

✓ Congratulations! You passed!

TO PASS 70% or higher

Keep Learning

GRADE  
92.85%

## Readiness Quiz

LATEST SUBMISSION GRADE

92.85%

- For a given input list: 1, 2, 3, 4

1 / 1 point

1. Cube (element to power 3) each element
2. Return the results as a list

```
1 def get_cubed(lst):
2     ...
3     INPUT: LIST (containing numeric elements)
4     OUTPUT: LIST (cubed value of each even number in originals list)
5     return a list containing each element cubed
6     ...
7
8     cubes = []
9     for number in lst:
10        cube = number*number*number
11        cubes.append(cube)
12    return cubes
```

Run

Reset

No Output

✓ Correct

Good Job!

- For a given input list: 1,2,3,4,5,6,7

1 / 1 point

1. Inspect each number in the input list and determine if it is even
2. Next square the even values
3. Finally return the squared evens in a list

```
1 def get_squared_evens(lst):
2     ...
3     INPUT: LIST (containing numeric elements)
4     OUTPUT: LIST (squared value of each even number in originals list)
5     return squared evens in a list
6     ...
7
8     squares = []
9     for num in lst:
10        if (num % 2) == 0:
11            square = num*num
12            squares.append(square)
13    return squares
```

Run

Reset

No Output

✓ Correct

Good job!

- Which of the following is/are NOT native or built-in data types in Python?

1 / 1 point

- boolean
- integer
- float
- heap

✓ Correct

Correct!

- string
- varchar

**Correct**

Correct!

4. For given input lists: a,b,c and 1,2,3

1 / 1 point

Create a dictionary from two input lists

```
1 def make_dict(lst1,lst2):
2     ...
3     INPUT: LST1, LST2
4     OUTPUT: DICT (LST 1 are the keys and LST2 are the values)
5     Given equal length lists create a dictionary where the first list is the keys
6     ...
7     list = {}
8     for key, value in zip(lst1, lst2):
9         list[key]=value
10    return list
11
```

Run

Reset

No Output

**Correct**

Good job!

5. Mutable data types/collections in Python can be changed in place. Immutable ones can not change in place. Which of the following are mutable?

1 / 1 point

- bool
- int
- float
- set

**Correct**

Correct!

- list

**Correct**

Correct!

- string
- tuple
- complex

6. Which of the following is **NOT** true about Python?

1 / 1 point

- Python code can run in IPython and Jupyter notebooks
- Python allows for the inclusion of comments and pseudocode to better organize code
- Users can save .py files with an editor then subsequently execute them from the command line
- Base Python automatically parallelizes processing across cores when multiple cores are available
- Python allows users to save multiple functions in a .py file then import those functions in a different file

**Correct**

Correct. Base Python by default processes code with a single core. Multiprocessing requires loading additional libraries/packages.

7. For a given input list: abbccddddeeeeefffffgggggghhhhhh

1 / 1 point

Return a dictionary of character counts

1. Count the number of times each character appears in the string
2. Characters with a count of 0 should not be included in the output dictionary

```
1 def count_characters(string):
2     ...
3     INPUT: STRING
4     OUTPUT: DICT (with counts of each character in input string)
5
6     Return a dictionary of character counts
7
```

```

8     all_freq = {}
9     for i in string:
10        if i in all_freq:
11            all_freq[i] += 1
12        else:
13            all_freq[i] = 1
14

```

Run

Reset

No Output

✓ Correct

Good job!

1 / 1 point

8. For the vector  $v = [2.0, -3.5, 5.1]$ :

1. Find the L1 norm of  $v$
2. Return the result as a float

```

1 import numpy as np
2
3 def calculate_l1_norm(v):
4     ...
5     INPUT: LIST or ARRAY (containing numeric elements)
6     OUTPUT: FLOAT (L1 norm of v)
7     calculate and return a norm for a given vector
8     ...
9     l1_norm = np.linalg.norm(v, 1)
10    return l1_norm

```

Run

Reset

No Output

✓ Correct

Good job!

