In [1]:

```
import numpy as np
PI2 = 2 * np.pi
```

Exercise01

In [2]:

```
ys = [-0.5, 0.1, 0.7, -0.1]
hs = np.fft.fft(ys)
print(hs)
```

```
[ 0.2+0.j  -1.2-0.2j  0.2+0.j  -1.2+0.2j]
```

In [3]:

```
def dft(ys):
    N = len(ys)
    ts = np.arange(N) / N
    freqs = np.arange(N)
    args = np.outer(ts, freqs)
    M = np.exp(1j * PI2 * args)
    amps = M.conj().transpose().dot(ys)
    return amps
```

In [4]:

```
hs2 = dft(ys)
np.sum(np.abs(hs - hs2))
```

Out[4]:

```
5.864775846765962e-16
```

In [5]:

```
def fft_norec(ys):
    N = len(ys)
    He = np.fft.fft(ys[::2])
    Ho = np.fft.fft(ys[1::2])

    ns = np.arange(N)
    W = np.exp(-1j * PI2 * ns / N)

    return np.tile(He, 2) + W * np.tile(Ho, 2)
```

In [6]:

```
hs3 = fft_norec(ys)
np.sum(np.abs(hs - hs3))
```

Out[6]:

```
0.0
```

In [7]:

```
def fft(ys):
    N = len(ys)
    if N == 1:
        return ys

    He = fft(ys[::2])
    Ho = fft(ys[1::2])

    ns = np.arange(N)
    W = np.exp(-1j * PI2 * ns / N)

    return np.tile(He, 2) + W * np.tile(Ho, 2)
```

In [8]:

```
hs4 = fft(ys)
np.sum(np.abs(hs - hs4))
```

Out[8]:

```
1.6653345369377348e-16
```

In [ ]:

```

```