

PPID31 2021 年 Python 程序设计模拟试题（new） 试卷预览

授权码：4b884b

单选题

1. (2 分)

下面（ ）不是有效的变量名 *数字 字母 下划线不能与 关键字相同。*

1. _demo
2. banana
3. Numbr
4. my-score ✓

2. (2 分) *运算符*

*算数 ** : 幂运算 //: 取整*

面向对象 3 大特性 P132

关系 逐个字符比较

逻辑 and or not 惰性求值

in not in is not is

关于 a and b 的描述错误的是（ ）。

1. 若 a=True b=True 则 a and b ==True
2. 若 a=True b=False 则 a and b ==True ✓
3. 若 a=False b=True 则 a and b ==False
4. 若 a=False b=False 则 a and b ==False

3. (2 分) *字符串 课本：21~24*

字符串操作函数：replace strip join split

内置函数对于字符串：P26 len()

*操作符： + **

其他内置函数： float() int() str() chr() ord() eval()

若 a = 'abca'，若想将 a 变为 'ebca'，则下列语句正确的是()

1. a[0] = 'e'
2. a.replace('a', 'e')
3. a[1] = 'e'
4. a = 'e' + a[1:] ✓

4. (2 分) 组合数据类型 列表 字典 集合 元组
创建

有序与无序 有：列表 元组 字符串

可变与不可变 可变：列表，字典，集合

数据类型对应的操作函数： P75

数据类型的操作运算： P65 P72 P79 P66 P85 P90 P92

执行下面操作后，list2 的值是：

```
list1 = ['a', 'b', 'c']
```

```
list2 = list1
```

```
list1.append('de')
```

1. ['a', 'b', 'c']
2. ['a', 'b', 'c', 'de'] ✓
3. ['d', 'e', 'a', 'b', 'c']
4. ['a', 'b', 'c', 'd', 'e']

5. (2 分)

获得字符串 s 长度的方法是().

1. s.len()
2. s.length
3. len(s) ✓
4. length(s)

6. (2 分)

下列 for 语句中，在 in 后使用不正确的是：

```
for var in ( ):
```

```
    print (var)
```

1. set('str')
2. (1) ✓
3. [1, 2, 3, 4, 5]
4. range(0, 10, 5)

7. (2 分)

()不能创建一个字典的语句.

1. dict = {}
2. dict = {(4,5,6):'dictionary'}
3. dict= {4: 6}
4. dict = {[4,5,6]:'dictionary'} ✓

8. (2 分)

for 或者 while 与 else 搭配使用时，关于执行 else 语句块描述正确的是 ()

1. 永不执行
2. 仅循环非正常结束后执行（以 break 结束）
3. 仅循环正常结束后执行 ✓
4. 总会执行

9. (2 分)

下面代码的输出结果是 ()

```
s =["seashell","gold","pink","brown","purple","tomato"]  
  
print(s[4:])
```

1. ['purple']
2. ['seashell', 'gold', 'pink', 'brown']
3. ['purple', 'tomato'] ✓
4. ['gold', 'pink', 'brown', 'purple', 'tomato']

10. (2 分) 函数定义 返回值

关于 Python 的 lambda 函数，()描述是错误的。

1. f = lambda x,y:x+y 执行后，f 的类型为数字类型 ✓
2. 可以使用 lambda 函数定义列表的排序原则
3. lambda 函数将函数名作为函数结果返回
4. lambda 用于定义简单的、能够在一行内表示的函数

11. (2 分)

```
a=int(input()) #输入 3
```

```
a**=2
```

```
print(A+2)
```

代码运行结果为:().

1. 报错 NameError: name 'A' is not defined ✓
2. 11
3. 8
4. 5

12. (2 分)

以下代码输出结果为:

```
s = "6.00 is 6.0001 and 6.0002"
```

```
new_str = ""
```

```
new_str += s[-1]
```

```
new_str += s[0]

new_str += s[4::30]

new_str += s[13:10:-1]

print(new_str)
```

1. 260000
2. 26100
3. 26 100 ✓
4. 6.00 is 6.0001 and 6.0002

13. (2 分)

已知

```
aList = [3, 4, 5, 6, 7, 9, 11, 13, 15, 17]
```

则

aList[1::2]的结果为:

1. [4, 6, 9, 13, 17] ✓
2. [3, 4, 5, 6, 7]
3. [3, 4]
4. [3, 6, 11, 17]

14. (2 分)

下面代码的执行结果是

```
>>>x = 2
```

```
>>>x *= 3 + 5**2
```

1. 8192
2. 13

- 3. 15
- 4. 56 ✓

15. (2 分)

`str="abcdefg"`

则 `str[2:4]` 的内容是 ()

- 1. cd ✓
- 2. bc
- 3. bcd
- 4. cde

16. (2 分)

下列代码运行后，会输出"common letter"()次。

```
s1 = "mit u rock"
```

```
s2 = "i rule mit"
```

```
if len(s1) == len(s2):
```

```
    for char1 in s1:
```

```
        for char2 in s2:
```

```
            if char1 == char2:
```

```
                print("common letter")
```

```
                break
```

- 1. 0
- 2. 7 ✓
- 3. 8
- 4. 10

17. (2 分)

给定字典 d, ()对 d.items() 的描述是正确的.

1. 返回一种 dict_items 类型, 包括字典 d 中所有键值对 ✓
2. 返回一个列表类型, 每个元素是一个二元元组, 包括字典 d 中所有键值对
3. 返回一个元组类型, 每个元素是一个二元元组, 包括字典 d 中所有键值对
4. 返回一个集合类型, 每个元素是一个二元元组, 包括字典 d 中所有键值对

18. (2 分)

关于字符串 ()说法是错误的.

