

# python题库

---

## 1000题库系列二

2.1 表达式`int('1111', 2)`的值为\_\_\_\_\_。

2.2 表达式`chr(ord('D')+2)`的值为\_\_\_\_\_。

2.3 简单解释Python基于值的内存管理模式。

2.3 Python采用的是基于值得内存管理方式，在Python中可以为不同变量赋值为相同值，这个值在内存中只有一份，多个变量指向同一个内存地址；Python具有自动内存管理功能，会自动跟踪内存中所有的值，对于没有任何变量指向的值，Python自动将其删除。

2.4 简单解释运算符/和//的区别。

2.5 运算符%\_\_\_\_\_（可以/不可以）对浮点数进行求余数操作。

2.6 一个数字5\_\_\_\_\_（是/不是）合法的Python表达式。

2.7 判断对错：在Python 3.x中，内置函数`input()`把用户的键盘输入一律作为字符串返回。

2.8 在Python中关键字\_\_\_\_\_表示空类型。

2.9 列表、元组、字符串是Python的\_\_\_\_\_（有序、无序？）序列。

2.10 查看变量类型的Python内置函数是\_\_\_\_\_。

2.11 查看对象内存地址的Python内置函数是\_\_\_\_\_。

2.12 以3为实部4为虚部，Python复数的表达形式为\_\_\_\_\_或\_\_\_\_\_。

2.13 Python运算符中用来计算整商的是\_\_\_\_\_。

2.14 Python运算符中用来计算集合并集的是\_\_\_\_\_。

2.15 Python运算符中用来计算集合差集的是\_\_\_\_\_。

2.16 Python运算符中用来计算集合交集的是\_\_\_\_\_。

2.17 使用运算符测试集合A是否为集合B的真子集的表达式可以写作\_\_\_\_\_。

2.18 \_\_\_\_\_命令既可以删除列表中的一个元素，也可以删除整个列表或其他任意类型的Python对象。

2.19 表达式`int('123', 16)`的值为\_\_\_\_\_。

2.20 表达式`int('123', 8)`的值为\_\_\_\_\_。

2.21 表达式`int('123')`的值为\_\_\_\_\_。

2.22 表达式`int('101', 2)`的值为\_\_\_\_\_。

2.23 表达式`abs(-3)`的值为\_\_\_\_\_。

2.24 Python 3.x语句`print(1, 2, 3, sep=',')`的输出结果为\_\_\_\_\_。

2.25 Python 3.x语句`print(1, 2, 3, sep=',')`的输出结果为\_\_\_\_\_。

2.26 表达式`int(4**0.5)`的值为\_\_\_\_\_。

**2.27 表达式`8 ** (1/3)`的值为\_\_\_\_\_。**

2.28 Python内置函数\_\_\_\_\_可以返回列表、元组、字典、集合、字符串以及`range`对象中元素个数。

2.29 Python内置函数\_\_\_\_\_用来返回序列中的最大元素。

2.30 Python内置函数\_\_\_\_\_用来返回序列中的最小元素。

2.31 Python内置函数\_\_\_\_\_用来返回数值型序列中所有元素之和。

2.32 已知`x = 3`，那么执行语句`x += 6`之后，`x`的值为\_\_\_\_\_。

2.33 表达式`13 / 4`的值为\_\_\_\_\_。

2.34 表达式`13 // 4`的值为\_\_\_\_\_。

**2.35 表达式`-13 // 4`的值为\_\_\_\_\_。**

2.36 表达式`3 ** 2`的值为\_\_\_\_\_。

2.37 表达式`chr(ord('a')-32)`的值为\_\_\_\_\_。

**2.38 表达式`abs(3+4j)`的值为\_\_\_\_\_。**

2.39 表达式`16 ** 0.5`的值为\_\_\_\_\_。

2.40 表达式`type({3})`的值为\_\_\_\_\_。

2.41 表达式`type({3:3})`的值为\_\_\_\_\_。

2.42 表达式`isinstance('Hello world', str)`的值为\_\_\_\_\_。

2.43 表达式`isinstance('abc', int)`的值为\_\_\_\_\_。

2.44 表达式`isinstance(4j, (int, float, complex))`的值为\_\_\_\_\_。

2.45 表达式`isinstance('4', (int, float, complex))`的值为\_\_\_\_\_。

2.46 表达式`type(3) in (int, float, complex)`的值为\_\_\_\_\_。

