

Тест начат	Пятница, 6 октября 2023, 14:52
Состояние	Завершены
Завершен	Пятница, 6 октября 2023, 16:28
Прошло времени	1 ч. 35 мин.
Оценка	Еще не оценено

Вопрос **1**

Верно

Баллов: 1,00 из 1,00

Базовые операции с массивами NumPy (Часть 1)

У вас есть два одномерных массива:

- `array = np.arange(10) ** 4` — массив платежей по играм из сервиса;
- `array_2 = np.arange(10) ** 3` — массив платежей по подпискам в сервисе.

Выполните последовательно следующие задания применительно к исходным м

1. Определите общую сумму дохода сразу по двум потокам. Для этого сначала массивы, после чего получите сумму всех элементов. Результат сохраните в пе
2. Вычислите, насколько больше денег принесла продажа игр за указанный п в переменную `array_difference`.
3. Сохраните 2-й элемент массива по платежам игр в переменную `game_paym`
4. Сохраните последний элемент массива по платежам подписок в переменн

Обратите внимание: в заданиях **не требуется** выполнять вывод полученных р

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 import numpy as np
2
3 array = np.arange(10) ** 4
4 array_2 = np.arange(10) ** 3
5 # далее напишите ваше решение
6 array_sum = np.sum(np.add(array, array_2))
7 array_difference = np.subtract(np.sum(array), np.sum(array_2))
8 game_payments2 = array[1]
9 subscription_last = array_2[9]
```

	Тест	Ожидается	Получил
✓	<code>print('Решение задания 1.1')</code> <code>print(array_sum)</code>	Решение задания 1.1 17358	Решение задани 17358
✓	<code>print('Решение задания 1.2')</code> <code>print(array_difference)</code>	Решение задания 1.2 13308	Решение задани 13308
✓	<code>print('Решение задания 1.3')</code> <code>print(game_payments2)</code>	Решение задания 1.3 1	Решение задани 1
✓	<code>print('Решение задания 1.4')</code> <code>print(subscription_last)</code>	Решение задания 1.4 729	Решение задани 729

Все тесты пройдены! ✓

<

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **2**

Верно

Баллов: 1,00 из 1,00

Базовые операции с массивами NumPy (часть 2)

Ваш коллега создал структуру данных (см. окно ввода ответа) и передал вам в массив NumPy и выполните следующие задания:

1. Создайте подмассив lines, содержащий все строки со 2-й по 4-ю включительно.
2. Создайте подмассив last_column, содержащий последний столбец исходного массива.
3. Транспонируйте исходный массив и сохраните результат в переменную array_t.
4. Вычислите сумму элементов исходного массива, результат сохраните в переменную array_sum.
5. Вычислите среднее значение элементов исходного массива, результат сохраните в переменную array_avg.

Обратите внимание: в заданиях не требуется выполнять вывод полученных результатов. Для выполнения заданий 2.2–2.5 используйте функции [библиотеки NumPy](#).

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 import numpy as np
2
3 some_data = [
4     [3, 8, 1, 0, 1, 2],
5     [9, 2, 7, 3, 0, 4],
6     [2, 5, 1, 3, 1, 8],
7     [5, 1, 2, 1, 1, 0]
8 ]
9
10 # далее запишите ваш код
11 lines = np.array(some_data)[1:4,:]
12 last_column = np.array(some_data[:, 5])
13 array_t = np.array(some_data).T
14 array_sum = np.sum(np.array(some_data))
15 array_avg = np.mean(np.array(some_data))
16
```

	Тест	Ожидается	Получил
✓	print('Решение задания 2.1') print(lines) print()	Решение задания 2.1 [[9 2 7 3 0 4] [2 5 1 3 1 8] [5 1 2 1 1 0]]	Решение задания 2.1 [[9 2 7 3 0 4] [2 5 1 3 1 8] [5 1 2 1 1 0]]
✓	print('Решение задания 2.2') print(last_column)	Решение задания 2.2 [2 4 8 0]	Решение задания 2.2 [2 4 8 0]
✓	print('Решение задания 2.3') print(array_t)	Решение задания 2.3 [[3 9 2 5] [8 2 5 1] [1 7 1 2] [0 3 3 1] [1 0 1 1] [2 4 8 0]]	Решение задания 2.3 [[3 9 2 5] [8 2 5 1] [1 7 1 2] [0 3 3 1] [1 0 1 1] [2 4 8 0]]
✓	print('Решение задания 2.4') print(array_sum)	Решение задания 2.4 70	Решение задания 2.4 70
✓	print('Решение задания 2.5') print(array_avg)	Решение задания 2.5 2.9166666666666665	Решение задания 2.5 2.9166666666666665

Все тесты пройдены! ✓

<

Верно

Баллы за эту попытку: 1,00/1,00.

