

复习题

一、选择题

1、 $3*1**3$ ，该表达式输出的结果为（C）

A、27 B、9 C、3 D、1

解析：在Python中，指数运算符 `**` 具有比乘法运算符 `*` 更高的优先级。因此，先计算 $1**3$ ，即将 1 进行立方运算，结果为 1。然后，将 3 乘以 1 的立方结果，即 $3*1$ 。最终的计算结果就是 3。

2、以下代码输出的结果是（C）

```
if None:
    print('hello')
```

A、False B、Hello C、没有任何输出 D、语法错误

解析：在 Python 中，`None` 是一个特殊的对象，代表空值或者没有值。在条件判断语句中，`None` 被视为假（False），因此 `if None` 的条件判断结果为 False。

3、以下代码输出的结果是（A）

```
for i in [1, 0]:
    print(i + 1)
```

A、2

1

B、[2, 1]

C、2

D、0

4、以下哪个描述是正确的（D）

A、break语句用于终止当前循环

B、continue语句用于跳过当前剩余要执行的代码，执行下一次循环

C、break和continue语句通常与if，if...else，if...elif...else语句一起使用

D、以上说法都是正确的。

5、以下代码输出的结果是（A）

```
for char in 'PYTHON STRING':
    if char == ' ':
        break

    print(char, end='')

if char == 'O':#PYTHON
```

`continue`

解析：如果是先输出`print` 那么遍历到`o`时已经在`o`之前进行打印了，遍历到`o`跳出循环进行下一次循环 直到遍历到空格时执行`break`终止循环

观察区别之处：

```
for char in 'PYTHON STRING':
    if char == ' ':
        break

    if char == 'O':
        continue
    print(char, end=' ')#PYTHN
```

A、PYTHON B、PYTHONSTRING C、PYTHN D、STRING

6、下列语句不正确的是（ B ）

A、`x = y = z = 1`

B、`x = (y = z + 1)`

C、`x , y = y , x`

D、`x += y`

解析：

A、`x = y = z = 1`： 这个表达式意味着将 1 赋值给变量 `z`，然后将 `z` 的值赋给 `y`，最后将 `y` 的值赋给 `x`。结果是 `x`、`y` 和 `z` 都被赋值为 1。

B、`x = (y = z + 1)`： 这是一个不合法的表达式。在 `python` 中，赋值语句并不能作为其他赋值语句的一部分进行嵌套使用。因此，这个表达式会引起语法错误。

C、`x , y = y , x`： 这个表达式使用了元组解包的方式，将变量 `x` 和 `y` 进行交换。首先，将 `y` 的值赋给 `x`，然后将 `x` 的值赋给 `y`。结果是变量 `x` 和 `y` 交换了它们的值。

D、`x += y`： 这个表达式是增量赋值运算符，等价于 `x = x + y`。它表示将变量 `x` 的值与变量 `y` 的值相加，然后将结果再赋值给变量 `x`。

二、填空题

1、以下代码输出的结果为—— 3

```

x = True
y = False
z = False

if not x or y: #if F or F: ----if F 不会进入
    print(1)
elif not x or not y and z:#elif F or (T and F) -- F or F ---F
    print(2)
elif not x or y or not y and x:#elif F or F or( T and T)---T 条件为真进入elif内部
    print(3)
else:
    print(4)

```

注：注意优先级 `not>and>or`

2、以下代码输出的结果为—— 10

```

i = sum = 0

while i <= 4:
    sum += i
    i = i + 1
print(sum)

```

使用debug查看内部执行过程

3、当函数中参数个数不确定是怎样定义形参—— *args , **kwargs

4、bool类型当作数值时，分别为哪两个数值—— 0, 1

5、0 and 1，输出的结果为—— 0

6、0 or 1，输出的结果为—— 1

7、函数的参数传递形式分别为：—— 位置参数，默认参数，关键字参数，可变数量参数（分为不定长位置、不定长关键词）

8、如何定义一个空集合A，—— A = set()

三、判断题

1、以下代码只会执行一次。（ F ）

```

while 4 == 4:
    print('hello')

