1. 判断题
2. 表达式 3 in {1, 2, 3} 的值为True。 （ √ ）
3. 表达式 'ac' in 'abce' 的值为False。 （ √ ）
4. 函数中必须包含return语句。（ × ）
5. 函数中的return语句一定能够得到执行。（ × ）
6. 在同一个作用域内，局部变量会隐藏同名的全局变量。（ √ ）
7. 在try...except...else结构中，如果try块的语句引发了异常则会执行else块中的代码。（ √ ）
8. 异常处理结构中的finally块中代码仍然有可能出错从而再次引发异常。（ √ ）
9. 在Python中定义函数时不需要声明函数参数的类型。（ √ ）
10. 在Python中定义函数时不需要声明函数的返回值类型。（ √ ）
11. 当作为条件表达式时，[]与None等价。（ √ ）
12. 表达式 [] == None 的值为True。（ × ）
13. 当作为条件表达式时，{}与None等价。（ √ ）
14. 表达式 {}==None 的值为True。（ × ）
15. Python类的构造函数是\_\_init\_\_()。（ √ ）
16. 表达式 ‘a’+1的值为’b’。（ × ）
17. 函数是代码复用的一种方式。（ √ ）
18. 定义Python函数时，如果函数中没有return语句，则默认返回空值None。（ √ ）
19. 表达式 {1, 3, 2} > {1, 2, 3} 的值为True。（ × ）
20. 假设x是含有5个元素的列表，那么切片操作x[10:]是无法执行的，会抛出异常。（ × ）
21. 只能通过切片访问元组中的元素，不能使用切片修改元组中的元素。（ √ ）
22. 问答题
23. Python语句list(range(1,10,3))执行结果为\_\_\_\_\_[1, 4, 7]\_\_\_\_\_\_ \_。
24. 表达式 key=lambda x: len(str(x))，则key(111) 的值为\_\_\_[2, 33, 111]\_\_\_\_。
25. 切片操作list(range(6))[::2]执行结果为\_\_\_\_[0, 2, 4]\_\_\_。
26. 字典对象的\_\_\_ items()\_\_\_方法返回字典中的“键-值对”列表。
27. 字典对象的\_\_\_\_ keys()\_\_\_方法返回字典的“键”列表。
28. 字典对象的\_\_\_ values()\_\_\_\_方法返回字典的“值”列表。
29. 已知 x = {'a':'b', 'c':'d'}，那么表达式 'a' in x 的值为\_\_\_\_\_ True \_\_\_\_\_。
30. 已知 x = {'a':'b', 'c':'d'}，那么表达式 'b' in x 的值为\_\_\_\_\_\_ False \_\_\_\_。
31. 已知 x = {'a':'b', 'c':'d'}，那么表达式 'b' in x.values() 的值为\_\_\_\_True\_\_\_\_。
32. 表达式 1<2<3 的值为\_\_\_\_True\_\_\_\_\_。
33. 表达式 3 or 5 的值为\_\_\_\_3\_\_\_\_。
34. 已知 x = [1, 2, 3, 2, 3]，执行语句 x.remove(2) 之后，x的值为\_\_[1, 3, 2, 3]\_\_。
35. Python中定义函数的关键字是\_\_\_\_\_\_ def \_\_\_\_\_\_\_。
36. 已知 x = [1, 2, 3]，那么执行语句 x[len(x)-1:] = [4, 5, 6]之后，变量x的值为

\_[1, 2, 4, 5, 6]\_。

1. 已知 x = {1:2, 2:3}，那么表达式 x.get(3, 4) 的值为\_\_4\_\_。
2. 已知 x = {1:2, 2:3}，那么表达式 x.get(2, 4) 的值为\_\_\_3\_\_。
3. 表达式 {1, 2, 3} | {3, 4, 5} 的值为\_\_\_\_\_{1, 2, 3, 4, 5}\_\_\_\_\_\_。
4. 表达式 {1, 2, 3} | {2, 3, 4} 的值为\_\_\_\_\_{1, 2, 3, 4}\_\_\_\_\_\_\_\_。
5. 已知 f = lambda x: 5，那么表达式 f(3)的值为\_\_\_\_\_5\_\_\_\_\_\_。
6. 表达式 not 3 的值为\_\_\_\_\_False\_\_\_\_\_\_。
7. 编程题
8. 请分别用for和while循环分别实现1到100的求和，并打印出计算结果。

# for循环  
**def sum**():  
 sum = 0  
 **for** n **in** range(1, 101):  
 sum = sum + n  
 **return** sum  
print(sum())  
  
# while循环  
**def sum**():  
 sum = 0  
 x=1  
 **while** x < 101:  
 sum = sum + x  
 x+=1  
 **return** sum  
print(sum())

1. 请应用百度云提供的语音相关API完成以下内容：  
   1. 自行注册相关账号，并建立应用；

2. 阅读相关语音解析文字和文字合成语音的相关API文档；

3. 解析“16k-23850.amr”语音文件内容，并将返回值“{'corpus\_no': '6568285558591004267', 'err\_msg': 'success.', 'sn': '182861598861529298154', 'result': ['北京科技馆，'], 'err\_no': 0}”，提取出对应的中文，即“北京科技馆”，注你可以应用以前学习的所有知识，关于字符处理相关函数请自行查阅相关资料；

4. 请应用语音合成函数，合成包括你们小组所有人员名字和你们小组的口号，要求用男播音员的口音。

**from** aip **import** AipSpeech  
  
# 定义常量，此处替换为你自己的应用信息  
APP\_ID = '11322263'  
API\_KEY = 'gfExV6h1dp3YS5jlwS8P5GmQ'  
SECRET\_KEY = 'GwgtAyGWKilXOYrTTI5mmlyelGK81XUj'  
  
# 初始化AipSpeech对象  
aipSpeech = AipSpeech(APP\_ID, API\_KEY, SECRET\_KEY)  
  
# # # 读取文件  
**def get\_file\_content**(filePath):  
 **with** open(filePath, 'rb') **as** fp:  
 **return** fp.read()  
# #  
# # 识别本地文件  
# #目前支持的格式较少，原始 PCM 的录音参数必须符合 8k/16k 采样率、16bit 位深、单声道，支持的格式有：pcm（不压缩）、wav（不压缩，pcm编码）、amr（压缩格式）。  
result = aipSpeech.asr(get\_file\_content(r'16k-23850.amr'), 'amr', 16000,{'lan': 'zh',})  
print(result)  
list1=list(result.get('result'))  
s=list1[0]  
print(s)  
cc=s.replace(',','')  
print(cc)  
  
result = aipSpeech.synthesis('第一必然是我们的！', 'zh', 1, {'vol': 5,'per':1})  
print(result)  
# # # 识别正确返回语音二进制 错误则返回dict 参照下面错误码  
**if not** isinstance(result, dict):  
 **with** open('口号.mp3', 'wb') **as** f:  
 f.write(result)