1 / 1 point

9. NumPy array practice

1. Create a vector that starts at 1 and increases until 150
2. Turn the vector into a matrix with 10 rows and 15 columns
3. Return the sums for the 10 rows as a **list**. HINT: there should be ten values for the printed sum

Use the following input vector values: `vectorLower = 1; vectorUpper = 151`

```

1 import numpy as np
2
3 def get_vector_sum(vectorLower, vectorUpper):
4     ...
5     INPUT: vector lower and upper bounds
6     OUTPUT: calculated value for vector sum
7     (1) create a vector ranging from 1:150
8     (2) transform the vector into a matrix with 10 rows and 15 columns
9     (3) print the sum for the 10 rows
10    ...
11    v=np.arange(vectorLower, vectorUpper)
12    s=v.reshape((10,15))
13    return np.sum(s,1)
14
15

```

Run

Reset

No Output

✓ Correct

Good job!

1 / 1 point

10. Which of the following pairs of events are **mutually exclusive**. There can be more than one answer.

- Odd numbers and the number 3
- Even numbers and numbers greater than 10
- Negative numbers and positive numbers less than 25

✓ Correct

Correct!

- Numbers between 100-200 and numbers between 201-300

 Correct

Correct!

None of the above

#### 11. Geometric distribution

1 / 1 point

The geometric distribution is a useful tool for modeling time to event data. A successful street vendor says that on average 1 out of every 10 people who walk by on the street stop to buy a taco.

1. Represent these data with a geometric distribution
2. What is the probability that the vendor *has to wait* until 20 people walk by before someone buys a taco?

```
1 import scipy.stats as stats
2
3 def geometric_distribution(p,k):
4     ...
5     INPUT: probability of success and trials
6     OUTPUT: determined probability
7     ...
8
9     distrib = stats.geom.pmf(k, p)
10    return distrib
```

Run

Reset

No Output

 Correct

Good job!

#### 12. Poisson distribution

0 / 1 point

The Poisson distribution is a useful tool for modeling count data given discrete intervals. Based on historical data the expected number of accidents at a busy intersection is 4 per month.

1. Represent these data with a Poisson distribution
2. What is the probability of more than 7 accidents at that intersection next month?

```
1 import scipy.stats as stats
2
3 def poisson_distribution(mu,k):
4     ...
5     INPUT: parameter of the poisson distribution and number of accidents
6     OUTPUT: determined probability
7     ...
8
9     distrib = stats.poisson.sf(k, mu)
10    total = 1 - distrib
11    return total
```

Run

Reset

No Output

 Incorrect

Try again!

#### 13. Gaussian distribution

1 / 1 point

The Gaussian or Normal distribution is used heavily throughout statistics and data science. Lets assume scores for this assessment have a mean of 50% and a standard deviation of 20%.

1. Represent these data with a Normal distribution
2. What is the probability of observing a score  $\geq 80$ ?

Use 50.0, 20.0, and 80 for your input values

```
1 import scipy.stats as stats
2
3 def gaussian_distribution(loc_val, scale_val, cdf_val):
4     ...
5     INPUT: loc (mean of the distribution), scale (standard deviation of the distribution), and cdf_val
6     OUTPUT: determined probability
7     ...
8
9     probability_gauss = stats.norm(loc_val, scale_val).cdf(cdf_val)
10    total = 1 - probability_gauss
11    return total
```

Run

Reset

No Output

 Correct

Good job!

14. Perform matrix multiplication on a square matrix HINT: A 2X2 matrix times a 2x2 matrix should yield a 2x2 matrix

1 / 1 point

```
1 def matrix_multiplication(A,B):
2     ...
3     INPUT: LIST (of length n) OF LIST (of length n) OF INTEGERS
4     LIST (of length n) OF LIST (of length n) OF INTEGERS
5     OUTPUT: LIST OF LIST OF INTEGERS
6     (storing the product of a matrix multiplication operation)
7     Return the matrix which is the product of matrix A and matrix B
8     where A and B will be (a) integer valued (b) square matrices
9     (c) of size n-by-n (d) encoded as lists of lists, e.g.
10    A = [[2, 3, 4], [6, 4, 2], [-1, 2, 0]] corresponds to the matrix
11    | 2 3 4 |
12    | 6 4 2 |
13    |-1 2 0 |
14    You may not use numpy. Write your solution in straight python
15    ...
16    result = [[sum(a*b for a, b in zip(A_row, B_col))
17               for B_col in zip(*B)]
18               for A_row in A]
19
20    return (result)
```

Run

Reset

No Output

✓ Correct

Good job!