

第1部分

1. 下面哪些是操作符，哪些是值？

```
*  
'hello'  
-88.8  
-  
/  
+  
5
```

答案

操作符是+、-、*和/

值是'hello'、-88.8和5

2. 下面哪个是变量，哪个是字符串？

```
spam  
'spam'
```

答案

字符串是'spam'，变量是spam。字符串总是以引号开始和结束。

3. 说出基本的数据类型。

答案

数据类型是整数（int）、浮点数（float）、字符串（str）、布尔（bool）。

4. 表达式由什么构成？所有表达式都做什么事？

答案

表达式是值和操作符的结合。所有表达式都求值为（即归约为）一个值。

5. 本章介绍了赋值语句，如`spam = 10`。表达式和语句有什么区别？

答案

表达式求值为一个值。语句是一段逻辑。

6. 下列语句运行后，变量`bacon`的值是什么？

```
bacon = 20
bacon + 1
```

答案

`bacon`变量被设置为20。表达式`bacon + 1`并没有对`bacon`重新赋值（重新赋值需要一个赋值语句：`bacon = bacon + 1`）。

7. 下面两个表达式求值的结果是什么？

```
'spam' + 'spamspam'
'spam' * 3
```

答案

两个表达式都求值为字符串`'spamspamspam'`。

8. 为什么`eggs`是有效的变量名，而`100`是无效的？

答案

变量名不能以数字开始。

9. 哪3个函数能分别取得一个值的整型、浮点型或字符串版本？

答案

`int()`、`float()` 和 `str()` 函数将返回传入值的整型、浮点型和字符串版本。

10. 为什么这个表达式会导致错误？如何修复？

```
'I have eaten ' + 99 + ' burritos.'
```

答案

该表达式导致错误是因为，99是一个整数，只有字符串能用+操作符与其他字符串连接。正确的方式是'I have eaten ' + str(99) + ' burritos.'。

附加题：在线查找len()函数的Python文档。它在一个标题为“Built-in Functions”的网页上。扫一眼Python的其他函数的列表，查看round()函数的功能，在交互式环境中使用它。

第2部分

1. 布尔数据类型的两个值是什么？如何拼写？

答案

True和False，使用大写的T和F，其他字母是小写。

2. 三个逻辑运算操作符是什么？

答案

and、or和not。

3. 写出每个布尔操作符的真值表（也就是操作数的每种可能组合，以及操作的结果）。

答案

True and True是True。

True and False是False。

False and True是False。

False and False是False。

True or True是True。

True or False是True。

False or True是True。

False or False是False。

not True是False。

`not False`是`True`。

4. 以下表达式求值的结果是什么？

```
(5 > 4) and (3 == 5)
not (5 > 4)
(5 > 4) or (3 == 5)
not ((5 > 4) or (3 == 5))
(True and True) and (True == False)
(not False) or (not True)
```

答案

`False`

`False`

`True`

`False`

`False`

`True`

5. 6个比较操作符是什么？

答案

`==`、`!=`、`<`、`>`、`<=`和`>=`。

6. 等于操作符和赋值操作符的区别是什么？

答案

`==`是等于操作符，它比较两个值，求值为一个布尔值，而`=`是赋值操作符，将值保存在变量中。

7. 解释什么是条件，可以在哪里使用条件。

答案

条件是一个表达式，它用于控制流语句中，求值为一个布尔值。

8. 识别这段代码中的3个语句块：

```
spam = 0
if spam == 10:
    print('eggs')
    if spam > 5:
        print('bacon')
    else:
        print('ham')
    print('spam')
print('spam')
```

答案

```
# 3个语句块是if语句中的全部内容，以及print('bacon')和print('ham')这两行。
spam = 0
# 语句块1
if spam == 10:
    print('eggs')
    # 语句块2
    if spam > 5:
        print('bacon')
    # 语句块3
    else:
        print('ham')
    print('spam')
print('spam')
```

9. 编写代码，如果变量spam是1，就打印Hello，如果变量是2，就打印Howdy，如果变量中存放其他值，就打印Greetings!

