NumPy and Matplotlib I

1. Have a look at the first chapter of the scipy-lectures: `https://scipy-lectures.org/` (namely, '1. Getting started with Python for science')
2. Play around with the numpy functions: `arange`, `linspace`, `ones`, `zeros`, `eye`, `diag`, `tile`, `repeat` and `reshape`.
3. Generate the following matrices:
   a) Identity matrix
   b) Zero matrix
   c) A diagonal matrix $\left(D \in \mathbb{F}^{m \times n}$ is called diagonal, if $d_{ij} = 0 \; \forall i \neq j\right)$
   d) Example matrix
   $$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 \\ 1 & 6 & 1 & 1 \end{pmatrix}$$
   e) Example matrix
   $$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 6 \end{pmatrix}$$
   f) Example matrix
   $$\begin{pmatrix} 4 & 3 & 4 & 3 & 4 & 3 \\ 2 & 1 & 2 & 1 & 2 & 1 \\ 4 & 3 & 4 & 3 & 4 & 3 \\ 2 & 1 & 2 & 1 & 2 & 1 \end{pmatrix}$$
   g) Example matrix
   $$\begin{pmatrix} 1 & 6 & 11 \\ 2 & 7 & 12 \\ 3 & 8 & 13 \\ 4 & 9 & 14 \\ 5 & 10 & 15 \end{pmatrix}$$
4. Apply the functions `len()` and `numpy.shape()` to these arrays. How do the values relate? What does the attribute `ndim` tell us?
5. Use `matplotlib.pyplot.imshow()` to visualize the matrices.

Solution:

```
import numpy as np
import matplotlib.pyplot as plt

""" 4.1. """

""" 4.2."""

m, n = 10, 5
```

```
# help(np.arange)
print(np.arange(1, n, 2))
print(np.arange(n))

# help(np.linspace)
print(np.linspace(0, 1, n, endpoint=False))

# help(np.ones)
print(np.ones(n))
print(np.ones((n, n)))

# help(np.zeros)
print(np.zeros(n))
print(np.zeros((n, n)))

# help(np.eye)
print(np.eye(n))
print(np.eye(m, n))
print(np.eye(m, n, 3))

# help(np.diag)
print(np.diag(np.ones(n)))
print(np.diag(np.ones(n), 3))


# help(np.tile)
print(np.tile(np.arange(3), 2))
print(np.tile(np.arange(3), (2, 3)))
print(np.tile(np.ones((2, 2)), 3))

# help(np.repeat)
# Repeat elements of an array
print(np.repeat(np.arange(3), 2))
# print(np.repeat(np.arange(3), (2, 3)))
print(np.repeat(np.ones((2, 2)), 3))
print(np.repeat(np.ones((2, 2)), 3, axis=1))

# help(np.reshape)
x = np.eye(3)
print(x)
print(np.reshape(x, 9))

""" 4.3."""
# a) identity matrix

# b) zero matrix

# c) diagonal matrix
d = np.arange(n)
A = np.diag(d)
print(np.diag(d))
print(len(A), np.shape(A), A.ndim)
plt.figure()
plt.imshow(A)
plt.show()

# d)
A = np.ones((4, 4))
A[3, 1] = 6
A[2, 3] = 2
print(A)
```

```
print(len(A), np.shape(A), A.ndim)
plt.figure()
plt.imshow(A)
plt.show()

# e)
A = np.diag(np.arange(2, 7, 1), -1)
print(A)
print(len(A), np.shape(A), A.ndim)
plt.figure()
plt.imshow(A)
plt.show()

# f)
x = np.arange(4, 0, -1).reshape(2, 2)
A = np.tile(x, (2, 3))
print(A)
print(len(A), np.shape(A), A.ndim)
plt.figure()
plt.imshow(A)
plt.show()

# g)
x = np.arange(1, 16)
A = np.reshape(x, (5, 3), order='F')
print(A)
print(len(A), np.shape(A), A.ndim)
plt.figure()
plt.imshow(A)
plt.show()
```