1. 字符应该视为长度为 1 的字符串
2. 字符串以 \0 标志字符串的结束 ✓
3. 既可以用单引号, 也可以用双引号创建字符串
4. 在三引号字符串中可以包含换行、回车等特殊字符

19. (2 分)

下面代码的输出结果是:

```
print( 0.1 + 0.2 == 0.3)
```

1. False ✓
2. -1
3. 0
4. True

20. (2 分)

Python 中对变量描述错误的选项是 ().

1. Python 不需要显式声明变量类型, 在第一次变量赋值时由值决定变量的类型
2. 变量通过变量名访问
3. 变量必须在创建和赋值后使用
4. 变量 PI 与变量 Pi 被看作相同的变量 ✓

程序填空题

1. (10 分) P52

程序的功能： 输入 1~9 间的整数作为行数 n，打印如图所示的 n 行回文数字字符图形。运行界面如下：

```
>>>
输入行数 (1-9) :5
5行
    1
   121
  12321
 1234321
123454321
>>> |
```

代码如下：

```
s = input('输入行数 (1-9) :')
```

#(1) 此处打印行数

字符串的格式化输出： P67

(1)

```
line = int(s)
```

```
for i in range(1, line+1):
```

```
    space = ' ' * (line-i)          #此句单引号中有一个空格
```

```
    print(space, end = '')
```

#(2) 此处用循环打印前半数字串

(2)

```
print(str(j), end = '')
```

```
for j in range(i-1, 0, -1):
```

```
#(3) 此处在此循环体中打印后一半数字串
```

```
_____  
(3)
```

```
print()
```

答案:

```
print(s+' 行');print("{} 行" .format(s));print("%s 行"%s)
```

```
for j in range(1, i+1): ;for j in range(1, i+1, 1):
```

```
print(str(j), end='')
```

2. (10 分)GUI

功能说明：设计一个加法计算器程序。将操作数填入输入框后，单击“加法”按钮将算式和结果填入下方的结果文本框中。单击“清空”则将清空输入框和结果文本框。请在空白处填入相应的代码，使程序功能完整。



代码如下：

```
from tkinter import *
```

```
def run():
```

```

a= t1.get()

b=_____ (1) _____(t2.get())

c=float(a)+b

d=' %s+%s=%s\n' % (a, b, c)

txt.insert(END, d)


def clear():

    t1.delete(0, END)

    t2.delete(0, END)

    txt.delete(0.0, END)


root=_____ (2) _____

root.title(' 加法')

root.geometry(' 320x240')

t1=Entry(root)

t1.place(relx=0.1, rely=0.05, relwidth=0.2, relheight=0.1)

t2=Entry(root)

t2.place(relx=0.5, rely=0.05, relwidth=0.2, relheight=0.1)

cmd1=Button(root, text=' 加法
', _____ (3) _____)####

cmd1.place(relx=0.1, rely=0.3, relwidth=0.2, relheight=0.1)

cmd2=Button(root, text=' 清空', command=clear)

cmd2.place(relx=0.5, rely=0.3, relwidth=0.2, relheight=0.1)

txt=Text(root)

```

```
txt.place(relx=0.1, rely=0.6, relwidth=0.6, relheight=0.3)
```

(4)

答案:

```
float
```

```
Tk()
```

```
command=run
```

```
root.mainloop()
```

3. (10 分)P

输入一个字符串作为密码，密码只能由数字与字母组成。编写程序判断输入的密码的强度，并输出。

判断标准为：满足其中一条，密码强度增加一级。

1) 有数字

2) 有大写字母

3) 有小写字母

4) 位数不少于 8 位

填空完成以下程序：

```
def judge(password):
```

```
    result=0
```

```
    n=_____ (1)
```

```
    if n>=8:
```

```
        result+=1
```

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

```
        if '0' <=password[i]<='9':
```

```

        result+=1

        break

for i in range(n):

    if 'A' <=password[i]<='Z':

        result+=1

        break

for i in range(n):

    if 'a' <=password[i]<='z':

        result+=1

        break

    (2) result

```

```

def main():

    """

    主函数

    """

    while True:

        password = input(' 请输入密码(直接回车为退出): ')

        if password=='':

            (3)

            s=judge( (4) )

            print("%s 的密码强度为%d 级"%(password, s))

```

```
if __name__ == '__main__':  
    main()
```

答案:

```
len(password)
```

```
return
```

```
break
```

```
password
```

设计操作题 P105 5-7 5-8 5-9

1. (10 分)

输入学生信息，包含如下信息：

ID	名字	英语成绩	Python成绩	C语言成绩
1001	张三	76	87	96
1002	李红	88	78	69

请编程计算每个学生的总分，并按总分从低到高排序。最后输出排序后的信息，如下。代码文件保存在 grade.py

ID	名字	英语成绩	Python成绩	C语言成绩	总成绩
1003	90	-1	-1	0	-2
1002	李红	88	78	69	235
1001	张三	76	87	96	259

2. (10 分)

回文数的判定。设 n 是一任意自然数，如果 n 的各位数字反向排列所得自然数与 n 相等，则 n 称为回文数。从键盘输入一个 5 位数，请编写程序判定这个数是否为回文数。代码保存在 huiwen.py

```
a=input( ' ' )  
  
b=a[::-1]  
  
print( ' ' ) if a==b else print( ' ' )
```

3. (10 分)

从文件 data.txt 中读入信息，将其中包含的“密”字（可能出现 0 次、1 次或者多次）删除，最后原样输出剩余的文本内容，同时统计出剩余文本中每个汉字出现的次数，并输出。代码保存到 opfile.py.