人　工　智　能

python基础知识

目录

一 Python基本知识回顾

一、Python基本知识回顾

编写第一个Python程序

- hello_world = 'hello world!'
- print(hello_world)

- 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。

一、Python固定语法

加入代码注释

- # print(hello_world) 注释语句，不会被执行

- '''
- 多行注释，以下语句不会被执行
- print(hello_world)
- print(hello_world)
- print(hello_world)
- '''

一、Python固定语法

缩进代码

代码块以Tab或空格缩进限制，而不以花括号”{}”等分割代码，故编写代码时一定要注意代码对齐

- hello_world = 'hello world!'
- print(hello_world) # 代码未对齐，执行报错

- 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 -*-*

一、Python字符串与数值

任务描述

利用Python完成以下任务：

1. 创建一个字符串变量“Apple's unit price is 9 yuan.”。

2. 提取出里面的数字9并赋值给新的变量。

3. 查看新变量的数据类型。

4. 将提取的数字9转成整型（int）。

5. 确认数据类型是否转换成功。

一、Python字符串与数值

创建字符串

- 在Python中可以利用单引号、双引号、三引号创建字符串
- "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

- applePriceStr = "Apple's unit price is 9 yuan" # 创建字符串变量
- applePriceStr = 'Apple unite price is 9 yuan' # 单引号创建字符串
- applePriceStr = '''Apple's unit price is 9 yuan''' # 三引号创建字符串

一、Python字符串与数值

字符串基本用法

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

索引与切片：

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

字符串	P	Y	T	H	O	N
索引	0	1	2	3	4	5
负索引	-6	-5	-4	-3	-2	-1

一、Python字符串与数值

数据类型查看与转换

Python 3支持的数值型数据类型有int、float、bool、complex

数值型数据类型	中文解释	示　例
int	整数类型	10；100；1000
float	浮点数	1.0；0.11；1e-12
bool	布尔型	True；False
complex	复数	1+1j；0.123j；1+0j

1. 创建一个字符串变量“Apple's unit price is 9 yuan.”
2. 提取出里面的数字9并赋值给新的变量。
3. 查看新变量的数据类型。
4. 将提取的数字9转成整型（int）。
5. 确认数据类型是否转换成功。

一、Python字符串与数值

任务实现

```
# 1.创建一个字符串变量“Apple's unit price is 9 yuan.”。
applePriceStr = "Apple's unit price is 9 yuan"
# 2.提取出里面的数字9并赋值给新的变量。
applePrice = applePriceStr[-6]
# 3.查看新变量的数据类型。
print(type(applePrice))
# 4.将提取的数字9转成整型（int）。
applePriceInt = int(applePrice)
# 5.确认数据类型是否转换成功。
print(type(applePriceInt))
```

一、Python字符串与数值

计算圆形的各参数

根据相应数学公式，完成以下任务

1.给定圆的半径，计算圆的周长和面积

2.给定圆的周长，计算圆的半径和面积

3.给定圆的面积，计算圆的半径和周长

一、Python字符串与数值

常用操作运算符

1.常用算术运算符

运算符	描　述	示　例
+	加，即两个对象相加	10+20输出结果30
–	减，即得到负数或是一个数减去另一个数	20–10输出结果10
*	乘，即两个数相乘或是返回一个被重复若干次的字符串	10*20输出结果200
/	除，即x除以y	20/10输出结果2.0
%	取模，即返回除法的余数	23%10输出结果3
**	幂，即返回x的y次方	2**3输出结果为8
//	取整除，即返回商的整数部分	23//10输出结果2

一、Python字符串与数值

常用操作运算符

1.常用比较运算符

运算符	描　述	示　例
==	等于，即比较对象是否相等	(1==2)返回False
!=	不等于，即比较两个对象是否不相等	(1!=2)返回True
>	大于，即返回x是否大于y	(1>2)返回False
<	小于，即返回x是否小于y	(1<2)返回True
>=	大于，等于即返回x是否大于等于y	(1>=2)返回False
<=	小于，等于即返回x是否小于等于y	(1<=2)返回True

