

▼ Python Assignment 5

Zaid Shariff

2282437

▼ (1) Implement the following Python regular expression concepts with examples

(a) Metacharacters: ., ^, \$, *, +, ?, {}, [], |, \, ()

```
import re
```

```
string = "apples"
match = re.search(r'^s', string)
print(match)
match = re.search(r'^a', string)
print(match)
match = re.search(r's$', string)
print(match)
match = re.search(r'a|z|s', string)
print(match)
match = re.search(r'...', "aaa")
print(match)
match = re.search(r'a*c', "abc")
print(match)
match = re.search(r'ab+c', "abbbc")
print(match)
```

None

```
<re.Match object; span=(0, 1), match='a'>
<re.Match object; span=(5, 6), match='s'>
<re.Match object; span=(0, 1), match='a'>
<re.Match object; span=(0, 3), match='aaa'>
<re.Match object; span=(2, 3), match='c'>
<re.Match object; span=(0, 5), match='abbbc'>
```

(b) Regex functions: search(), match(), findall(), split(), sub()

```
phrase = 'Christ University'
match="University"

match = re.search(match, phrase)
print(match)
```

```
<re.Match object; span=(7, 17), match='University'>
```

```
pattern = '^a....s$'
test_string = 'apples'
result = re.match(pattern, test_string)
if result:
    print("Search successful.")
else:
    print("Search unsuccessful.")
```

```
Search successful.
```

```
string = """Hello my Number is 123456789 and
           my friend's number is 987654321"""
```

```
regex = '\d+'
```

```
match = re.findall(regex, string)
print(match)
```

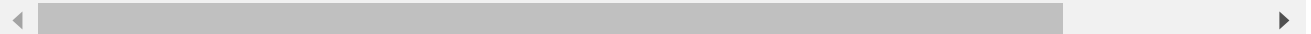
```
['123456789', '987654321']
```

```
from re import split
```

```
string = """Hello my Number is 123456789 and
           my friend's number is 987654321"""
```

```
lst=split('\W+', string)
print(lst)
```

```
['Hello', 'my', 'Number', 'is', '123456789', 'and', 'my', 'friend', 's', 'number', 'i
```



▼ (2) Demonstrate the following Numpy operations:

(a) Array Creation `array()`, `arange()`, `zeros`, `ones`, `full()`, `eye()`, `random()`, `linspace()`

```
import numpy as np
```

```
aray=np.array([20,11,55,36,78])
print(array)
```

```
[20 11 55 36 78]
```

```
steps=np.arange(0,100,10)
print(steps)
```

```
[ 0 10 20 30 40 50 60 70 80 90]
```

```
zero_matrix=np.zeros((3,3),int)
zeros_matrix=np.zeros((3,3),float)
print(zero_matrix)
print("\n")
print(zeros_matrix)
```

```
[[0 0 0]
 [0 0 0]
 [0 0 0]]
```

```
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
```

```
ones_matrix=np.ones((3,3),int)
oness_matrix=np.ones((3,3),float)
print(ones_matrix)
print("\n")
print(oness_matrix)
```

```
[[1 1 1]
 [1 1 1]
 [1 1 1]]
```

```
[[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]
```

```
filled=np.full([4,4],22)
print(filled)
```

```
[[22 22 22 22]
 [22 22 22 22]
 [22 22 22 22]
 [22 22 22 22]]
```

```
identity_matrix=np.eye(4,5,k=0)
identity_matrix1=np.eye(4,5,k=-3)
identity_matrix2=np.eye(4,5,k=1)
print(identity_matrix)
print("\n")
print(identity_matrix1)
print("\n")
print(identity_matrix2)
```

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]]
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0.]]
```

```
[[0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

```
random_nos=np.random.random((3,3,3))
print(random_nos)
```

```
[[[0.59284462 0.84426575 0.85794562]
  [0.84725174 0.6235637 0.38438171]
  [0.29753461 0.05671298 0.27265629]]

 [[0.47766512 0.81216873 0.47997717]
  [0.3927848 0.83607876 0.33739616]
  [0.64817187 0.36824154 0.95715516]]

 [[0.14035078 0.87008726 0.47360805]
  [0.80091075 0.52047748 0.67887953]
  [0.72063265 0.58201979 0.53737323]]]
```

```
np.random.seed(0)
random_int=np.random.randint( 1,25 )
print(random_int)
```

13

```
lins=np.linspace(2.0, 3.0, num=10)
print(lins)
```

```
[2.          2.11111111 2.22222222 2.33333333 2.44444444 2.55555556
 2.66666667 2.77777778 2.88888889 3.          ]
```

(b) Attributes of arrays Determining the size, shape, memory consumption, and data types of arrays – ndim, shape, size, dtype, itemsize and data

```
array=np.array([[1,2,"dsd"],[4,5,6],[7,8,9]])
print(array)
print(array.ndim)

print(array.shape)

print(array.size)

print(array.dtype)
```

```
print(array.itemsize)
```

```
print(array.data)
```

```
[[ '1' '2' 'dsd']  
 [ '4' '5' '6']  
 [ '7' '8' '9']]  
2  
(3, 3)  
9  
<U21  
84  
<memory at 0x7fe33d752bb0>
```

(c) Indexing of arrays Getting and setting the value of individual array elements

```
elemet=array[0,0]  
print(elemet)
```

```
1
```

```
array[0][0]=50  
print(array)
```

```
[[ '50' '2' 'dsd']  
 [ '4' '5' '6']  
 [ '7' '8' '9']]
```

(d) Slicing of arrays Getting and setting smaller subarrays within a larger array

```
print(array[:2])  
array[:2]=np.full([2,3],6)  
print("\n")  
print(array[:2])
```

```
[[ '50' '2' 'dsd']  
 [ '4' '5' '6']]
```

```
[[ '6' '6' '6']  
 [ '6' '6' '6']]
```

(e) Reshaping of arrays Changing the shape of a given array – reshape()

```
print(array)  
print("\n")  
new=np.reshape(array,(1,9))  
print(new)
```

```
[[ '6' '6' '6']
```

```
['6' '6' '6']  
['7' '8' '9']
```

```
[['6' '6' '6' '6' '6' '6' '7' '8' '9']]
```

✓ 0s completed at 7:53 PM

