

第三章 第七节 Python 基础综合练习 2

Python 基础综合练习 2

练习 1：用户输入INPUT和条件语句

编写一个程序，要求用户输入一个数字，然后多次打印以下句子：“I am back to check on my skills!”如果数字大于 10，则打印这句话：“Python conditions and loops are a piece of cake.” 假设您只能传递正整数。

练习 2：小写和大写字符

下面是一个字符串，text。它包含一长串字符。您的任务是迭代字符串的字符，计算大写字母和小写字母，并打印结果：

LOWERCASE: <count>

UPPERCASE: <count>

```
text = "ABgvddVICJSdkeprsmgn UTPDvndhtuwPOTNRSjfisurNSUesajdsa"
```

Use the string.islower() and string.isupper() methods to check if a string contains lowercase and uppercase characters.

第三章 第七节 Python 基础综合练习 2

Python 基础综合练习 2

练习3：构建三角形

创建一个名为 `is_triangle_possible()` 的函数，该函数接受三个正数。如果可以从给定长度的线段创建三角形，则应返回 `True`，否则返回 `False`。使用 3 个数字，有时（但并非总是）可以创建一个三角形：您不能从 `a = 13`、`b = 2` 和 `c = 3` 创建三角形，但可以从 `a = 13`、`b = 9` 和 `c = 10`。

练习 4：从另一个函数调用一个函数

创建两个函数：`print_Five_times ()` 和 `speak ()`。函数 `print_Five_times()` 应该接受一个参数（称为句子）并打印五次。函数 `speak(sentence,repeat)` 应该有两个参数：`sentence`（一串字母）和 `repeat`（一个默认值设置为 `False` 的布尔值）。如果 `repeat` 参数设置为 `False`，该函数应该只打印一个句子一次。如果 `repeat` 参数设置为 `True`，该函数应调用 `print_Five_times ()` 函数。

第三章 第七节 Python 基础综合练习 2

Python 基础综合练习 2

练习5：循环和条件语句

编写一个名为 `find_greater_than()` 的函数，它接受两个参数：数字列表和整数阈值。

该函数应创建一个新列表，其中包含输入列表中大于给定阈值的所有数字。

结果列表中的数字顺序应与输入列表中的数字顺序相同。例如：

```
find_greater_than([-3, 2, 8, 15, 31, 5, 4, 8], 5)
[8, 15, 31, 8]
```

练习 6：嵌套循环和条件语句

编写一个名为 `find_censored_words()` 的函数，该函数接受字符串列表和特殊字符列表作为其参数，并在单独的行中一一打印其中的所有审查单词。

如果一个单词至少有一个来自 `special_chars` 列表的字符，则该单词被视为已被审查。使用 `word_list` 变量来测试您的函数。我们为您准备了两份清单：

例如：

```
special_chars = ['!', '@', '#', '$', '%', '^', '&', '*']
word_list = ['se#set', 'Ver*%&$lo', 'di$#^$nt', 'c*%e', 'is', '#%$#@!@#%^$#']
```

('go', 'stay')

第三章 第七节 Python 基础综合练习 2

Python 基础综合练习 2

练习7： Lists and Tuples 列表和元组

创建函数 `find_short_long_word(words_list)`。该函数应返回列表中最短单词和列表中最长单词（按该顺序）的元组。如果有多个单词符合最短单词的条件，则返回列表中第一个最短的单词。如果有多个单词符合最长单词的条件，则返回列表中最后一个最长的单词。例如，对于以下列表：

例如：

```
words = ['go', 'run', 'play', 'see', 'my', 'stay']
```

结果应该是：

```
('go', 'stay')
```

练习 8： Dictionaries 词典

如您所见，我们已经为您准备了 `test_results` 变量。您的任务是迭代字典的值并打印得分低于 45 分的人员的所有姓名。

```
test_results = {'Jenny': 50, 'Mathew': 45, 'Joe': 30, 'Peter': 40, 'Agness': 50, 'Samantha': 45, 'John': 20}
```

结果是：

Joe

Peter

John

第三章 第七节 Python 基础综合练习 2

Python 基础综合练习 2

练习 9: Dictionaries + 更多词典

编写一个名为 `consonant_vowels_count(frequency_dictionary, Vowels)` 的函数，该函数将字典和元音列表作为参数。字典的键是字母，值是它们的频率。该函数应按以下格式打印辅音总数和元音总数：

VOWELS: <vowels>

CONSONANTS: <consonants>

例如： `frequencies_dictionary = {'a': 2, 's': 1, 'i': 3, 'u': 3, 'm': 4, 'l': 2, 'z': 1, 'n': 3, 'k': 1, 'e': 4, 'y': 2, 'd': 2, 'o': 1}`

输出：

VOWELS: 13

CONSONANTS: 16

第三章 第七节 Python 基础综合练习 2

Python 基础综合练习 2

练习 10：字符串加密

实施凯撒密码CAESAR CIPHER。这是一种简单的加密技术，它将单词中的每个字母替换为字母表中固定数量位置的另一个字母。

例如，考虑字符串“word”。如果我们将字母表中的每个字母向下移动一个位置，我们就得到了“xpse”。移动 2 个位置得到字符串“yqtf”。首先用字母表中的每个字母定义一个字符串：

```
alphabet = "abcdefghijklmnopqrstuvwxyz".
```

输出：

cdefghijklmnopqrstuvwxyzab