一、Python字符串与数值

常用操作运算符

1.常用赋值运算符

运算符	描　述	示　例
=	简单的赋值运算符	c=a+b将a+b的运算结果赋值为c
+=	加法赋值运算符	a+=b等效于a=a+b
-=	减法赋值运算符	a-=b等效于a=a-b
*=	乘法赋值运算符	a*=b等效于a=a*b
/=	除法赋值运算符	a/=b等效于a=a/b
%=	取模赋值运算符	a%=b等效于a=a%b
**=	幂赋值运算符	a**=b等效于a=a**b
//=	取整除赋值运算符	a//=b等效于a=a//b

一、Python字符串与数值

常用操作运算符

1.常用逻辑运算符

运算符	逻辑表达式	描　述	示　例
and	x and y	布尔“与”，即x and y，若x为False，则返回False；否则它返回y的计算值	a and b，返回22
or	x or y	布尔“或”，即x or y，若x是True，则返回True；否则它返回y的计算值	a or b，返回11
not	not x	布尔“非”，即not(x)，若x为True，则返回False。若x为 False，则返回True	not(a and b)，返回False

一、Python字符串与数值

常用操作运算符

1.成员运算符

运算符	描　述	示　例
in	如果在指定的序列中找到值，那么返回True，否则返回False	x in y，x在y序列中，返回True
not in	如果在指定的序列中没有找到值，那么返回True，否则返回False	x not in y，x不在y序列中，返回True

一、Python字符串与数值

常用操作运算符

1.身份运算符

运算符	描　述	示　例
is	用于判断两个标识符是不是引用自一个对象	x is y，如果id(x)等于id(y)，那么返回结果1
is not	用于判断两个标识符是不是引用自不同对象	x is not y，如果id(x)不等于id(y)，那么返回结果1

一、Python字符串与数值

常用操作运算符

1.运算符优先级排序，从上到下按由优先级从高到低进行排序

运算符	描　述
**	指数（最高优先级）
~　+　–	按位翻转、一元加号和减号
*　/　%　//	乘、除、取模和取整除
+　–	加法减法
>>　<<	右移、左移运算符
&	按位与运算符
^　|	按位或运算符
<=　<　>　>=	比较运算符
<>　==　!=	等于运算符
=　%=　/=　//=　–=　+=　*=　**=	赋值运算符
is is not	身份运算符
in not in	成员运算符
not or and	逻辑运算符

一、Python字符串与数值

任务实现

任务实现1：给定圆的半径，计算圆的周长和面积

$C = 2\pi r$，$S=\pi r^2$

- pi = 3.14 # 设置常量
- r = 3 # 输入圆形的半径
- C = 2 * pi * r # 计算圆形的周长
- S = pi * r ** 2 # 计算圆形的面积
- print('半径为', r, '的圆形，其周长等于', C, '；面积等于', S, '。')

一、Python字符串与数值

任务实现

任务实现1：给定圆的周长，计算圆的半径和面积

- C = 5 # 输入圆形的周长
- r = C / (2 * pi) # 计算圆形的半径
- S = pi * r ** 2 # 计算圆形的面积
- print('周长为' + str(C) + '的圆形，其半径为' + str(r) + '；面积等于' + str(S) + '。')

$$r = \frac{C}{2\pi} = \sqrt{\frac{S}{\pi}}$$

$$C = 2\pi r，\ S = \pi r^2$$

一、Python字符串与数值

任务实现

任务实现1：给定圆的面积，计算圆的半径和周长

$$r = \frac{C}{2\pi} = \sqrt{\frac{S}{\pi}}$$

$C = 2\pi r$，$S = \pi r^2$

- S = 5 # 输入圆形的面积
- r = round((S / pi) ** 0.5 , 2) # 计算圆形的半径，并保留两位小数
- C = round(2 * pi * r , 2) # 计算圆形的周长，并保留两位小数
- str_print = '面积为' + str(S) + '的圆形，其半径为' + str(r) + '；周长等于' + str(C) + '。'
- print(str_print)

