Python String Exercise

1: Given a string of odd length greater than 7, return a new string made of the middle three characters of a given String

Given:

Case 1

```
str1 = "JhonDipPeta"
```

Output

```
Dip
```

Case 2

```
str2 = "JaSonAy"
```

Output

```
Son
```

2: Given two strings, s1 and s2, create a new string by appending s2 in the middle of s1

Given:

```
s1 = "Ault"
```

```
s2 = "Kelly"
```

Expected Output:

```
AuKellylt
```

3: Given two strings, s1, and s2 return a new string made of the first, middle, and last characters each input string

Given:

```
s1 = "America"

s2 = "Japan"
```

Expected Output:

```
AJrpan
```

4: Arrange string characters such that lowercase letters should come first

Given an input string with the combination of the lower and upper case arrange characters in such a way that all lowercase letters should come first.

Given:

```
str1 = PyNaTive
```

Expected Output:

```
yaivePNT
```

5: Count all lower case, upper case, digits, and special symbols from a given string

`str1 = "P@#yn26at^&i5ve"`

Expected Outcome:

```
Total counts of chars, digits,and symbols



Chars = 8

Digits = 3

Symbol = 4
```

6: Given two strings, s1 and s2, create a mixed String using the following rules

Note: create a third-string made of the first char of s1 then the last char of s2, Next, the second char of s1 and second last char of s2, and so on. Any leftover chars go at the end of the result.

Given:

```
s1 = "Abc"

s2 = "Xyz"
```

Expected Output:

```
AzbycX
```

7: String characters balance Test

We'll assume that a String s1 and s2 is balanced if all the chars in the s1 are there in s2. characters' position doesn't matter.

Given:

Case 1:

```
s1 = "Yn"

s2 = "PYnative"
```

Expected Output:

```
True
```

Case 2:

```
s1 = "Ynf"

s2 = "PYnative"
```

Expected Output:

```
False
```

8: Find all occurrences of "USA" in a given string ignoring the case

Given:

```
str1 = "Welcome to USA. usa awesome, isn't it?"
```

Expected Outcome:

```
The USA count is: 2
```

9: Given a string, return the sum and average of the digits that appear in the string, ignoring all other characters

Given:

```
str1 = "English = 78 Science = 83 Math = 68 History = 65"
```

Expected Outcome:

```
sum is 294

average is 73.5
```

10: Given an input string, count occurrences of all characters within a string

Given:

```
str1 = "Apple"
```

Expected Outcome:

```
{'A': 1, 'p': 2, 'l': 1, 'e': 1}
```

11: Reverse a given string

Given:

```
str1 = "PYnative"
```

Expected Output:

```
evitanYP
```

12: Find the last position of a substring “Emma” in a given string

Given:

```
str1 = "Emma is a data scientist who knows Python. Emma works at google."
```

Expected Output:

```
Last occurrence of Emma starts at index 43
```

13: Split a given string on hyphens into several substrings and display each substring

Given:

```
str1 = Emma-is-a-data-scientist
```

Expected Output:

```
Displaying each substring
```

```
Emma

is

a

data

scientist
```

14: Remove empty strings from a list of strings

Given:

```
str_list = ["Emma", "Jon", "", "Kelly", None, "Eric", ""]
```

Expected Output:

```
Original list of sting

['Emma', 'Jon', '', 'Kelly', None, 'Eric', '']



After removing empty strings

['Emma', 'Jon', 'Kelly', 'Eric']
```

15: Remove special symbols/Punctuation from a given string

Given:

```
str1 = "/*Jon is @developer & musician"
```

Expected Output:

```
"Jon is developer musician"
```

16: Removal all the characters other than integers from a string

Given:

```
str1 = 'I am 25 years and 10 months old'
```

Expected Output:

```
2510
```

17: Find words with both alphabets and numbers

Given:

```
str1 = "Emma25 is Data scientist50 and AI Expert"
```

Expected Output:

```
Emma25
scientist50
```

18: Replace each punctuation with # in the following string

Given:

```
str1 = '/*Jon is @developer & musician!!'
```

Expected Output:

```
##Jon is #developer # musician##
```