```

解析：

这段代码会无限循环输出 'hello'。

条件 `4 == 4` 恒为真，因此 `while` 循环条件一直满足，循环会一直执行。每次循环迭代时，都会执行 `print('hello')`，输出 'hello'。由于条件永远为真，所以循环会一直进行下去，直到被手动中断或程序终止。

2、字典中值 (value) 不可以重复。（ F ）

解析---例如:

```
date = {'name':[1,2], 'age':[1,2], '数据':True}
print(date['name'])#[1, 2]
print(date['age'])#[1, 2]
print(date['数据'])#True
```

3、字典中的键 (key) 可以是列表。(F)

注: 字典中的键必须为不可变数据

```
date = {'name':[1,2], 'age':[1,2], [1,2]:True}
print(date[[1,2]])#报错
```

4、a = {2, 5, 3, [2, 4, 6, 8], 10}这个集合是正确的。(F)

解析: 集合的元素必须为不可变

5、以下代码输出的x为196。(F)

```
def func():
    global x
    x = 56
```

```
x = 196
func()
print(x)
```

解析:

在函数 `func` 中使用了 `global` 关键字, 表示声明变量 `x` 为全局变量。然后在函数体内将全局变量 `x` 的值修改为 `56`。

在调用函数 `func` 之前, 变量 `x` 被赋值为 `196`。但由于在函数内部修改了全局变量 `x` 的值为 `56`, 所以在函数调用后, `print(x)` 输出的结果是函数内修改后的值, 即 `56`

四、问答题

1、写出Python中的数据类型 (在每个数据类型后面用括号备注上可变或不可变):

数值类型: 整型(`int` , 不可变) , 浮点型(`float` , 不可变) , 布尔类型(`bool` , 不可变)

序列类型: 字符串(`str` , 不可变) , 列表(`list` , 可变) , 元组(`tuple` , 不可变)

散列类型: 字典(`dict` , 可变) , 集合(`set` , 可变)

2、序列类型跟散列类型的区别

序列类型: 有序的、可迭代的数据类型, 它可以按照顺序存储多个元素, 并且支持索引和切片操作

散列类型主要包括字典, 适用于键值对的映射关系, 每个键必须唯一且不可变。

也包括集合, 集合是一种无序、不重复的散列类型, 它只能包含不可变 (可散列) 的元素, 常用于存储一组唯一的元素。集合使用花括号 `{}` 或者 `set()` 函数来创建。

3、讲过的几个常用模块

```
os , re, math
```

4、什么是模块

模块（Module）是一个包含 Python 代码的文件。它可以包含变量、函数、类和其他可执行的代码。通过将相关的代码组织在模块中，可以提高代码的可重用性和维护性。在程序中，可以使用 `import` 语句来导入模块，并使用模块名来访问其中的代码。

四、代码题

1. 在终端中录入一个疫情确诊人数,再录入一个治愈人数,打印治愈比例。格式: 治愈比例为xx%

```
sick = int(input('疫情确诊人数: '))
heal = int(input('疫情治愈人数: '))
a = int(heal / sick*100)
print('治愈比例: {:.2f}%'.format(a))
```

结果:

疫情确诊人数: 1000

疫情治愈人数: 12

治愈比例: 1.00%

2. 在终端中输入账户类型,消费金额,计算折扣。

如果是vip客户,消费小于等于500,享受85折。消费大于500, 享受8折

如果不是vip客户,消费大于等于800,享受9折。消费小于800, 原价

```
is_vip = input('请问是vip客户吗')
money = float(input('请问消费了多少钱? '))
if is_vip == '是':
    if money <= 500:
        a = money*0.85
        print(f'享受八五折消费{a}')
    else:
        a = money * 0.8
        print(f'享受八折消费{a}')
else:
    if money >= 800:
        a = money*0.9
        print(f'享受九折消费{a}')
    else:
        a = money
        print(f'原价消费{a}')
```

结果:

请问是vip客户吗不是

请问消费了多少钱? 1000

享受九折消费900.0

3. 一家公司有如下岗位:

经理 : "曹操", "刘备", "孙权"

技术 : "曹操", "刘备", "张飞", "关羽"

- (1). 使用集合存储以上信息.
- (2). 是经理也是技术的都有谁?
- (3). 是经理但不是技术的都有谁?
- (4). 不是经理是技术的都有谁?
- (5). 公司总共有哪几个人?

