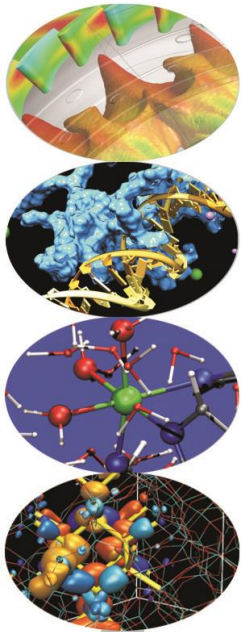
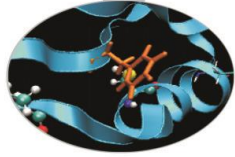


# Esercitazione

## Dati Strutturati



# Esercitazione



## •Esercizio 0 (Liste):

Scrivere un programma che generi tutti i numeri pari e dispari tra 1 e N. Dove N viene inserito da tastiera. Creare due liste `odd` e `even` per salvare i dati.

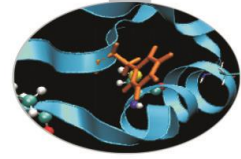
Usare un ciclo `for` e la funzione `append`.

Riscrivere il codice usando la funzione `range` per creare le liste. (Solution: `odd_even.py`)

## •Esercizio 1(Liste):

Provare ad eseguire il seguente programma che elimina tutti i numeri  $>2$  dalla lista data in input:

```
print "plain remove in a for loop:"
lista = [3,4,2,1]
for item in lista:
    print "visiting item %s in list %s" % (item, lista)
    if item > 2:
        lista.remove(item)
```

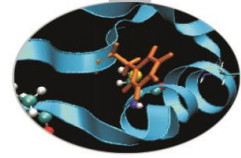


# Esercitazione

L'algoritmo torna i risultati corretti??perchè?

Implementare una versione corretta dell'algoritmo:

- 1- Usando una lista di copia.
- 2- Usando un ciclo `while` ed un'opportuno `index` da aggiornare correttamente ad ogni passo del loop



# Esercitazione

## Esercizio2 (lista – ciclo for):

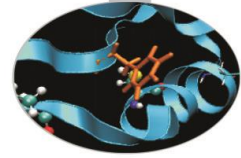
- Costruire una discretizzazione dell'intervallo  $[a,b]$  con passo uniforme  $h$ .  
Immagazzinare le coordinate in una lista  $L$  inizialmente vuota.
- Costruire una rappresentazione discreta della funzione  $\sin(x)$  nei valori nodali definiti sull'intervallo discretizzato  $[0,\pi,passo]$ ;  $passo=0.1$ .
- Calcolare un'approssimazione della derivata della funzione  $\sin(x)$  tramite un rapporto incrementale.

$$f'(x_0) := \lim_{h \rightarrow 0} \frac{\Delta f}{\Delta x}(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

- Calcolare un'approssimazione dell'integrale della funzione  $\sin(x)$  sull'intervallo  $(0,\pi)$  tramite la regola dei trapezi generalizzata:

$$\int_0^{\pi} \sin(x) = \frac{h}{2} \left[ s_0 + 2 \sum_{k=1}^{n-1} s_k + s_n \right]$$

([Soluzione: derivata.py](#))



# Esercitazione

## Esercizio2 (lista – ciclo for):

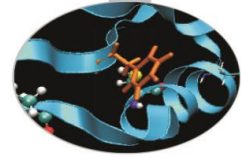
- Costruire una discretizzazione dell'intervallo  $[a,b]$  con passo uniforme  $h$ . Immagazzinare le coordinate in una lista  $L$  inizialmente vuota.
- Costruire una rappresentazione discreta della funzione  $\sin(x)$  nei valori nodali definiti sull'intervallo discretizzato  $[0,\pi, \text{passo}]$ ;  $\text{passo}=0.1$ .
- Calcolare un'approssimazione della derivata della funzione  $\sin(x)$  tramite un rapporto incrementale.

$$f'(x_0) := \lim_{h \rightarrow 0} \frac{\Delta f}{\Delta x}(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

- Calcolare un'approssimazione dell'integrale della funzione  $\sin(x)$  sull'intervallo  $(0,\pi)$  tramite la regola dei trapezi generalizzata:

$$\int_0^{\pi} \sin(x) = \frac{h}{2} \left[ s_0 + 2 \sum_{k=1}^{n-1} s_k + s_n \right]$$

([Soluzione: derivata.py](#))



# Esercitazione

## Esercizio 0 (Dictionary)

Scrivere un programma che presa una stringa calcoli la frequenza di ciascun carattere. Costruire un dictionary: carattere-frequenza. ([Solution](#) frequency.py)

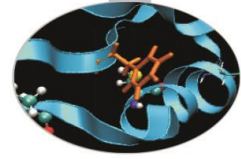
## Esercizio 1 (Dictionary)

Creare un dizionario `d={'uno':1, 'tre':3, 'due':2, 'quattro':4}`

Stampare le chiavi in ordine alfabetico.

Stampare le coppie chiavi valore in ordine alfabetico rispetto alla chiave

Stampare le coppie chiavi valore in ordine alfabetico rispetto al valore (usare la funzione `sorted`) ([Solution](#): sorted.py)



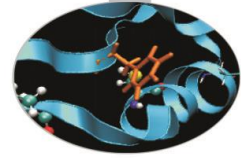
# Esercitazione

## Esercizio 2 (Dictionary):

- Un polinomio può essere rappresentato opportunamente tramite un dizionario. Per esempio il polinomio  $5x^4+2x^2+3x+4$  può essere rappresentato tramite il dizionario  $\{4:5, 2:2, 1:3, 0:4\}$  dove la key corrisponde al grado e il value al coefficiente.
- Scrivere un programma che facendo uso dei dizionari calcoli il valore della derivata di un dato polinomio :  $5x^4+2x^2+3x+4$  nel punto  $x=4$ .

• Si ricordi che

$$\frac{d}{dx} \sum x^j c_j = \sum j c_j x^{j-1} \quad (\text{Solution diff\_pol.py})$$



# Esercitazione

Esercizio 3: (Exception Handling - Costrutti – Dizionari):

- Implementare il gioco della Morra Cinese. Assumere di :
- Leggere l'input da tastiera e gestire le eccezioni: input mancante o diverso dai valori 'Rock'- 'Scissor'- 'Paper'
- Utilizzare la funzione `randint(self,a,b)` per generare un numero casuale tra a e b.  
(Solution: Scissor-Paper-Rock.py)

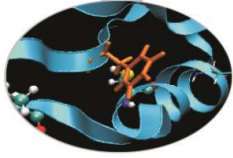
Esercizio 4 (Dictionary):

Scrivere un programma per il login di un utente ad un sistema. Gli utenti vengono salvati tramite un dictionary nome-password. Il dictionary iniziale è così formato

```
db={'Admin':'Admin'}
```

Il menù deve prevedere le seguente opzioni:





# Esercitazione

- (N)ew User
- (E)xisting User Login
- (A)dministrator
- (Q)uit

Il programma termina quando viene inserito da terminale il comando Q.

L'opzione N permette l'inserimento di un nuovo utente. Se la username scelta non è già in uso

L'opzione E permette il login ad un utente già esistente

L'opzione A permette di entrare nel sottomenù di administrator. Che prevedere il seguente menù:

- (R) remove account
- (S) show user
- (Q) exit Admin

L'opzione R permette di rimuovere un utente se presente

L'opzione S mostra gli utenti presenti nel database

L'opzione Q permette di uscire dal sottomenù di Administrator (Solution: login.py)