

## Programming Language Learning Series

### Mastery of Python Language

#### (Interview Questions/Assignment-Numpy Array)

---

Q1: Write a function to compute  $1/2+2/3+3/4+\dots+n/n+1$  with a given  $n$  ( $n>0$ ).

Q2: Write a function to find the sum of all the multiples of 3 or 5 below 1000.

Q3: A palindromic number reads the same both ways. The largest palindrome made from the product of two 2-digit numbers is  $9009 = 91 \times 99$ . Write a function to find the largest palindrome made from the product of two 3-digit numbers.

Q4: We count 35 heads and 94 legs among the chickens and rabbits in a farm. Write a python function that returns how many rabbits and how many chickens do we have.

Q5: Given a text file as input, we are interested to computing the following text analytics on that input:

- Compute the number of words in the given file
- Find the 10 most frequent words in the given file
- Find the number of times a given word appears in the file

Assuming that we want to develop a solution for the required text analytics using procedural abstractions. Which abstraction do you prefer and why?

Write a function `pdsist(xs)` which returns a matrix of the pairwise distance between the collection of vectors in `xs` using Euclidean distance.

Recall that Euclidean distance between two vectors  $x$

and  $y$

is

$$d(x,y)=\sqrt{\sum (y-x)^2}$$

Find the square distance matrix for

```
xs = np.array([[0.20981496, 0.54777461, 0.9398527 ],
               [0.63149939, 0.935947  , 0.29834026],
               [0.46302941, 0.25515557, 0.0698739 ],
               [0.38192644, 0.42378508, 0.26055664],
```

## Programming Language Learning Series

### Mastery of Python Language

#### (Interview Questions/Assignment-Numpy Array)

---

```
[0.46307302, 0.05943961, 0.60204931]])
```

Do this **without** using any for loops.

```
ns = np.arange(1, 13)
```

```
n = len(ns)
```

```
m = np.empty((n, n), dtype='int')
```

```
for i, x in enumerate(ns):
```

```
    for j, y in enumerate(ns):
```

```
        m[i, j] = x*y
```

```
m
```

```
ns[:, None] * ns[None, :]
```

```
a = np.array([10,20]) diff between print(a) & a
```

array slicing is view/copy

array indexing a[::-1], a[::-1,::-1],..

apply broad casting rules:

```
M = np.ones((2, 3))
```

```
a = np.arange(3)
```

**Programming Language Learning Series**  
**Mastery of Python Language**  
**(Interview Questions/Assignment-Numpy Array)**

---

```
a = np.arange(3).reshape((3, 1))
```

```
b = np.arange(3)
```

```
M = np.ones((3, 2))
```

```
a = np.arange(3)
```

shape of  $M + a$ ? why?

masking  $\&$ ,  $|$  &  $\wedge$  rules

```
temp= [10, 20, 30]
```

```
np.sum(temp>20)
```

```
(temp > 15 & temp < 10)
```

```
np.sum((temp > 15 & temp < 10))
```

```
bin(42 & 59)
```

Calculate the pairwise distance matrix between the following points

- (0,0)
- (4,0)
- (4,3)
- (0,3)