#1

```
a = {'曹操', '刘备', '孙权'}#经理
b = {'曹操', '刘备', '张飞', '关羽'}#技术
```

#2.求并集

```
print(a&b)
```

3.求a的补集 , a有但是b没有的

```
print(a-b)
```

4.求b的补集

```
print(b-a)
```

5.求并集

```
c = a|b
```

```
print(c)
```

4.定义一个函数 , 传入一个有重复元素列表 `a=[1,2,3,4,4,5,6,2,1]` , 让函数返回的是一个无重复元素的列表

```
a = [1,2,3,4,5,6,2,5,6,1]
```

```
def fun():
```

```
    return list(set(a))
```

```
print(fun())
```

5.求1~100间所有奇数的和

```
sum1 = 0
```

```
for i in range(1,101,2):
```

```
    sum1 += i
```

```
print(sum1)
```

6.用函数写一个计算器 (有加减乘除的功能供用户选择计算)

```
def add(num1, num2):
```

```
    return num1 + num2
```

```
def sub(num1, num2):
```

```
    return num1 - num2
```

```
def mlu(num1, num2):
```

```
    return num1 * num2
```

```
def div(num1, num2):
```

```
    return num1 / num2
```

```
print('欢迎来到某某计算器')
```

```
a = float(input('请输入你要计算的第一个值>>>'))
```

```
b = float(input('请输入你要计算的第二个值>>>'))
```

```
select = int(input('请选择你要进行的操作1加法,2减法,3乘法,4.除法'))
```

```
if select == 1:
```

```

print(add(a,b))

elif select == 2:
    print(sub(a,b))

elif select == 3:
    print(mlu(a,b))

elif select == 4:
    print(div(a,b))

```

7. 一个函数只有一个形参，当接收多个元素时要求返回的是一个元组类型。这个函数内部对接收到的数字进行过滤，要求所得元组的第一个元素是这些元素的平均值，后面的元素是大于这个平均值的数。

第一步：先求平均值， $avg = \text{和} / \text{数据长度}$

```

def fun_1(*args):
    sum1 = 0
    for i in args:
        sum1 += i
    avg = sum1 / len(args) #求平均值
    li = [] #定义一个空列表
    for i in args:
        if i > avg:
            li.append(i) #将大于平均值的元素添加到这个空列表中
    li.insert(0, avg) #在列表中的索引下标为0的位置插入平均值
    return tuple(li) #将列表转为元组

print(fun_1(1, 2, 3, 4, 5, 6, 7, 15, 20))

```

面向对象

第一题：

```

class People:
    def __init__(self,name,weight):
        self.name = name
        self.weight = weight

    def to_run(self):
        print('正在跑步')
        self.weight -= 0.5

    def to_eat(self):
        print('正在吃东西')
        self.weight += 1

xiaoming = People(name="xiaoming",weight=75)
xiaomei = People(name="xiaomei",weight=45)
xiaomei.to_run()
print(xiaomei.weight)
xiaoming.to_eat()
print(xiaoming.weight)

```

第二题：

```

class Instrument:
    def makeSound(self):

```

```
print("乐器发出美妙的声音")

class Erhu(Instrument):
    def makeSound(self):
        print("二胡拉响人生")

class Piano(Instrument):
    def makeSound(self):
        print("钢琴美妙无比")

class Violin(Instrument):
    def makeSound(self):
        print("小提琴来啦")

def music(obj):
    obj.makeSound()

eh = Erhu()
po = Piano()
vl = Violin()

music(vl)
music(po)
music(eh)

第三题:
class WuMingFen:
    def __init__(self, m: str, q: int, s:bool):
        self.theMa = m
        self.quantity = q
        self.isSoup = s

    def check(self):
        c= '带汤' if self.isSoup == True else '不带汤'
        print(f"粉的类型:{self.theMa}, 粉的分量:{self.quantity}两,是否带汤:{c}")

wumingfen = WuMingFen("酸辣粉",2,True)
wumingfen.check()
```