

```
import numpy as np
```

```
# Consideriamo un array generico
```

```
A = np.random.randint(6,40,(2,4))
```

```
A
```

```
array([[16, 21, 33, 27],  
       [15, 17, 38, 18]])
```

```
# Somma per 2. Ogni elemento dell'array sarà sommato per 2
```

```
A+2
```

```
array([[18, 23, 35, 29],  
       [17, 19, 40, 20]])
```

```
# Ai nostri fini, consideriamo due array unidimensionali. Per una  
questione pratica, estraiamo le righe del  
# nostro array A. La prima riga sarà il prezzo, la seconda la quantità
```

```
prezzo = A[0,:]
```

```
prezzo
```

```
array([16, 21, 33, 27])
```

```
quantita = A[1,:]
```

```
quantita
```

```
array([15, 17, 38, 18])
```

```
# Il profitto sarà prezzo per quantita. Si noti che il prodotto può  
essere calcolato solo tra array dello stesso elemento,  
# e tale prodotto sarà effettuato elemento per elemento.
```

```
profitto = prezzo * quantita
```

```
profitto
```

```
array([ 240,  357, 1254,  486])
```

```
# Riprendiamo il nostro array quantita, ed effettuiamo altre operazioni su di esso.
```

```
quantita - 24
```

```
array([-9, -7, 14, -6])
```

```
quantita / 2
```

```
array([ 7.5,  8.5, 19. ,  9. ])
```

```
# Per una questione didattica, creiamo un array unidimensionale casuale utilizzando la funzione default_rng avente 6 elementi compresi tra 0 e 1
```

```
rng = np.random.default_rng(616)
```

```
quantita = rng.random(7)
```

```
quantita
```

```
array([0.39682145, 0.86568572, 0.46040359, 0.30599848, 0.57381588,  
       0.08888468, 0.88194347])
```

```
# Rendiamo questo array più leggibile. Moltiplichiamo per 10 gli elementi e mettiamo solo due elementi decimali
```

```
quantita = (quantita*10).round(2)
```

```
quantita
```

```
array([39.7, 86.6, 46. , 30.6, 57.4,  8.9, 88.2])
```

```
# Ripetiamo quindi i calcoli visti in precedenza
```

```
quantita - 24
```

```
array([ 15.7,  62.6,  22. ,   6.6,  33.4, -15.1,  64.2])
```

```
quantita / 2
```

```
array([19.85, 43.3 , 23. ,  15.3 , 28.7 ,   4.45, 44.1 ])
```

