CC 2.2.1: Introduction to NumPy Arrays

Introduction to NumPy Arrays: Question 1

1/1 point (graded)

True or False: a `numpy` array's length may be modified after being created.

True

False

correct

Introduction to NumPy Arrays: Question 2

1/1 point (graded)

Consider the following object:

```
numpy.array([0., 0., 0., 0., 0.])
```

What code will produce that object?

```
numpy.zeros(5)
```

correct

```
numpy.zeros(5, 0.)
```

```
numpy.zeros(0,0,0,0,0)
```

```
numpy.array(0,0,0,0,0)
```

Introduction to NumPy Arrays: Question 3

1/1 point (graded)

Consider the following code:

```
x = numpy.array([[3,6],[5,7]])
y = x.transpose()
print(y)
```

What does this print?

```
array([[5 7],[3 6]])
```

```
array([[3 5],[6 7]])
```
correct

```
array([[3 6],[5 7]])
```

```
array([[3 7],[5 6]])
```

This code contains an error.

CC 2.2.2: Slicing NumPy Arrays

Slicing NumPy Arrays: Question 1

0/1 point (graded)

Consider the following code:

```
x = np.array([1,2,5])
x[1:2]
```

What does this return?

```
array([1, 2])
```

```
array([2, 5])
```

```
array([1])
```

```
array([2])
```

correct

This code contains an error.

Explanation
Recall that indices start at 0 in Python. The end of a slice (i.e., the number to the right of the colon) is not included in the slice.

Slicing NumPy Arrays: Question 2

1/1 point (graded)

Consider the following code:

```
a = np.array([1,2])
b = np.array([3,4,5])
a + b
```

What does this return?

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

```
array([4, 6, 5])
```

```
array([4, 6])
```

This code contains an error.
correct

CC 2.2.3: Indexing NumPy Arrays

Indexing NumPy Arrays: Question 1

1/1 point (graded)
Consider the following code:

```
a = np.array([1,2])
b = np.array([3,4,5])
b[a]
```

What does this return?

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

```
array([4, 6, 5])
```

```
array([4, 5])
```
correct

This code contains an error.

Indexing NumPy Arrays: Question 2

1/1 point (graded)
Consider again the above code, as well as the following:

```
c = b[1:]
b[a] is c
```

What does Python return?

`True`

`False`
correct

This code contains an error.

Explanation
The `is` comparison operator tests if two objects are the same exact object --- not if they have the same exact values. When testing values, you could try `b[a] == c` or `all(b[a] == c)`.

CC 2.2.4: Building and Examining NumPy Arrays

Building and Examining NumPy Arrays: Question 1

1/1 point (graded)
Consider the following code:

```
x = 20
not np.any([x%i == 0 for i in range(2, x)])
```

What does the above code do?

Finds whether `x` is `0`.

Finds whether `x` contains `0`.

Finds whether `x` is even.

Finds whether `x` is prime.
correct

Explanation
`x%i == 0` tests if `x` has a remainder when divided by `i`. If this is not true for all values strictly between `1` and `x`, it must be prime!