2.47 表达式`type('3') in (int, float, complex)`的值为\_\_\_\_\_。

2.48 表达式`type(3) == int`的值为\_\_\_\_\_。

2.49 判断对错：已知`x = 3`，那么赋值语句`x = 'abcdefg'`是无法正常执行的。

2.50 判断对错：Python变量使用前必须先声明，并且一旦声明就不能在当前作用域内改变其类型了。

**2.51 判断对错：Python不允许使用关键字作为变量名，但是允许使用内置函数名作为变量名，不过这会改变函数名的含义，所以不建议这样做。**

- 2.52 判断对错：在Python中可以使用if作为变量名。
- 2.53 判断对错：在Python 3.x中可以使用中文作为变量名。
- 2.54 判断对错：Python变量名必须以字母或下划线开头，并且区分字母大小写。**
- 2.55 判断对错：加法运算符可以用来连接字符串并生成新字符串。
- 2.56 判断对错：`x = 9999**9999`这样的语句在Python中无法运行，因为数字太大了超出了整型变量的表示范围。
- 2.57 判断对错：`3+4j`不是合法的Python表达式。
- 2.58 判断对错：0o12f是合法的八进制数字。**
- 2.59 判断对错：只有Python扩展库才需要导入以后才能使用其中的对象，Python标准库不需要导入即可使用其中的所有对象。**
- 2.60 判断对错：在Python中0xad是合法的十六进制数字表示形式。**
- 2.61 判断对错：`4j`是合法Python数字类型。
- 2.62 判断对错：Python使用缩进来体现代码之间的逻辑关系，对缩进的要求非常严格。
- 2.63 判断对错：Python代码的注释只有一种方式，那就是使用#符号。
- 2.64 判断对错：放在一对三引号之间的任何内容将被认为是注释。**
- 2.65 判断对错：尽管可以使用import语句一次导入任意多个标准库或扩展库，但是仍建议每次只导入一个标准库或扩展库。
- 2.66 判断对错：为了让代码更加紧凑，编写Python程序时应尽量避免加入空格和空行。
- 2.67 判断对错：在Python 3.5中运算符+不仅可以实现数值的相加、字符串连接，还可以实现列表、元组的连接和集合的并集运算。**
- 2.68 判断对错：在Python中可以使用for作为变量名。
- 2.69 判断对错：在Python中可以使用id作为变量名，但是不建议这样做。**
- 2.70 判断对错：一个数字5也是合法的Python表达式。
- 2.71 判断对错：执行语句`from math import sin`之后，可以直接使用`sin()`函数，例如`sin(3)`。
- 2.72 判断对错：一般来说，Python扩展库没有通用于所有版本Python的，安装时应选择与已安装Python的版本对应的扩展库。**
- 2.73 判断对错：Python变量名区分大小写，所以`student`和`Student`不是同一个变量。
- 2.74 判断对错：在Python 3.x中`reduce()`是内置函数。
- 2.75 判断对错：如果只需要math模块中的`sin()`函数，建议使用`from math import sin`来导入，而不要使用`import math`导入整个模块。
- 2.76 判断对错：表达式`pow(3, 2) == 3 ** 2`的值为True。**

2.77 判断对错：已知 $x = 3$ ，那么执行语句 $x += 6$ 之后， $x$ 的内存地址不变。

2.78 判断对错：安装Python扩展库时只能使用pip工具在线安装，如果安装不成功就没有别的办法了。

## 1000题库系列分享三

3.1 为什么应尽量从列表的尾部进行元素的增加与删除操作？

3.2 Python 3.x的range()函数返回一个\_\_\_\_\_。

3.3 编写程序，生成包含1000个0到100之间的随机整数，并统计每个元素的出现次数。

3.4 表达式`[3] in [1, 2, 3, 4]`的值为\_\_\_\_\_。

3.5 编写程序，用户输入一个列表和2个整数作为下标，然后使用切片获取并输出列表中介于2个下标之间的元素组成的子列表。例如用户输入`[1, 2, 3, 4, 5, 6]`和`2, 5`，程序输出`[3, 4, 5, 6]`。

3.6 列表对象的sort()方法用来对列表元素进行原地排序，该函数返回值为\_\_\_\_\_。

3.7 列表对象的\_\_\_\_\_方法删除首次出现的指定元素，如果列表中不存在要删除的元素，则抛出异常。

3.8 假设列表对象aList的值为`[3, 4, 5, 6, 7, 9, 11, 13, 15, 17]`，那么切片`aList[3:7]`得到的值是\_\_\_\_\_。

3.9 设计一个字典，并编写程序，用户输入内容作为“键”，然后输出字典中对应的“值”，如果用户输入的“键”不存在，则输出“您输入的键不存在！”