Вопрос **3**

Верно

Баллов: 1,00 из 1,00

Решение системы линейных уравнений

Из определенного листового материала необходимо выкроить 360 заготовок типа А, 300 заготовок типа Б и 675 заготовок типа В. При этом можно применять три способа раскроя.

Количество заготовок, получаемых из каждого листа при каждом способе раскроя, приведено в таблице.

Тип заготовки	Способ раскроя		
	1	2	3
А	3	2	1
Б	1	6	2
В	4	1	5

Найдите решение задачи, при котором расход материала и отходы окажутся минимальными.

Важно:

- Для решения используйте модуль linalg библиотеки NumPy.
- Ответом должен быть массив NumPy из чисел листов, необходимых на раскрой каждого типа заготовки третьим способом соответственно.

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 import numpy as np
2
3 # запишите здесь ваше решение
4 cutting_counts = np.array([[3, 2, 1],[1,6,2],[4,1,5]])
5 order_items = np.array([360,300,675])
6 answer = np.linalg.solve(cutting_counts, order_items)
7 print(answer)
8
```

	Тест	Ожидается	Получил	
✓	print()	[90. 15. 60.]	[90. 15. 60.]	✓
✓	print(answer[1] == 15)	True	True	✓
✓	print(answer[2] == 60)	True	True	✓

Все тесты пройдены! ✓

<

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **4**

Выполнен

Балл: 1,00

Анализ массива с помощью NumPy в Python

Это задание не проверяется LMS, но выполняя его, вы тренируете и закрепляете навыки работы с функциями библиотеки NumPy на различных исходных данных.

Для этого задания вы можете использовать любой доступный вам набор данных. Либо сгенерировать NumPy массив самостоятельно.

Что нужно сделать:

1. Создайте Python-ноутбук, загрузите в него данные из csv-файла и преобразуйте их в NumPy массив. Либо сгенерируйте массив самостоятельно.
2. Получите атрибуты массива — число измерений, размеры массива, количество элементов.
3. Потренируйтесь в выборке элемента по индексу и выполнении срезов данных.
4. Потренируйтесь в выполнении математических и статистических функций (например, найдите сумму элементов, среднее значение, медиану, отклонение от среднего и т. д.), которые считаете полезными при анализе исходного массива.

По каждому пункту задания сделайте выводы и отразите их в созданном Python-ноутбуке. Решение задания прикрепите ссылку на свой ноутбук.

```
def np_stat(a):
    print('=== math & stat data ', '='*10)
    print('np.array:', a)
    print('ndim:',a.ndim)
    print('shape:',a.shape)
    print('size:',a.size)
    print('dtype:',a.dtype)
    print('sum:',np.sum(a))
    print('mean:',np.mean(a))
    print('std:',np.std(a))
    print('var:',np.var(a))
    print('min:',np.min(a),' index: ',np.argmin(a))
    print('max:',np.max(a),' index: ',np.argmax(a))
    print('cumsum:',np.cumsum(a))
    print('cumprod:',np.cumprod(a))
```

Спасибо! Уверены, что вы хорошо постарались и поработали!

Вот некоторые возможности [библиотеки NumPy](#), которые будут вам полезны при работе с массивами:

1. Функции для создания массивов: `numpy.array()`, `numpy.zeros()`, `numpy.ones()`, `numpy.arange()`.
2. Атрибуты массивов: `numpy.ndarray.ndim`, `numpy.ndarray.shape`, `numpy.ndarray.size`, `numpy.ndarray.dtype`.
3. Индексация и срезы осуществляются по одному или нескольким индексам. Можно получить элемент из массива, так и получить подмассив из исходных данных.
4. Математические операции: `numpy.add()`, `numpy.subtract()`, `numpy.multiply()`, `numpy.divide()`.
5. Функции для вычисления статистических показателей массива: `numpy.mean()`, `numpy.average()`, `numpy.std()` и т. д.
6. Функции для генерации случайных чисел: `numpy.random.rand()`, `numpy.random.randn()`, `numpy.random.randint()` и т. д.

Если у вас возникнут вопросы по заданию, пожалуйста, обратитесь к преподавателю на ближайшем семинаре.

Желаем продуктивного обучения!