

سری دوم تمرینات توابع

به سوالات زیر پاسخ دهید.:

تابعی بنویسید تا حجم کره را با دریافت شعاع آن محاسبه نماید.

حجم کره براساس رابطه زیر محاسبه می شود

$$\frac{4}{3}\pi r^3$$

In [6]:

```
1 import math
2
3
4 def vol(rad):
5     return (4 / 3) * (math.pi) * (rad ** 3)
```

In [7]:

```
1 # Check
2 vol(2)
```

Out[7]:

33.510321638291124

In [8]:

```
1 math.pi
```

Out[8]:

3.141592653589793

تابعی بنویسید که بررسی کند یک عدد در بازه بسته داده شده قرار دارد یا نه و برای هرکدام از حالات یک جمله چاپ کند
تابع سه ورودی خواهد داشت، مقدار عددی که باید بررسی شود، ابتدای بازه و انتهای بازه

In [10]:

```
1 def ran_check(num,low,high):
2     if num in range(low, high + 1):
3         print(f'{num} is in the range between {low} and {high}')
4     else:
5         print('The number is outside of the range')
```

In [13]:

```
1 # Check
2 ran_check(5,2,7)
```

5 is in the range between 2 and 7

در صورتیکه بخواهیم برای گزارش نتیجه یک مقدار بولین برگرداند چطور؟

In [14]:

```
1 def ran_bool(num,low,high):
2     return num in range(low, high + 1)
```

In [16]:

```
1 ran_bool(15,1,10)
```

Out[16]:

False

تابع پایتونی بنویسید که یک رشته را می پذیرد و تعداد حروف بزرگ و کوچک را محاسبه می کند.

رشته نمونه: 'Hello Mr. Rogers, how are you this fine Tuesday' خروجی مورد انتظار: تعداد حروف بزرگ: 4 تعداد کاراکتر های کوچک: 33

راهنمایی: دو متد رشته ای که ممکن است مفید باشد: `.islower()` و `.isupper()`.

اگر احساس کردید، بهتر است ماژول Collections را برای حل این مسئله بررسی کنید!

In [17]:

```
1 def up_low(text):
2     d = {'upper' : 0, 'lower' : 0}
3
4     for char in text:
5         if char.isupper():
6             d['upper'] += 1
7         elif char.islower():
8             d['lower'] += 1
9         else:
10            pass
11    print(f'Original String : {text}')
12    print(f'No. of Upper case characters : {d['upper']}")
13    print(f'No. of Lower case characters : {d['lower']}")
```

In [18]:

```
1 s = 'Hello Mr. Rogers, how are you this fine Tuesday?'
2 up_low(s)
```

Original String : Hello Mr. Rogers, how are you this fine Tuesday?

No. of Upper case characters : 4

No. of Lower case characters : 33

تابع پایتونی بنویسید که یک لیست را می گیرد و لیست جدیدی با عناصر منحصر به فرد لیست اول برمی گرداند.

Sample List : [1,1,1,1,2,2,3,3,3,3,4,5]

Unique List : [1, 2, 3, 4, 5]

In [19]:

```
1 def unique_list_set(lst):
2     return list(set(lst))
```

In [20]:

```
1 unique_list_set([1,1,1,1,2,2,3,3,3,3,4,5])
```

Out[20]:

[1, 2, 3, 4, 5]

In [21]:

```
1 def unique_list_for_loop(lst):
2
3     x = []
4
5     for a in lst:
6         if a not in x:
7             x.append(a)
8     return x
9
10
```

In [22]:

```
1 unique_list_for_loop([1,1,1,1,2,2,3,3,3,3,4,5])
```

Out[22]:

[1, 2, 3, 4, 5]

تابع پایتونی بنویسید که تمام اعداد یک لیست را در هم ضرب کند.

Sample List : [1, 2, 3, -4]

Expected Output : -24

In [23]:

```
1 def multiply(numbers):
2     total = 1
3
4     for x in numbers:
5         total *= x # total = total * x
6     return total
```

In [24]:

```
1 multiply([1,2,3,-4])
```

Out[24]:

-24

تابع پایتونی بنویسید که چک کند که یک کلمه یا عبارت پالیندروم است یا خیر.

توجه: پالیندروم، کلمه، عبارت یا دنباله ای است که از هر دو طرف مانند هم خوانده می شود، به عنوان مثال، madam، kayak، racecar، یا عبارت "nurses run".

راهنمایی: شما ممکن است بخواهید به روش `replace()` در رشته برای کمک به مقابله با فضاها نگاه کنید. همچنین جستجو در گوگل برای روش های باور نکردنی برای برعکس کردن یک رشته در پایتون، با استفاده از نوتاسیون اسلایسینگ.

In [27]:

```
1 def palindrome(s):
2     s = s.replace(' ', '')
3
4     return s == s[::-1]
```

In [31]:

```
1 palindrome('raccar')
```

Out[31]:

True

تمرین سخت:

تابع پایتونی بنویسید که بررسی کند که یک رشته پانگرام است یا خیر. (فرض کنید رشته وارد شده هیچ علامت نگارشی ندارد)

توجه: پانگرام ها کلمات یا جملاتی هستند که حداقل یک بار هر حرف الفبای انگلیسی را دارا می باشند. به عنوان مثال: "The quick brown fox jumps over the lazy dog"

راهنمایی: شما ممکن است بخواهید از روش `replace()` برای رهایی از فضاها استفاده کنید.

میتوانید به این لینک مراجعه کنید: [string module \(https://stackoverflow.com/questions/16060899/alphabet-range-in-python\)](https://stackoverflow.com/questions/16060899/alphabet-range-in-python).

برای مثال، بیشتر به لینک نگاه کنید: [set comparisons \(https://medium.com/better-programming/a-visual-guide-to-set-comparisons\)](https://medium.com/better-programming/a-visual-guide-to-set-comparisons).

In [32]:

```
1 import string
2
3 def ispangram(str1, alphabet=string.ascii_lowercase):
4
5     alphabet = set(alphabet)
6
7     # Remove spaces from str1
8     str1 = str1.replace(' ', '')
9
10    # Lowercase all strings in the passed string
11    str1 = str1.lower()
12
13    # Grab all unique letters in the string as a set
14    str1 = set(str1)
15
16    # Now check that the alphabet set is same as string set
17
18    return str1 == alphabet
19
```

In [33]:

```
1 ispangram("The quick brown fox jumps over the lazy dog")
```

Out[33]:

True

In [34]:

```
1 string.ascii_lowercase
```

Out[34]:

'abcdefghijklmnopqrstuvwxyz'