

Tuples

Ορισμός ενός Tuple

- Τα **tuples** είναι ένας ειδικός τύπος **list**
- **Δεν** μπορούμε να αλλάξουμε τα στοιχεία
- Ένα **tuple** ξεχωρίζει τα στοιχεία του με κόμμα μέσα σε ένα ζευγάρι παρενθέσεων
- Μπορούμε να έχουμε διαφορετικούς τύπους δεδομένων μέσα σε ένα **tuple**

Παραδείγματα δημιουργίας tuple

- `tuple1 = (1, 2, 3, 4, 5, 6)`
- `tuple2 = ('python', 'student', 20, 1.5)`
- `tuple3 = "1", "2", "3", "4", "5"`
- `tuple4 = ()`
- `tuple5 = ("single value",)`

Πρόσβαση στα στοιχεία ενός Tuple

```
MyTuple = (35, 30, 100, 200, 30, 2048)
```

```
print('element 2 in tuple is: ', MyTuple[1])
```

- Μπορούμε να εμφανίσουμε και ένα πλήθος στοιχείων με τον εξής τρόπο

```
print('elements : ' + MyTuple[1:4])
```

- Ποιά στοιχεία θα εμφανιστούν;

Ενημέρωση των tuples

- Μπορεί να μην μπορούμε να αλλάξουμε τα στοιχεία ενός **tuple** αλλά μπορούμε να προσθέσουμε 2 **tuples** και το αποτέλεσμα να είναι ένα τρίτο που θα περιέχει τα στοιχεία αυτών που προσθέσαμε

```
tup1 = (23, 46, 92)
```

```
tup2 = (25, 50, 100)
```

```
tup3 = tup1 + tup2
```

```
# tup3 = (23, 46, 92, 25, 50, 100)
```


Διαγραφή στοιχείων ενός tuple

- Η διαγραφή ενός στοιχείου είναι αδύνατη όταν έχουμε να χειριστούμε **tuples**
- Αυτό που μπορούμε να κάνουμε ωστόσο είναι να διαγράψουμε ολόκληρο το tuple με την δεσμευμένη λέξη της Python **del**

```
myTuple = (1, 2, 3)
```

```
del myTuple
```

Βασικές Πράξεις των Tuples

- **len**((1, 2, 3)) # μέγεθος
- (1, 2) + (3, 4) # σύμπτυξη
- ('γεια',) * 2
- 4 **in** (1, 2, 3, 4) # έλεγχος μέλους
- χρήση μαζί με την **for**
for x in (1, 2, 3): **print**(x)

Συναρτήσεις για Tuples

len(tuple) επιστροφή του μεγέθους του

max(tup) , **min**(tup) επιστροφή του
μεγαλύτερου και μικρότερου στοιχείου
αντίστοιχα

tuple(sequence) μετατροπή μιας λίστας σε
ένα **tuple**