```python
# NumPy - Numerical Python

# - tablice wielowymiarowe (miacierze) [ndarray = N-Dimensional Array]
# - dużo złożonych funkcji matematycznych
# - operacje na duzych zbiorach danych
# - szybkość - zoptymalizowane operacje (oparte na językach niższego poziomu)
# - !tablice szybsze od list!


# 1. Tablica vs Lista

# 2. Ćwiczenia
import numpy as np

# Zadanie 1
# Stwórz funkcję print_array() w której zaimplementowana zostanie dwuwymiarowa
# tablica oraz wyświetlone zostaną:
# a)Tablica    [[-1,   2,  -3]
#               [ 4,   5,   6]
#               [ 7,   8,   9]]

# b)Pierwszy element tablicy
# c)Pierwszy zagnieżdżony element tablicy
# d)Typ utworzonego obiektu
# e)Kształt utworzonego obiektu
# Funkcja powinna zwracać utworzoną tablicę.
print("\n\n\n")

def print_array():
    tablica = np.array([[-1,2,-3], [4,5,6], [7,8,9]])
    print(f"Tablica:\n {tablica}")
    print(f"Pierwszy element:\n {tablica[0]}")
    print(f"Pierwszy zagnieżdżony: {tablica[0][0]}")
    print(f"Typ: {type(tablica)}")
    print(f"Kształt: {tablica.shape}")
    return tablica

arr = print_array()
```

```python
# ------------------------------------------------------------------------------
# Zadanie 2
# Stwórz funkcję shapeshifter która przyjmuje jako argument utworzoną wcześniej
# tablicę.
# Funkcja powinna:
# a)Zmienić rozmiar tablicy na 9x1
# b)Zmienić rozmiar tablicy na 1x9
# c)Zmienić rozmiar tablicy na 3x3
# d)Zmienić rozmiar tablicy na -1x9
# e)Zmienić rozmiar tablicy na 3x-1
# f)Podzielić tablicę na 3 nowe tablice

# '-1' : wstawiamy kiedy nie wiemy ile i komputer liczby sam
# (wiersze, kolumny)
# (ile_'list', ile_elementów_w_'liście')
print("\n\n\n")

def shapeshifter(arr):
    print(f"Rozmiar  9 x 1:\n {arr.reshape(9,1)}")
    print(f"Rozmiar  1 x 9:\n {arr.reshape(1,9)}")
    print(f"Rozmiar  3 x 3:\n {arr.reshape(3,3)}")
    print(f"Rozmiar -1 x 9:\n {arr.reshape(-1,9)}")
    print(f"Rozmiar  3 x-1:\n {arr.reshape(3,-1)}")
    new_arr = np.array_split(arr.reshape(-1,9), 3)
    print(f"Nowe:\ {new_arr}")

shapeshifter(arr)


# ------------------------------------------------------------------------------
# Zadanie 3
# Utwórz funkcję data_format, która zawierać będzie tablicę zawierającą dane
# różnych typów - wcześniej wspomniane zostało że tablice mogą przechowywać dane
# tylko 1 typu, dlatego należy obsłużyć ewentualny wyjątek. Dodatkowo utwórz tablice
# w których dane rzutowane będą kolejno na stringi, inty i floaty.

# dtype = ...
# | Kategoria | Zalecany `dtype` | Kod literowy |
Opis                                  |
# | --------- | ---------------- | ------------ | ---------------------------------------|
# | Integer   | `int64`          | ``i``, ``u`` | Liczby całkowite                       |
# | Float     | `float64`        | ``f``        | Liczby zmiennoprzecinkowe              |
# | Boolean   | `bool`           | ``?``        | Wartości logiczne                      |
# | Complex   | `complex128`     | ``c``        | Liczby zespolone                       |
# | String    | `str` / ``U``    | ``U``        | Napisy Unicode                         |
# | Daty      | `datetime64`     | ``M``        | Daty i czas                            |
# | Czas      | `timedelta64`    | ``m``        | Różnice czasu                          |
# | Obiekty   | `object`         | ``O``        | Dowolne obiekty Pythona                |

# [[1.1, 2.2, 3.3], [4.4, 5.5, 6.6], [7.7, 8.8, 9.9]] - do zmiany formatu
print("\n\n\n")
```

```python
def data_format():
    try:
        arr = np.array([[1.1,2.2,3.3], ["kot", 12, "ola"], ['a','b','5']], dtype="U")
        print(arr)
        print(type(arr[0][0]))
    except Exception as e:
        print(e)

    arr = np.array([[1.1, 2.2, 3.3], [4.4, 5.5, 6.6], [7.7, 8.8, 9.9]], dtype=str)
    print(arr)
    print(type(arr[0][0]))

    arr = np.array([[1.1, 2.2, 3.3], [4.4, 5.5, 6.6], [7.7, 8.8, 9.9]], dtype='int64')
    print(arr)
    print(type(arr[0][0]))

    arr = np.array([[1.1, 2.2, 3.3], [4.4, 5.5, 6.6], [7.7, 8.8, 9.9]], dtype='f')
    print(arr)
    print(type(arr[0][0]))

data_format()


