

금공프3 퀴즈4

MFE 20249433 최재필

1.

Answer: 3

Why?

- Python's import system is hierarchical, which means that importing a top-level package does not automatically import its subpackages or modules.
- Given `__init__.py` is empty, importing just a top-level package `game` does not import its subpackages/modules.

2.

```
In [ ]: import numpy as np

abc = np.arange(5)
abc[0] = 10.2345
abc
```

```
Out[ ]: array([10,  1,  2,  3,  4])
```

```
In [ ]: abc.dtype
```

```
Out[ ]: dtype('int32')
```

Why?

- Because the array was originally created as an array of integers, its `dtype` is `int32`.

- To maintain the same type across all elements, numpy transformed the float value to int.

3.

```
In [ ]: np.random.seed(1)
rmat = np.random.randint(10, size=(3, 4))
rmat
```

```
Out[ ]: array([[5, 8, 9, 5],
               [0, 0, 1, 7],
               [6, 9, 2, 4]])
```

```
In [ ]: rmat2 = rmat[0]
rmat2
```

```
Out[ ]: array([5, 8, 9, 5])
```

Why?

- `rmat2` is of course one-dimensional because it's selecting the first (0th) element of `rmat`, which is basically an array of arrays.
- The first row is the first (0th) element of `rmat`, so naturally `rmat2` is one-dimensional instead of two.

4.

```
In [ ]: a = np.arange(15).reshape(3, 5)
a.shape
```

```
Out[ ]: (3, 5)
```

```
In [ ]: b = np.arange(3)
b.shape
```

```
Out[ ]: (3,)
```

```
In [ ]: np.hstack([a, b])
```

```

-----
ValueError                                Traceback (most recent call last)
Cell In[18], line 1
----> 1 np.hstack([a, b])

File f:\VSCodeProjects\KAIST_MFE\.venv\lib\site-packages\numpy\core\shape_base.py:359, in hstack(tup, dtype, casting)
    357     return _nx.concatenate(arrs, 0, dtype=dtype, casting=casting)
    358 else:
--> 359     return _nx.concatenate(arrs, 1, dtype=dtype, casting=casting)

ValueError: all the input arrays must have same number of dimensions, but the array at index 0 has 2 dimension(s) and the array at index 1 has 1 dimension(s)

```

Why?

- `a`'s shape is (3, 5) while `b` is (3,)
- To perform `hstack`, all dimensions should match except for the concatenation axis.
- Since `a`'s concatenation axis is 5 and `b` is None, they cannot be stacked.

5.

```
In [ ]: a = np.arange(15).reshape(3, 5)
a.shape
```

```
Out[ ]: (3, 5)
```

```
In [ ]: b = np.arange(3)
b.shape
```

```
Out[ ]: (3,)
```

```
In [ ]: b[None, :].shape # The same as b[np.newaxis, :]
```

```
Out[ ]: (1, 3)
```

Why?

- `+` is element-wise operation in Numpy. To perform element-wise operation, the arrays should be in the same shape or are broadcastable.
- To use broadcasting, `b` should be changed like the below.

```
In [ ]: a + b[:, None]
```

```
Out[ ]: array([[ 0,  1,  2,  3,  4],  
              [ 6,  7,  8,  9, 10],  
              [12, 13, 14, 15, 16]])
```