 逻辑非运算符。用于反转操作数的逻辑状态。 | not True 返回False |



Python基础数据类型

逻辑控制

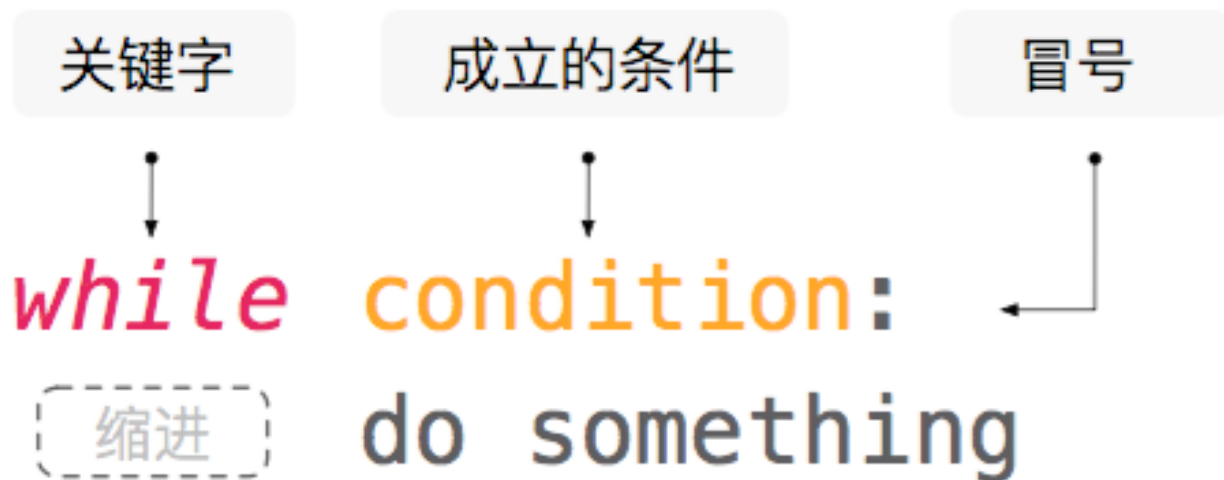
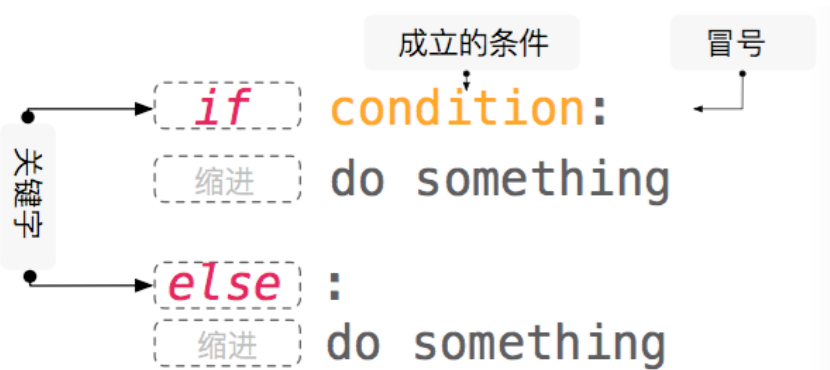
逻辑控制：

- 逻辑判断——True & False
- $1 > 2$
- $1 < 2 < 3$
- $42 \neq 42$
- `'Name' == 'name'`
- `'M' in 'Magic'`
- 返回值为True或False



Python基础数据类型

条件判断与循环



Python基础数据类型

练习2:小说《Walden》单词词频统计

Walden中文译名《瓦尔登湖》，是美国作家梭罗独居瓦尔登湖畔的记录，描绘了他两年多时间里的所见、所闻和所思。该书崇尚简朴生活，热爱大自然的风光，内容丰厚，意义深远，语言生动。请用Python统计小说Walden中各单词出现的频次，并按频次由高到低排序。



Python基础数据类型

字符串

- "任何在这双引号之间的文字"
- "单引号其实和双引号完全一样"
- '''三个引号被用于长段文字
或说明,只要引号不结束,你就可以任意
换行'''
- 字符串属不可变数据类型



Python基础数据类型

字符串基本用法

- 合并：'char1'+'char2'+'char3'
- 重复：'word' * 3
- 转换：int(string)

切片与索引：

- str[0]
- str[-4]
- str[1:4]
- str[3:]
- str[:3]

| | | | | | | |
|----|----|----|----|----|----|----|
| M | y | | n | a | m | e |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| -7 | -6 | -5 | -4 | -3 | -2 | -1 |