答案

```
if spam == 1:
    print('Hello')
elif spam == 2:
    print('Howdy')
else:
    print('Greetings!')
```

10. 如果程序陷在一个无限循环中，你可以按什么键？

答案

按Ctrl-C来停止陷在无限循环中的程序。

11. break和continue之间的区别是什么？

答案

`break`语句将执行移出循环，接着循环之后执行。`continue`语句将执行移到循环的开始。

12. 在for循环中，`range(10)`、`range(0, 10)`和`range(0, 10, 1)`之间的区别是什么？

答案

它们都是做同样的事。`range(10)`调用产生的范围是从0直到（但不包括）10，`range(0, 10)`明确告诉循环从0开始，`range(0, 10, 1)`明确告诉循环每次迭代让变量增加1。

13. 编写一小段程序，利用for循环，打印出从1到10的数字。然后利用while循环，编写一个等价的程序，打印出从1到10的数字。

答案

```
# 解法1
for i in range(1, 11):
    print(i)

# 解法2
i = 1
while i <= 10:
    print(i)
    i = i + 1
```

14. 如果在名为spam的模块中，有一个名为bacon()的函数，那么在导入spam模块后，如何调用它？

答案

该函数的调用方式是

`spam.bacon()`。

附加题：在因特网上查找`round()`和`abs()`函数，弄清楚它们的作用。在交互式环境中尝试使用它们。

第3部分

1. 为什么在程序中加入函数会有好处？

答案

函数减少了重复的代码。这让程序更短，更容易阅读，更容易修改。

2. 函数中的代码何时执行：在函数被定义时，还是在函数被调用时？

答案

函数中的代码在函数被调用时执行，而不是在函数定义时。

3. 什么语句创建一个函数？

答案

`def`语句定义了（即创建了）一个函数。

4. 一个函数和一次函数调用有什么区别？

答案

函数包含`def`语句和在`def`子句中的代码。函数调用让程序执行转到函数内，函数调用求值为该函数的返回值。

5. Python程序中有多少全局作用域？有多少局部作用域？

答案

在调用一个函数时，创建了一个全局函数和一个局部作用域。

6. 当函数调用返回时，局部作用域中的变量发生了什么？

答案

函数返回时，局部作用域被销毁，其中所有的变量都被遗忘了。

7. 什么是返回值？返回值可以作为表达式的一部分吗？

答案

返回值是函数调用求值的结果。像所有值一样，返回值可以作为表达式的一部分。

8. 如果函数没有返回语句，对它调用的返回值是什么？

答案

如果函数没有`return`语句，它的返回值就是`None`。

9. 如何强制函数中的一个变量指的是全局变量？

答案

`global`语句强制函数中的一个变量引用该全局变量。

10. `None`的数据类型是什么？

答案

`None`的数据类型是`NoneType`。

11. `import areallyourpetsnamederic`语句做了什么？

答案

`import`语句导入了`areallyourpetsnamederic`模块（顺便说一句，这不是一个真正的Python模块）。

12. 如果在名为`spam`的模块中，有一个名为`bacon()`的函数，在引入`spam`后，如何调用它？

答案

该函数可以通过`spam.bacon()`调用。

13. 如何防止程序在遇到错误时崩溃？

答案

将可能导致错误的代码行放在一个`try`子句中。

14. `try`子句中发生了什么？`except`子句中发生了什么？

答案

可能导致错误的代码放在 `try` 子句中。发生错误时要执行的代码放在`except`子句中。

第4部分

1. 什么是`[]`？

答案

空的列表值，它是一个列表，不包含任何列表项。这类似于`''`是空的字符串值。

2. 如何将`'hello'`赋给列表的第三个值，而列表保存在名为`spam`的变量中？（假定变量包含`[2, 4, 6, 8, 10]`）。

答案

`spam[2] = 'hello'`（注意，列表中的第3个值下标是2，因为第1个值下标是0。）

对接下来的3个问题，假定`spam`包含列表`['a', 'b', 'c', 'd']`。

3. `spam[int('3' * 2) / 11]`求值为多少？

答案

`'d'`（注意`'3' * 2`是字符串`'33'`，它被传入`int()`，然后再除以11。这最终求值为3。在使用值的地方，都可以使用表达式）。

4. `spam[-1]`求值为多少？

答案

`'d'`（负数下标从末尾倒数）。

5. `spam[:2]`求值为多少？

答案

```
['a', 'b']
```

对接下来的3个问题。假定`bacon`包含列表`[3.14, 'cat', 11, 'cat', True]`。

6. `bacon.index('cat')`求值为多少？

答案

```
1
```

7. `bacon.append(99)`让`bacon`中的列表值变成什么样？

