

Представьтесь: \_\_\_\_\_

## 1 Какие из следующих утверждений про массивы NumPy **неверны**?

- Сложение, умножение, деление и другие арифметические операции работают над массивами поэлементно.
- Все бинарные операции над массивами требуют явного приведения типов с помощью метода `np.array.astype`:

```
>>> x, y = np.array([1, 2]), np.array([3.0, 4.0])
>>> x + y
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'numpy.ndarray' and
'numpy.ndarray'
>>> x.astype(float) + y
array([ 4.,  6.])
```

- Специальное значение `np.newaxis` добавляет массиву размерность длины 1:

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

- Массивы NumPy, как и списки в Python, динамически изменяют свой размер при добавлении элемента:

```
>>> x = np.array([1, 2])
>>> x.append(3)
>>> x
array([1, 2, 3])
```

- Чтобы скопировать массив NumPy вместе с данными, достаточно воспользоваться срезом без аргументов:

```
>>> x = np.array([1, 2])
>>> y = x[:]
>>> y[0] = 42
>>> x
array([1, 2])
```

## 2 Что будет выведено на экран в строчках, отмеченных тремя знаками вопроса?

```
>>> x = np.array([[1, 2, 3], [4, 5, 6]])
>>> x.shape
???.
>>> x[:1, :1].shape
???.
>>> x[:, 0].shape
???.
>>> x[0].shape
???.
```

3 Перепишите фрагмент кода с использованием пакета NumPy.

```
import math
```

```
def f(xs, ys):  
    num = 0  
    sx, sy = 0, 0  
    for x, y in zip(xs, ys):  
        num += x * y  
        sx += x * x  
        sy += y * y  
    return num / math.sqrt(sx * sy)
```

4 Какое значение axis нужно использовать, чтобы получить приведенный вывод интерпретатора?

```
>>> x = np.array([[1, 2], [3, 4], [5, 6]])  
>>> x.min(axis=???)  
array([1, 3, 5])
```

5 Для каждой пары массивов укажите приводимы ли их размерности.

[1, 2, 3]	[[1, 2, 3]]
[[1], [2]]	[[1, 2]]
[1],	[1, 2]
[[1]]	[1, 2]