Python基础数据类型

字符串的方法

| 名称 | 说明 |
|---|---|
| <code>split(sep="", num=string.count(str))</code> | <code>num=string.count(str))</code> 以 <code>sep</code> 为分隔符截取字符串，如果 <code>num</code> 有指定值，则仅截取 <code>num</code> 个子字符串 |
| <code>S.strip([chars])</code> | 返回字符串的一个副本，删除前导和尾随字符。 <code>chars</code> 参数是一个字符串，指定要移除的字符集。如果省略或为 <code>None</code> ，则 <code>chars</code> 参数默认为删除空白字符。 |
| <code>S.lower()</code> | 将字符串所有大写字符变为小写 |
| <code>S.isalnum()</code> | 如果字符串至少有一个字符，并且所有字符都是数字或者字母，则返回 <code>true</code> ，否则返回 <code>false</code> 。 |
| <code>S.count(sub[,start[,end]])</code> | 返回在 <code>[start, end]</code> 范围内的子串 <code>sub</code> 非重叠出现的次数。可选参数 <code>start</code> 和 <code>end</code> 都以切片表示法解释。 |
| <code>replace(old, new [, max])</code> | 将字符串中的 <code>old</code> 替换成 <code>new</code> ,如果 <code>max</code> 指定，则替换不超过 <code>max</code> 次。 |



Python基础数据类型

字典

- NASDAQ_code = {
- 'BIDU': 'Baidu',
- 'SINA': 'Sina',
- 'YOKU': 'Youku'
- }
- 1、键 - 值成对出现；
- 2、键不能重复；
- 3、键不可更改，值可修改；
- 4、键来索引值。



Python基础数据类型

字典的增删改查

- `NASDAQ_code = {`
- `'BIDU': 'Baidu',`
- `'SINA': 'Sina',`
- `'YOKU': 'Youku'`
- `}`
- `NASDAQ_code['YOJKU'] = 'Yoku'`
- `NASDAQ_code.update({'FB': 'Facebook', 'TSLA': 'Tesla'})`
- `del NASDAQ_code['FB']`
- `NASDAQ_code['FB']`



Python基础数据类型

字典推导式

- `a = []`
- `for i in range(1,11):`
- `a.append(i)`
- `b = [i for i in range(1,11)]`
- `a = [i**2 for i in range(1,10)]`
- `c = [j+1 for j in range(1,10)]`
- `k = [n for n in range(1,10) if n % 2 == 0]`
- `z = [letter.lower() for letter in 'ABCDEFGHIIJKLMN']`