3.10 编写程序，生成包含20个随机数的列表，然后将前10个元素升序排列，后10个元素降序排列，并输出结果。

3.11 在Python中，字典和集合都是用一对\_\_\_\_\_作为界定符，字典的每个元素有两部分组成，即\_\_\_\_\_和\_\_\_\_\_，其中\_\_\_\_\_不允许重复。

3.12 使用字典对象的\_\_\_\_\_方法可以返回字典的“键-值对”，使用字典对象的\_\_\_\_\_方法可以返回字典的“键”，使用字典对象的\_\_\_\_\_方法可以返回字典的“值”。

3.13 假设有列表`a = ['name', 'age', 'sex']`和`b = ['Dong', 38, 'Male']`，请使用一个语句将这两个列表的内容转换为字典，并且以列表a中的元素为“键”，以列表b中的元素为“值”，这个语句可以写为\_\_\_\_\_。

3.14 假设有一个列表a，现要求从列表a中每3个元素取1个，并且将取到的元素组成新的列表b，可以使用语句\_\_\_\_\_。

3.15 使用列表推导式生成包含10个数字5的列表，语句可以写为\_\_\_\_\_。

3.16 \_\_\_\_\_（可以、不可以）使用del命令来删除元组中的部分元素。

3.17 表达式`[1, 2, 3]*3`的值为\_\_\_\_\_。

3.18 表达式`list(map(str, [1, 2, 3]))`的值为\_\_\_\_\_。

3.19 语句`x = 3 == 3, 5`执行结束后，变量x的值为\_\_\_\_\_。

- 3.20 已知 $x = 3$ ，并且 $\text{id}(x)$ 的返回值为496103280，那么执行语句 $x += 6$ 之后，表达式 $\text{id}(x) == 496103280$ 的值为\_\_\_\_\_。
- 3.21 已知 $x = 3$ ，那么执行语句 $x *= 6$ 之后， $x$ 的值为\_\_\_\_\_。
- 3.22 表达式 $3 \text{ in } [1, 2, 3, 4]$ 的值为\_\_\_\_\_。
- 3.23 使用列表与整数的乘法生成包含10个数字5的列表，可以写为\_\_\_\_\_。
- 3.24 任意长度的非空列表、元组和字符串中最后一个元素的下标为\_\_\_\_\_。
- 3.25 表达式 $\text{list}(\text{range}(1, 10, 3))$ 的值为\_\_\_\_\_。
- 3.26 表达式 $\text{list}(\text{range}(10, 1, -3))$ 的值为\_\_\_\_\_。
- 3.27 表达式 $\text{list}(\text{range}(5))$ 的值为\_\_\_\_\_。
- 3.28 已知 $a = [1, 2, 3]$ 和 $b = [1, 2, 4]$ ，那么 $\text{id}(a[1]) == \text{id}(b[1])$ 的执行结果为\_\_\_\_\_。
- 3.29 切片操作 $\text{list}(\text{range}(6))[:2]$ 执行结果为\_\_\_\_\_。
- 3.30 使用切片操作在列表对象 $x$ 的开始处增加一个元素3的代码为\_\_\_\_\_。
- 3.31 语句 $\text{sorted}([1, 2, 3], \text{reverse}=\text{True}) == \text{reversed}([1, 2, 3])$ 执行结果为\_\_\_\_\_。
- 3.32 表达式 $\text{sorted}([111, 2, 33], \text{key}=\text{lambda } x: \text{len}(\text{str}(x)))$ 的值为\_\_\_\_\_。
- 3.33 表达式 $\text{sorted}([111, 2, 33], \text{key}=\text{lambda } x: -\text{len}(\text{str}(x)))$ 的值为\_\_\_\_\_。
- 3.34 表达式 $\text{max}([111, 22, 3], \text{key}=\text{str})$ 的值为\_\_\_\_\_。
- 3.35 语句 $x = (3,)$ 执行后 $x$ 的值为\_\_\_\_\_。
- 3.36 语句 $x = (3)$ 执行后 $x$ 的值为\_\_\_\_\_。
- 3.37 已知 $x=3$ 和 $y=5$ ，执行语句  $x, y = y, x$  后 $x$ 的值是\_\_\_\_\_。
- 3.38 可以使用内置函数\_\_\_\_\_查看包含当前作用域内所有全局变量和值的字典。
- 3.39 可以使用内置函数\_\_\_\_\_查看包含当前作用域内所有局部变量和值的字典。
- 3.40 字典中多个元素之间使用\_\_\_\_\_分隔开，每个元素的“键”与“值”之间使用\_\_\_\_\_分隔开。