

EXERCISE

CSV

Write a text file using `np.savetxt()` which contains 3 columns `x, y, z`:

```
x = 2, 4, 6, ..., 18  
y = 4, 8, 12, ..., 36  
z = 98, 95, 92, ..., 82
```

And try to read the file you just wrote using `np.loadtxt()`

HINT: `data = np.vstack((x, y, z)).T`

SOLUTION

```
filename = os.path.join(f_dir, 'exercise.csv')

x = np.arange(2, 20, 2)
y = x * 2
z = np.arange(98, 80, -2)

data = np.vstack((x, y, z)).T
np.savetxt(filename, data)

print(np.loadtxt(filename))
```

```
>>> >>> >>> >>> >>> >>> >>> >>> [[ 2.  4. 98.]
[ 4.  8. 96.]
[ 6. 12. 94.]
[ 8. 16. 92.]
[10. 20. 90.]
[12. 24. 88.]
[14. 28. 86.]
[16. 32. 84.]
[18. 36. 82.]]
```

NPZ

Use x, y, z again and save them using `np.savez()` and read them using `np.load()`

SOLUTION

```
filename = os.path.join(f_dir, 'exercise.npz')  
  
np.savez(filename, x=x, y=y, z=z)  
data = np.load(filename)  
print(data['x'], data['y'], data['z'])
```

```
>>> >>> >>> (array([ 2,  4,  6,  8, 10, 12, 14, 16, 18]), array([ 4,  
    8, 12, 16, 20, 24, 28, 32, 36]), array([98, 96, 94, 92, 90, 88, 86,  
    , 84, 82]))
```

BINARY

Write a binary file using `ndarray.tofile()` with 10 records, where a record looks like:

```
dtype = np.dtype([('id', np.float64),  
                  ('height', np.float32),  
                  ('weight', np.float32)])
```

And read it using `np.fromfile()`

SOLUTION

```
filename = os.path.join(f_dir, 'exercise.bin')

dtype = np.dtype([('id', np.float64),
                  ('height', np.float32),
                  ('weight', np.float32)])

records = np.zeros(10, dtype=dtype)
records.tofile(filename)

print(np.fromfile(filename, dtype=dtype))
```

```
>>> ... .. >>> >>> >>> >>> >>> [(0.0, 0.0, 0.0) (0.0, 0.0, 0.0) (0.
0, 0.0, 0.0) (0.0, 0.0, 0.0)
(0.0, 0.0, 0.0) (0.0, 0.0, 0.0) (0.0, 0.0, 0.0) (0.0, 0.0, 0.0)
(0.0, 0.0, 0.0) (0.0, 0.0, 0.0)]
```