二 Python函数

二、Python函数

使用def定义函数

- 使用def关键字定义一个求列表均值的自定义函数。
- 列表为：[1,2,6,0.3,2,0.5,-1,2.4]

get_features()

```
'''
整理训练集数据
'''
offline_train = pd.read_csv(data_train, parse_dates=['Date_received', 'Date'])   # 读取训练集样本

index1 = (offline_train['Date'] - offline_train['Date_received']).apply(lambda x: x.days <= 15)   # 时间间隔是否在15天
index2 = (offline_train['Date'] - offline_train['Date_received']).apply(lambda x: x.days > 15)    # 时间间隔是否大于15
index3 = offline_train['Coupon_id'].notnull() & offline_train['Date'].isnull()                    # 领了优惠券却未使用

offline_train['label'] = -1                          # 普通用户
offline_train.loc[index1, 'label'] = 1               # 正样本
offline_train.loc[(index2 | index3), 'label'] = 0    # 负样本
# offline_train['label'].value_counts()

offline_train['Discount_rate'] = offline_train['Discount_rate'].apply(get_discount)   # 特征1：优惠券的折扣率
offline_train['Distance'].fillna(method='ffill', inplace=True)                        # 特征2：商户与用户活跃地区的距离

Coupon_popu = offline_train[['Coupon_id', 'label']].groupby('Coupon_id').agg(lambda x: sum(x == 1)/sum(x != -1))  #
number_user_merchant_cust = offline_train[['User_id', 'Merchant_id', 'Date']].groupby(['User_id', 'Merchant_id']).a
Coupon_popu.columns = ['Coupon_popu']
number_user_merchant_cust.columns = ['number_user_merchant_cust']

offline_train3 = pd.merge(offline_train, Coupon_popu, left_on='Coupon_id', right_index=True, how='left')        #
offline_train4 = pd.merge(offline_train3, number_user_merchant_cust, left_on=['User_id', 'Merchant_id'], right_inde
offline_train4.fillna(method='ffill', inplace=True)   # 插补缺失值
offline_train4.fillna(method='bfill', inplace=True)   # 插补缺失值
# offline_train4.isnull().sum()
```

- 函数实现了对整段程序逻辑的封装
- 从程序代码中独立出来
- 避免出现大段重复代码
- 便于维护

```
def function(x,y):
    return 'result'
```

二、Python函数

使用lambda创建匿名函数

匿名函数，没有具体名称的函数

- lambda定义的是单行函数，如果需要复杂的函数，那么应使用def关键字。
- lambda函数可以包含多个参数，但
- lambda函数有且只有一个返回值。

- example = lambda x, y: x+y

二、Python函数

任务实现

- 使用def关键字定义一个求列表均值的自定义函数。
- 列表为：[1,2,6,0.3,2,0.5,-1,2.4]

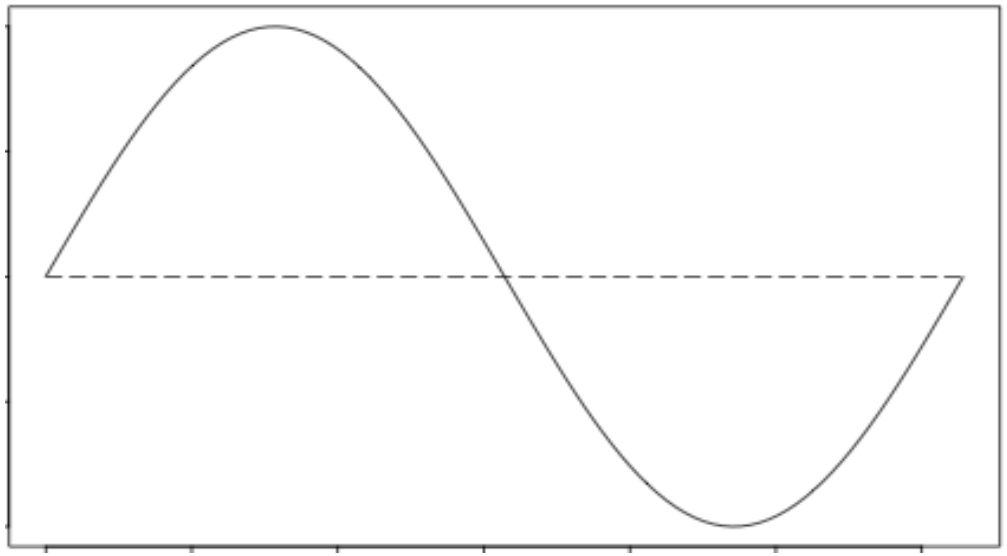
```
def mean(x):
    sum1 = 0
    for i in x:
        sum1 += i
    return sum1/len(x)
```

