

## Diff:

### Differences between given skeleton and solution

In order to make the sample solution easier to understand, the differences between it and the given skeleton source code were highlighted with the help of the program `diff`.

### Legend:

- Gray: unchanged text (only excerpts).
- Green: new lines
- Yellow: changed lines
- Red: deleted lines

Note: Files not listed have not been changed.

This document was created with the help of [diff2html](#) erstellt.

```
diff -u ../course08-performance-optimization/exercise/code/01_task1.py ../course08-performance-optimization/exercise/solution/01_task1.py
```

```
../course08-performance-optimization/exercise/code/01_task1.py          ../course08-performance-optimization/exercise/solution/01_task1.py
:                                                                        :
:                                                                        1
:                                                                        2
1 # adapted from source: http://numba.pydata.org/numba-doc/dev/user/examples.html 3 # adapted from source: http://numba.pydata.org/numba-doc/dev/user/examples.html
2                                                                        4
3 from matplotlib.pyplot import imshow, show, cm, savefig                    5 from matplotlib.pyplot import imshow, show, cm, savefig
:                                                                        6 import matplotlib.pyplot as plt
4 import numpy as np                                                        7 import numpy as np
5                                                                        8
:                                                                        9 import time
:                                                                        10
6                                                                        11
7 def mandel(x, y, max_iters):                                              12 def mandel(x, y, max_iters):
8     """                                                                    13     """
:                                                                        :
:                                                                        :
36                                                                           41
37     return image                                                         42     return image
38                                                                           43
39 resx = 500                                                                44 dt_results = []
40 resy = 500                                                                45 for r in range(1, 6):
41                                                                           46     res = r*100
42 image = np.zeros((resx, resy), dtype=np.uint8)                          47     image = np.zeros((res, res), dtype=np.uint8)
43                                                                           48
44 xmin, xmax, ymin, ymax = -2.0, 1.0, -1.0, 1.0                          49     xmin, xmax, ymin, ymax = -2.0, 1.0, -1.0, 1.0
45                                                                           50
46 create_fractal(xmin, xmax, ymin, ymax, image, 255)                       51     t0 = time.time()
:                                                                           52     create_fractal(xmin, xmax, ymin, ymax, image, 255)
:                                                                           53     dt = time.time()-t0
:                                                                           54     print("res = {}; Time needed: {}".format(res, dt))
:                                                                           55     dt_results.append((res, dt))
:                                                                           56
:                                                                           57     # Time to create the image is not included in the calculation
:                                                                           58     plt.figure()
:                                                                           59     # set special colormap
:                                                                           60     imshow(image, extent=(xmin, xmax, ymin, ymax), cmap=cm.plasma)
:                                                                           61
62 # conversion from [(1, a), (2, b), (3, c), ...] to [[1, 2, 3, ...], [a, b, c, ...]]
63 # siehe https://docs.python.org/3/library/functions.html#zip
64 res_list, dt_list = zip(*dt_results)
65 plt.figure()
66 plt.plot(res_list, dt_list, 'b.-')
67 plt.xlabel("resolution")
68 plt.ylabel("computation time")
47                                                                           69
48 # set special colormap
49 imshow(image, extent=(xmin, xmax, ymin, ymax), cmap=cm.plasma)
50 show()                                                                    70 show()
```

```
Nur in ../course08-performance-optimization/exercise/solution/: 02_task2.py.
Nur in ../course08-performance-optimization/exercise/solution/: build.
Nur in ../course08-performance-optimization/exercise/solution/: .directory.
Nur in ../course08-performance-optimization/exercise/solution/: .gitignore.
Nur in ../course08-performance-optimization/exercise/solution/: mandelcy.cpython-38-x86_64-linux-gnu.so.
Nur in ../course08-performance-optimization/exercise/solution/: mandel-cython.c.
Nur in ../course08-performance-optimization/exercise/solution/: mandel-cython-main.py.
Nur in ../course08-performance-optimization/exercise/solution/: mandel-cython.png.
Nur in ../course08-performance-optimization/exercise/solution/: mandel-cython.pyx.
```

```
Nur in ../course08-performance-optimization/exercise/solution/: mandel-cython-setup.py.
```