# ------------------------------------------------------------------------------
# Zadanie 4
# Utwórz funkcję sortder_ndarray w której utworzona zostanie tablica, a następnie
# wyświetlona, oraz posortowana i ponownie wyświetlona.
print("\n\n\n")

def sorted_array():
    arr = np.array([[5,3,6], [3,7,8], [9,0,9]])
    print(f"Tablica przed sortowaniem:\n {arr}")
    print(f"Tablica posortowana:\n {np.sort(arr)}")

sorted_array()


# ------------------------------------------------------------------------------
# Zadanie 5
# Stwórz funkcję generate_random_numbers która wygeneruje 10 losowych liczb
# całkowitych od 0 do 100, oraz 10 losowych liczb typu float z zakresu od 0 do 1.
print("\n\n\n")

from numpy import random

def generate_random_numbers():
    for _ in range(10):
        print(random.randint(100))
    for _ in range(10):
        print(random.rand())

generate_random_numbers()
```

```python
# ------------------------------------------------------------------------------
# Zadanie 6
# Stwórz funkcję pick_random_numbers która:
# a) wygeneruje tablicę losowych liczb z zakresu od 0 do 1 o wymiarach 3x5.
# b) wybierze losową liczbę ze zbioru liczb 3,5,7,9
# c) utworzy tablicę o wymiarach 3x5 z losowych elementów ze zbioru liczb 3,5,7,9
# d) wygeneruj 100 losowych liczb, korzystając ze zbioru liczb 3,5,7,9,
# gdzie szansa na: 3 wynosi 10%, na 5-30 %, na 7-60%, a na 9-0%.
print("\n\n\n")

def pick_random_numbers():
    print(random.rand(3, 5))
    print(random.choice([3, 5, 7, 9]))
    print(random.choice([3, 5, 7, 9], size=(3, 5)))
    x = random.choice([3, 5, 7, 9], p=[0.1, 0.3, 0.6, 0.0], size=(100))
    print(x)

pick_random_numbers()


# ------------------------------------------------------------------------------
# Zadanie 7
# Stwórz funkcję shuffle_ndarray w której przemieszasz tablicę oraz utworzysz
# permutację tablicy.
print("\n\n\n")

def shuffle_ndarray():
    arr = np.array([1, 2, 3, 4, 5])
    print(f"Przed tasowaniem: {arr}")
    print(f"Przetasowana tablica: {random.shuffle(arr)}")
    print(f"Po tasowaniu: {arr}\n")

    arr = np.array([1, 2, 3, 4, 5])
    print(f"Przed tasowaniem: {arr}")
    print(f"Przetasowana tablica: {random.permutation(arr)}")
    print(f"Po tasowaniu: {arr}")

shuffle_ndarray()

# shuffle - tasuje oryginalną tablicę
# permutacja - tworzy nową pomieszaną tablicę z podanej
```

```python
# ------------------------------------------------------------------------------
# Zadanie 8
# Stwórz funkcję array_functions w której:
# a)Wygenerujesz tablicę z 11 wartościami równo rozmieszczonymi na przedziale od 0 do 10
# b)Wyświetlisz tą tablicę
# c)Wyświetlisz sumę elementów tej tablicy
# d)Wyświetlisz najmniejszy element tablicy
# e)Wyświetlisz największy element tablicy
# f) Wyświetlisz średnią wartość tablicy
# g)Wyświetlisz wariancje elementów tablicy (jak daleko, przeciętnie, elementy są od
średniej)
# (https://pogotowiestatystyczne.pl/slowniki/wariancja/)
# h)Wyświetlisz odchylenie standardowe elementów tablicy (typowa odległość od średniej)
# (https://pogotowiestatystyczne.pl/slowniki/odchylenie-standardowe/)
print("\n\n\n")

def array_functions():
    arr = np.linspace(0,10,11)
    print(f"Tablica: {arr}")
    print(f"Suma elementów: {np.sum(arr)}")
    print(f"Najmniejszy element: {np.min(arr)}")
    print(f"Najwiekszy element: {np.max(arr)}")
    print(f"Srednia elementów: {np.mean(arr)}")
    print(f"Wariancja elementów: {np.var(arr)}")
    print(f"Odchylenie standardowe elementów: {np.std(arr)}")

array_functions()


# ------------------------------------------------------------------------------
# Zadanie 9
# Utwórz funkcję mathematical_functions w której:
# a)Wyświetlisz wartość liczby pi
# b)Wyświetlisz wartość liczby eulera
# c)Wyświetlisz wartość kwadratu liczby eulera
# d)Wyświetlisz wartość sinus pi/2
# e)Wyświetlisz wartość cosinus pi/2
# f)Wyświetlisz wartość tangens pi/2
# g)Wyświetlisz wartość pi/2 zamienioną na stopnie i 360 stopni zamienioną na radiany.
print("\n\n\n")

def matematical_functions():
    print(f"Liczba pi: {np.pi}")
    print(f"Liczba Eulera: {np.e}")
    print(f"Liczba Eulera do potegi x: {np.exp(2)}")
    print(f"Sin pi/2: {np.sin(0.5*np.pi)}")
    print(f"Cos pi/2: {np.round(np.cos(0.5*np.pi))}")
    print(f"Tan pi/2: {np.round(np.tan(0.5*np.pi))}")
    print(f"Pi/2 to {np.degrees(0.5*np.pi)} stopni, a 360 stopni to {np.radians(360)}
radianów")

matematical_functions()
```