答案

```
[3.14, 'cat', 11, 'cat', True, 99]
```

8. `bacon.remove('cat')`让`bacon`中的列表时变成什么样？

答案

```
[3.14, 11, 'cat', True]
```

9. 列表连接和复制的操作符是什么？

答案

列表连接的操作符是`+`，复制的操作符是`*`（这和字符串一样）。

10. `append()`和`insert()`列表方法之间的区别是什么？

答案

`append()` 只会将值添加在列表末尾，而`insert()` 可以将值添加在列表的任何位置。

11. 从列表中删除值有哪两种方法？

答案

`del` 语句和 `remove()` 列表方法是从列表中删除值的两种方法。

12. 请说出列表值和字符串的几点相似之处。

答案

列表和字符串都可以传入 `len()`，都有下标和切片，用于 `for` 循环，连接或复制，并与 `in` 和 `not in` 操作符一起使用。

13. 列表和元组之间的区别是什么？

答案

列表是可以修改的，它们可以添加值、删除值和修改值。元组是不可修改的，它们根本不能改变。而且，元组使用的是括号（和），而列表使用的是方括号 [和]。

14. 如果元组中只有一个整数值42，如何输入该元组？

答案

`(42,)`（末尾的逗号是必须的）。

15. 如何从列表值得到元组形式？如何从元组值得到列表形式？

答案

分别使用 `tuple()` 和 `list()` 函数。

16. “包含”列表的变量，实际上并未真地直接包含列表。它们包含的是什么？

答案

它们包含对列表值的引用。

17. `copy.copy()`和`copy.deepcopy()`之间的区别是什么？

答案

`copy.copy()` 函数将浅拷贝列表，而`copy.deepcopy()` 函数将深拷贝列表。也就是说，只有`copy.deepcopy()`

第5部分

1. 空字典的代码是怎样的？

答案

两个花括号：`{}`

2. 一个字典包含键'foo'和值42，看起来是怎样的？

答案

`{'foo': 42}`

3. 字典和列表的主要区别是什么？

答案

保存在字典中的项是无序的，而列表中的项是有序的。

4. 如果spam是{'bar': 100}，你试图访问spam['foo']，会发生什么？

答案

会得到`KeyError`错误。

5. 如果一个字典保存在spam中，表达式'cat' in spam和'cat' in spam.keys()之间的区别是什么？

答案

没有区别。`in`操作符检查一个值是不是字典中的一个键。

6. 如果一个字典保存在变量中，表达式`'cat' in spam`和`'cat' in spam.values()`之间的区别是什么？

答案

`'cat' in spam`检查字典中是不是有一个 `'cat'` 键，而`'cat' in spam.values()` 检查是否有一个值 `'cat'` 对应于`spam`中的某个键。

7. 什么模块和函数可以用于“漂亮打印”字典值？

答案

```
pprint.pprint()
```

第6部分

1. 什么是转义字符？

答案

转义字符表示字符串中的一些字符，这些字符用别的方式很难在代码中打出来。

2. 转义字符`\n`和`\t`代表什么？

答案

`\n`是换行符，`\t`是制表符。

3. 如何在字符串中放入一个倒斜杠字符`\`？

答案

`\`转义字符表示一个反斜杠。

4. 字符串`"Howl's Moving Castle"`是有效字符串。为什么单词中的单引号没有转义，却没有问题？

答案

Howl's 中的单引号没有问题，因为你用了双引号来标识字符串的开始和结束。

5. 如果你不希望在字符串中加入\n，怎样写一个带有换行的字符串？

答案

多行字符串让你在字符串中使用换行符，而不必用\n转义字符。

6. 下面的表达式求值为什么？

- 'Hello world!'[1]
- 'Hello world!'[0:5]
- 'Hello world!':5]
- 'Hello world!':[3:]

答案

```
'e'  
'Hello'  
'Hello'  
'lo world!'
```

7. 下面的表达式求值为什么？

- 'Hello'.upper()
- 'Hello'.upper().isupper()
- 'Hello'.upper().lower()

答案

```
'HELLO'  
True  
'hello'
```

8. 下面的表达式求值为什么？

- 'Remember, remember, the fifth of November.'.split()
- '-'.join('There can be only one.'.split())

答案

```
['Remember,', 'remember,', 'the', 'fifth', 'of', 'November.']  
'There-can-be-only-one.'
```

9. 如何去掉字符串开始或末尾的空白字符?

答案

```
strip() 移除字符串两端的空白字符。
```