

# Corso ITS: *ARTIFICIAL INTELLIGENCE SPECIALIST*

## Modulo: Programmazione Procedurale in Python

Docente: *Andrea Ribuoli*

---

### Martedì 28 Gennaio 2025

### 09:00 - 14:00

---

- per memorizzare raccolte di valori Python usa le **liste**
- trattandosi di contenitori verranno scandite tramite `for`
- supponiamo di leggere (con un ciclo) una serie di numeri
- vogliamo stamparli evidenziando il **valore massimo**
- non possiamo farlo prima di aver acquisito l'intero elenco
- una lista si inizializza con le **parentesi quadre** `[]`
- i valori iniziali sono separati da virgole
- accedo agli elementi con l'**indice** `( lista[indice] )`
- per accedere tramite indice la posizione **deve già esistere**
- `esempio = []` crea un **lista vuota**
- ha senso? Sì, perchè posso **aggiungere** elementi
- `esempio.append(valore)` metodo **append()**
- il metodo `append()` accoda alla fine della lista
- col metodo **insert(posizione, valore)** si aggiunge ovunque
- gli elementi successivi vengono spostati
- non posso indicare come posizione un indice fuori dal range
- `if 30 in esempio` : è vero se uno dei valori presenti è 30
- `esempio.index(30)` ci dice la prima posizione in cui è presente

```
In [1]: esempio = []
esempio.append(24)
esempio.append(27)
esempio.append(30)
esempio.append(22)
```

```

esempio.append(28)
esempio.append(30)
n = esempio.index(30)
n2 = esempio.index(30, n + 1)
print("Ho preso 30 negli esami", n, "e", n2)

```

Ho preso 30 negli esami 2 e 5

- se non presente la *index* genera una eccezione
- nell'incertezza devo verificare con `if 30 in esempio :`
- è possibile eliminare (estrarre) elementi con `pop(indice)`
- i successivi "*slittano*"
- se ometto l'indice, `pop()` , estraggo l'ultimo elemento
- il metodo `remove(valore)` elimina in base al valore
- il valore deve esistere

```
In [2]: a = esempio.remove(30)
```

```
In [3]: a = esempio.remove(30)
```

```
In [4]: a = esempio.remove(30)
```

```

-----
ValueError                                Traceback (most recent call last)
Cell In[4], line 1
----> 1 a = esempio.remove(30)

ValueError: list.remove(x): x not in list

```

- il confronto tra due liste considera l'ordine dei valori

Prima di rimuovere i due 30:

```
In [15]: print(sum(esempio)/len(esempio))
```

26.833333333333332

dopo:

```
In [13]: print(sum(esempio)/len(esempio))
```

25.25

```
In [16]: print(max(esempio), min(esempio))
```

30 22

- il metodo **list()** crea "*veramente*" una nuova lista
- l'operatore **porzione**(*slice*) si indica coi **due punti**( `:` )

- esempio[a : b]
- **a**: indice elemento da includere
- **b**: indice elemento da escludere
- **entrambi facoltativi**
- esempio[ : b] : tutti fino alla posizione **b esclusa**
- esempio[a : ] : tutti dalla posizione **a inclusa**

```
valori = []
print('Inserisci i voti degli esami separati da Invio')
print('Termina con Q:')
inputUtente = input("")
while inputUtente.upper() != "Q" :
    valori.append(int(inputUtente))
    inputUtente = input("")
```

- esiste un metodo per l'ordinamento: **sort()**
- in Python non esistono **tabelle**
- possiamo però creare **liste di liste**

```
In [17]: scacchiera = []
RIGHE   = 8
COLONNE = 8

for i in range(RIGHE) :
    riga = [" "] * COLONNE
    scacchiera.append(riga)
```