

---

# Λίστες στην ML

# Λίστες

---

- Ο πιο απλός τρόπος δόμησης δεδομένων.
- Συλλογές σταθερών τιμών, εφόσον είναι του **ίδιου τύπου**.
- Επιτρέπονται πολλαπλές εμφανίσεις στοιχείου (όπως και στον LP, οι λίστες δεν είναι σύνολα).

# Δημιουργία Λίστας: ορισμός του τύπου της

---

- Ορισμός του τύπου της
  - Νέος μοναδιαίος τελεστής τύπων **list**, που εφαρμόζεται σε postfix μορφή.
- Παραδείγματα

**int list** λίστα ακεραίων  
**(bool x string) list**  
**(int list) list**  
**(int → string) list**  
**int → (string list)**  
**α list**

---

*T is a type*

*T list is a type*

# Δημιουργία Λίστας: ορισμός των μελών της

---

- Η κενή λίστα (που δεν περιέχει κανένα στοιχείο): [ ]
- Η ένταξη μελών στη λίστα μέσω της συνάρτησης κατασκευής λιστών cons ::
- **Τύπος cons** :: : a x a list  $\rightarrow$  a list
- Εφαρμογή cons:  $5 :: [6, 7, 8] = [5, 6, 7, 8]$  (infix)  
(προσθέτει στοιχείο στην κεφαλή της λίστας, αρκεί το στοιχείο να είναι του τύπου της λίστας)
- Οποιαδήποτε λίστα μπορεί να κατασκευαστεί από την [ ], με διαδοχικές εφαρμογές του cons, για κάθε στοιχείο που εντάσσεται στη λίστα.

# Παράδειγμα: κατασκευή της [5, 6, 3, 2, 8]

---

$$8 :: [] = [8]$$

$$2 :: [8] = [2, 8]$$

$$3 :: [2, 8] = [3, 2, 8]$$

$$6 :: [3, 2, 8] = [6, 3, 2, 8]$$

$$5 :: [6, 3, 2, 8] = [5, 6, 3, 2, 8]$$

- Ισοδύναμα, μετεγγράφοντας/αντικαθιστώντας:

$$[5, 6, 3, 2, 8] = 5 :: [6, 3, 2, 8] = 5 :: 6 :: [3, 2, 8] =$$

$$5 :: 6 :: 3 :: [2, 8] = 5 :: 6 :: 3 :: 2 :: [8] = 5 :: 6 :: 3 :: 2 :: 8 :: []$$

# Πράξεις σε Λίστες

---

- **Συνένωση** λιστών, εφόσον έχουν **ίδιο τύπο**

$@ : a \text{ list} \times a \text{ list} \rightarrow a \text{ list}$

$[a_1, \dots, a_n] @ [b_1, \dots, b_m] = [a_1, \dots, a_n, b_1, \dots, b_m]$

```
fun    [] @ x = x
      |   x @ [] = x
      | (x1 :: xs) @ y = x1 :: (xs @ y)
```

- Είναι προσεταιριστική δηλ  $(x @ y) @ z = x @ (y @ z)$
- Δεν είναι αντιμεταθετική!

---

- **Αντιστροφή** λίστας

$rev : \alpha \text{ list} \rightarrow \alpha \text{ list}$

$rev [a_1, a_2, \dots, a_n] = [a_n, \dots, a_2, a_1]$

**fun**  $rev [] = []$

|  $rev x :: xs = rev xs @ [x]$

---

- **Μήκος** λίστας

**length** :  $\alpha$  list  $\rightarrow$  int

**length** [a1, ..., an] = n

**fun** length [ ] = 0

| length x :: xs = 1 + length xs



- 
- **Διαγραφή** στοιχείου από λίστα (μια φορά)

$\text{del\_elem} : \alpha \times \alpha \text{ list} \rightarrow \alpha \text{ list}$

$\text{del\_elem } a_k [a_1, \dots, a_n] = [a_1, \dots, a_{(k-1)}, a_{(k+1)}, \dots, a_n]$

**fun** del\_elem x [x] = []

| del\_elem x x :: xs = xs

| del\_elem x y :: ys = y :: del\_elem x ys

# Ασκήσεις

---

- Ορίστε
  - Εύρεση στοιχείου δεδομένης θέσης στη λίστα  
 $\text{rtn\_elem} : \text{int} \times \text{a list} \rightarrow \text{a}$
  - Εισαγωγή στοιχείου σε δεδομένη θέση της λίστας  
 $\text{ins\_pos} : \text{a} \times \text{int} \times \text{a list} \rightarrow \text{a list}$
  - Διαγραφή στοιχείου από δεδομένη θέση της λίστας  
 $\text{del\_pos} : \text{a} \times \text{int} \times \text{a list} \rightarrow \text{a list}$
  - Διάταξη στοιχείων λίστας ακεραίων (με όποιον αλγόριθμο διάταξης θέλετε)  
 $\text{sort} : \text{a list} \rightarrow \text{a list}$