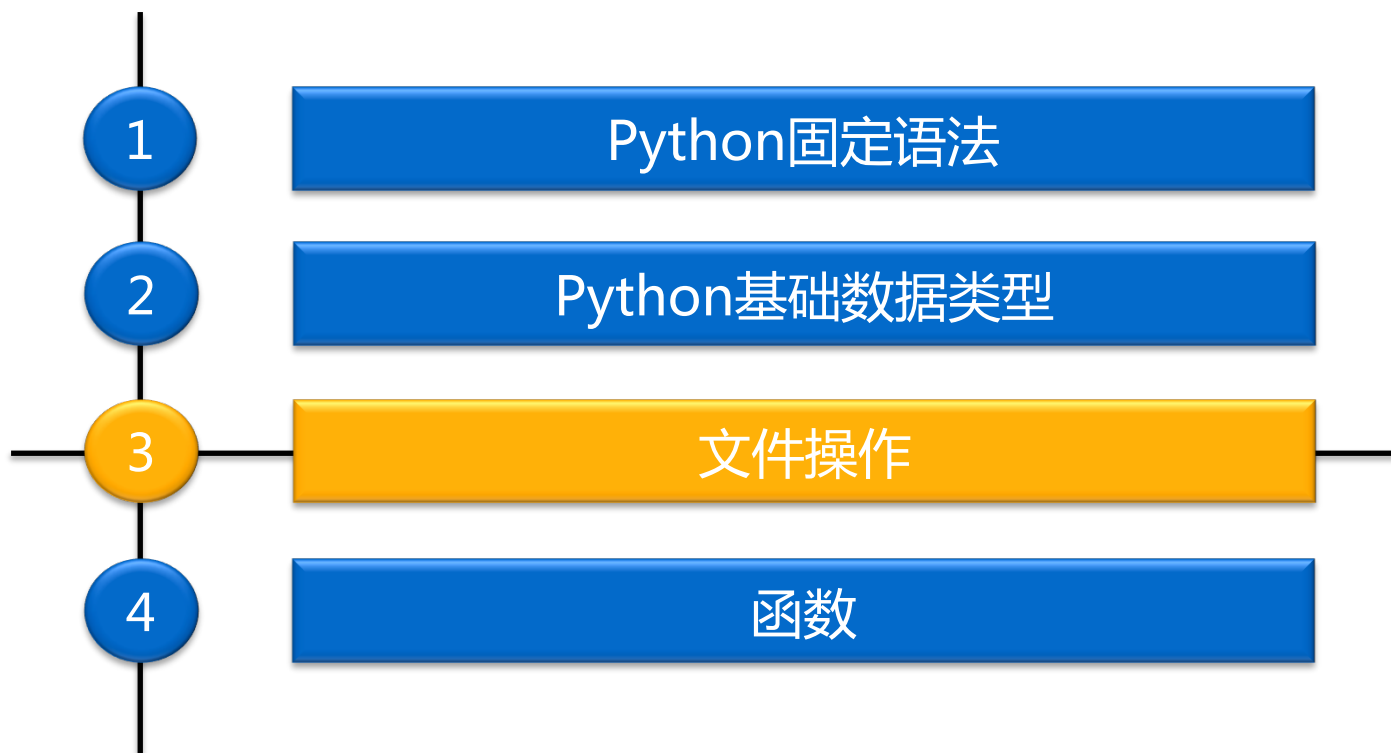
Python基础数据类型

练习2:小说《Walden》单词词频统计

- lyric = 'The night begin to shine, the night begin to shine'
- words = lyric.split()
- print(words)
- words.count(words[1])



目录



文件操作

打开文件

在python，使用open函数，可以打开一个已经存在的文件，或者创建一个新文件：

- open(文件名，访问模式)

示例如下：

- `f = open('test.txt', 'w')`



文件操作

写数据(write)

➤ 使用write()可以完成向文件写入数据

- `f = open('test.txt', 'w')`
- `f.write('hello world,\n')`
- `f.write('i am here!\n')`
- `f.close()`



文件操作

读数据(read)

- 使用read(num)可以从文件中读取数据，num表示要从文件中读取的数据的长度（单位是字节），如果没有传入num，那么就表示读取文件中所有的数据
- `f = open('test.txt', 'r')`
- `content = f.read(5)`
- `print(content)`
- `print("-"*30)`
- `content = f.read()`
- `print(content)`
- `f.close()`



文件操作

读数据 (readlines)

- 就像read没有参数时一样，readlines可以按照行的方式把整个文件中的内容进行一次性读取，并且返回的是一个列表，其中每一行的数据为一个元素
 - `f = open('test.txt', 'r')`
 - `content = f.readlines()`
 - `print(type(content))`



文件操作

打开文件

| 访问模式 | 说明 |
|------|---|
| r | 以只读方式打开文件。文件的指针将会放在文件的开头。这是默认模式。 |
| w | 打开一个文件只用于写入。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。 |
| a | 打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。也就是说，新的内容将会被写入到已有内容之后。如果该文件不存在，创建新文件进行写入。 |
| rb | 以二进制格式打开一个文件用于只读。文件指针将会放在文件的开头。这是默认模式。 |
| wb | 以二进制格式打开一个文件只用于写入。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。 |
| ab | 以二进制格式打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。也就是说，新的内容将会被写入到已有内容之后。如果该文件不存在，创建新文件进行写入。 |



文件操作

打开文件

| 访问模式 | 说明 |
|------|---|
| r+ | 打开一个文件用于读写。文件指针将会放在文件的开头。 |
| w+ | 打开一个文件用于读写。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。 |
| a+ | 打开一个文件用于读写。如果该文件已存在，文件指针将会放在文件的结尾。文件打开时会是追加模式。如果该文件不存在，创建新文件用于读写。 |
| rb+ | 以二进制格式打开一个文件用于读写。文件指针将会放在文件的开头。 |
| wb+ | 以二进制格式打开一个文件用于读写。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。 |
| ab+ | 以二进制格式打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。如果该文件不存在，创建新文件用于读写。 |



文件操作

文件操作方法

| 模式 | 说明 |
|--------------------|--|
| f.close() | 关闭文件，记住用open()打开文件后需得关闭它，否则会占用系统的可打开文件句柄数。 |
| f.flush() | 刷新输出缓存。 |
| f.read([count]) | 读出文件全。若有count值，则读取count个字符 |
| f.readline() | 读出一行信息。 |
| f.readlines | 读出所有行，也就是读出整个文件的信息。 |
| f.write(string) | 把string字符串写入文件。 |
| f.writelines(list) | 把list中的字符串一行一行地写入文件，是连续写入文件，没有换行。 |