三 Python数据结构

三、Python数据结构

任务描述

求曲线y=sin(x)从0到2*pi，与x轴围成的面积。



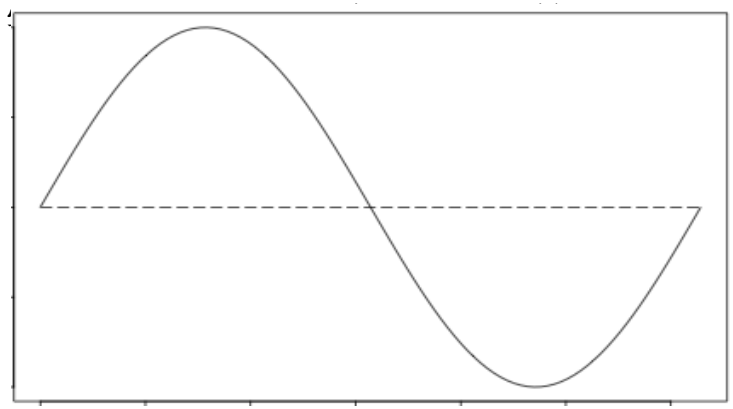
三、Python数据结构

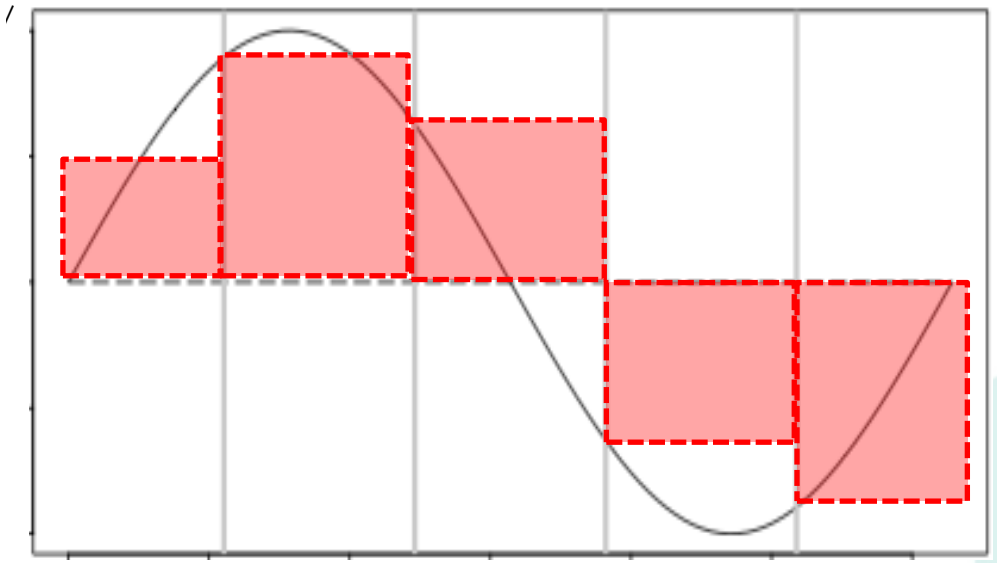
任务分析

1. 将图形等份划分，得到若干小矩形。

2. 求出各小矩形的高度。

3. 将各高度乘以宽度，得各小矩形面积，最后求和。





三、Python数据结构

数据结构

- 数据结构：根据某种方式将数据元素组合起来形成的一个数据元素集合。
- Python数据结构主要包含序列（如列表和元组）、映射（如字典）和集合3种基本的数据结构类型。

```
[0.0, 0.7, 1.4, 2.09, 2.79, 3.49, 4.19, 4.89, 5.59, 6.28]
```