

```
import numpy as np
```

Δημιουργία 1D πίνακα

```
a = np.array([1, 2, 3, 4, 5])
```

Δημιουργία 2D πίνακα

```
b = np.array([[1, 2, 3], [4, 5, 6]])
```

Ιδιότητες των Arrays

```
print(a.shape) # Διάσταση του πίνακα (π.χ. (5,))
```

```
print(b.ndim) # Αριθμός διαστάσεων (π.χ. 2)
```

```
print(a.size) # Συνολικός αριθμός στοιχείων
```

```
print(b.dtype) # Τύπος δεδομένων (π.χ. int64)
```

Συναρτήσεις και πράξεις με Arrays

```
# Άθροισμα στοιχείων του πίνακα
```

```
print(np.sum(a))
```

Μέση τιμή

```
print(np.mean(a))
```

Πράξεις στοιχείο προς στοιχείο

```
c = a + 5 # Προσθήκη 5 σε κάθε στοιχείο
```

```
d = a * 2 # Πολλαπλασιασμός με 2
```

Γινόμενο πινάκων (dot product)

```
e = np.dot(a[:3], b[0])
```

Δημιουργία Ειδικών Πινάκων

```
# Πίνακας με μηδενικά
```

```
zeros = np.zeros((3, 3))
```

```
# Πίνακας με μονάδες
```

```
ones = np.ones((2, 2))
```

```
# Τυχαίοι αριθμοί
```

```
random = np.random.rand(3, 3)
```

Φιλτράρισμα και Slicing

```
# Επιλογή του πρώτου στοιχείου
```

```
print(a[0])
```

```
# Επιλογή μιας στήλης από 2D πίνακα
```

```
print(b[:, 1])
```

```
# Φιλτράρισμα με βάση συνθήκες
```

```
filtered = a[a > 2]
```

```
print(filtered) # Θα εμφανίσει: [3 4 5]
```

Ανασχηματισμός (Reshaping) Πινάκων

```
a = np.array([1, 2, 3, 4, 5, 6])
reshaped_a = a.reshape(2, 3) # Μετατροπή σε 2x3 πίνακα
print(reshaped_a)
```

Συναρτήσεις Aggregation

```
# Εύρεση του ελάχιστου και μέγιστου
print(np.min(a)) # Ελάχιστη τιμή
print(np.max(a)) # Μέγιστη τιμή
```

Αθροίσματα κατά άξονα

```
print(np.sum(reshaped_a, axis=0)) # Άθροισμα στήλης
print(np.sum(reshaped_a, axis=1)) # Άθροισμα γραμμής
```

Τυπική απόκλιση και διακύμανση

```
print(np.std(a)) # Τυπική απόκλιση
print(np.var(a)) # Διακύμανση
```

Στοιχίση (Stacking) και Διαχωρισμός (Splitting) Πινάκων

```
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
```

Κάθετη στοιχίση (στοιχείο προς στοιχείο προσθέτει γραμμές)

```
stacked = np.vstack((a, b))
print(stacked)

# [[1 2 3]
```

```
# [4 5 6]]
```

```
# Οριζόντια στοίχιση (στοιχείο προς στοιχείο προσθέτει στήλες)
```

```
stacked_h = np.hstack((a, b))
```

```
print(stacked_h)
```

```
# [1 2 3 4 5 6]
```

```
Διαχωρισμός (Splitting)
```

```
a = np.array([1, 2, 3, 4, 5, 6])
```

```
split_a = np.split(a, 3)
```

```
print(split_a)
```

```
# [array([1, 2]), array([3, 4]), array([5, 6])]
```

```
Πολυδιάστατοι Πίνακες και Άξονες
```

```
a = np.array([[1, 2, 3], [4, 5, 6]])
```

```
# Άθροισμα κατά γραμμή (axis=1)
```

```
print(np.sum(a, axis=1)) # [6 15]
```

```
# Άθροισμα κατά στήλη (axis=0)
```

```
print(np.sum(a, axis=0)) # [5 7 9]
```

```
Συναρτήσεις για Τυχαίους Αριθμούς
```

```
# Τυχαίοι αριθμοί από κανονική κατανομή
```

```
rand_normal = np.random.randn(3, 3)
```

```
# Τυχαίοι ακέραιοι
```

```
rand_integers = np.random.randint(1, 10, size=(2, 3))
```

Εισαγωγική Άσκηση NumPy

Οδηγίες

Δημιουργία Πίνακα:

Δημιούργησε έναν πίνακα NumPy 1 διάστασης που περιέχει τους αριθμούς από το 1 έως το 10.

Reshape Πίνακα:

Μετασχημάτισε τον παραπάνω πίνακα σε πίνακα 2 διαστάσεων με 2 γραμμές και 5 στήλες.

Συναρτήσεις Συγκέντρωσης:

Βρες το άθροισμα, τη μέση τιμή, την ελάχιστη και τη μέγιστη τιμή του πίνακα.

Πράξεις με Πίνακες:

Δημιούργησε έναν νέο πίνακα 2x5 με τυχαίους ακέραιους αριθμούς από το 1 έως το 10.

Πρόσθεσε τον νέο πίνακα στον αρχικό που δημιούργησες και βρες το αποτέλεσμα.

Φιλτράρισμα Δεδομένων:

Φιλτράρισε τον αρχικό πίνακα ώστε να κρατήσεις μόνο τους αριθμούς που είναι μεγαλύτεροι από 5.

Στοιχίση Πινάκων:

Στοιχίσε τον αρχικό πίνακα με τον νέο πίνακα που δημιούργησες, ώστε να προκύψει ένας νέος πίνακας με διαστάσεις 4x5.