文件操作

练习2:小说《Walden》单词词频统计

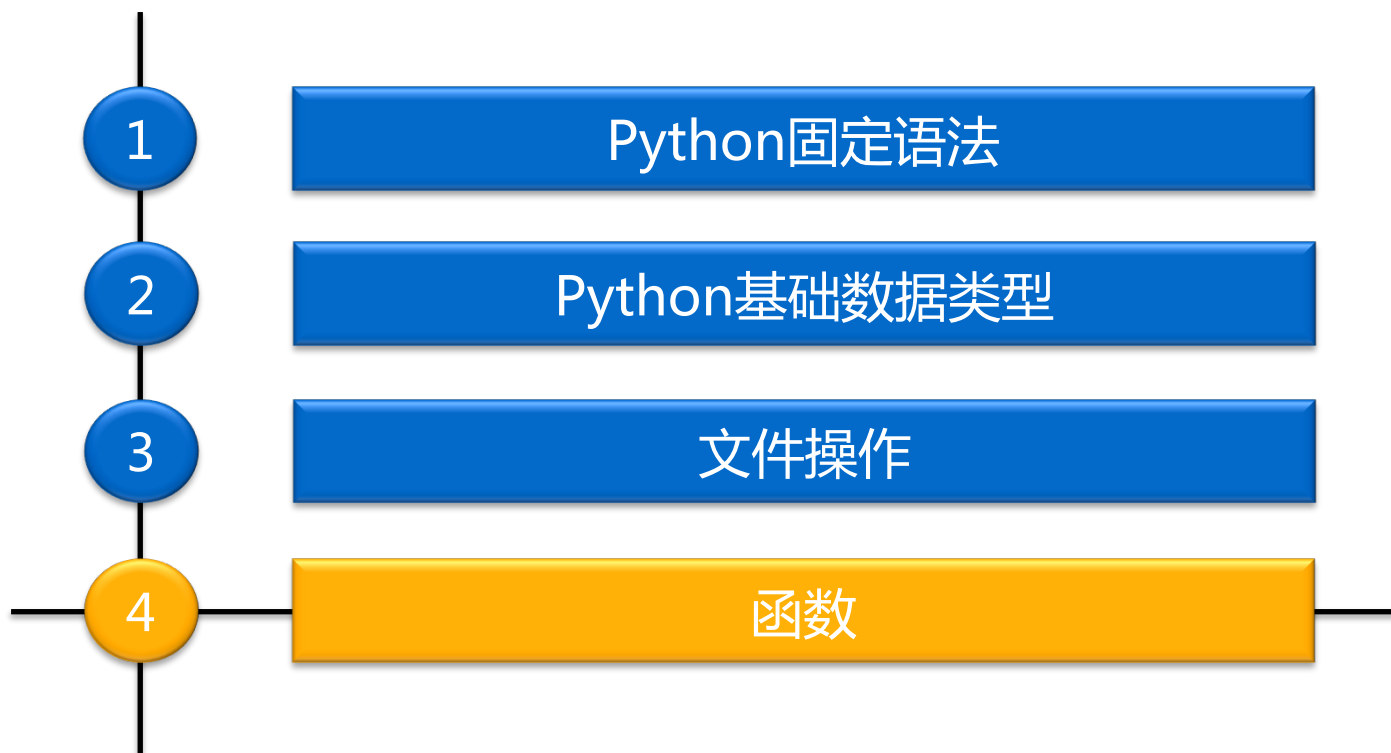
- Walden中文译名《瓦尔登湖》，是美国作家梭罗独居瓦尔登湖畔的记录，描绘了他两年多时间里的所见、所闻和所思。该书崇尚简朴生活，热爱大自然的风光，内容丰厚，意义深远，语言生动。请用Python统计小说Walden中各单词出现的频次，并按频次由高到低排序。

示例：

- lyric = 'The night begin to shine, the night begin to shine'
- words = lyric.split()
- print(words)
- words.count(words[1])



目录



Python函数

内建函数，3.5.0版共68个

- `print()`
- `input()`
- `int()`

方法：限定具体对象

- `'ab,c'.split(',')`
- `split('ab,c')` #name 'split' is not defined
- `['abc'].split()` #'list' object has no attribute 'split'



Python函数

自定义函数

- `def function(x,y):`
 `return 'result'`
- 例：自定义一个函数，能求出一组给定数组中奇数的个数

`y = lambda x:x+1`



练习3:自定义求序列偶数个数的函数



模块和第三方类库

模块

- 模块是一个包含所有你定义的函数和变量的文件，其后缀名是.py
- ```
!/usr/bin/python3
```

```
Filename: test.py
```

```
导入模块
```

```
import support
```

```
现在可以调用模块里包含的函数了
```

```
support.print_func("Runoob")
```
- 标准模块：math，random



# 模块和第三方类库

---

## 更强大的（第三方）类库

➤ 安装：pip3 install packagename (numpy , pandas)

➤ 导入：

1. import packagename

2. import packagename as new\_name

3. from packagename import function\_name

➤ 调用

1. packagename.function\_name

2. new\_name. function\_name

3. function\_name





大数据成就未来



# Thank you!

泰迪科技 : [www.tipdm.com](http://www.tipdm.com)  
热线电话 : 40068-40020

