








Introduzione a **NumPy**: laboratorio 1

NumPy: Laboratorio 1


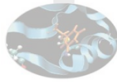
1. Creare i seguenti array, con il corretto *data-type* ed usando il minor numero di istruzioni possibile

```
[[1, 1, 1, 1],
 [1, 1, 1, 1],
 [1, 1, 1, 2],
 [1, 6, 1, 1]]

[[0., 0., 0., 0., 0.],
 [2., 0., 0., 0., 0.],
 [0., 3., 0., 0., 0.],
 [0., 0., 4., 0., 0.],
 [0., 0., 0., 5., 0.],
 [0., 0., 0., 0., 6.]]
```

Hint: consultare l'help di `np.diag`

2

NumPy: Laboratorio 1 (2)

2. Usare la funzione `np.tile` (`np.tile?` per accedere all'help) per creare il seguente array


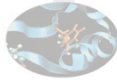
```
[[4, 3, 4, 3, 4, 3],
 [2, 1, 2, 1, 2, 1],
 [4, 3, 4, 3, 4, 3],
 [2, 1, 2, 1, 2, 1]]
```

3. Senza scriverlo esplicitamente, generare il seguente array e, costruire un secondo array contenente solo la 2° e la 4° riga

```
[[1, 6, 11],
 [2, 7, 12],
 [3, 8, 13],
 [4, 9, 14],
 [5, 10, 15]]
```

Hint: consultare l'help di `np.reshape`

3

NumPy: Laboratorio 1 (3)

4. A partire esclusivamente dagli array `a = np.arange(10)` e `b = np.arange(5)`, costruire l'array

```
[0, 1, 2, 3, 4, 4, 3, 2, 1, 0]
```


5. Dato l'array in figura:

- generare i sotto-array evidenziati in arancio, azzurro, rosso e verde
- generare altri sotto-array a piacere, usando *indexing & slicing* discussi

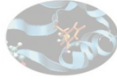
0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

NB: l'array in figura si ottiene con l'istruzione
`a = np.arange(6) + np.arange(0,51,10)[:,np.newaxis]`

4

 **SCAI**
SuperComputing Applications and Innovation

NumPy: Laboratorio 1 (4)



6. Costruire un array 2D e calcolare la media lungo le singole righe dei suoi valori che appartengono ad una colonna ad indice pari (dispari)
hint: `np.mean?`
7. Scrivere uno script Python che stampa un array con i numeri primi compresi tra 0 e 100.
NB: utilizzare l'algoritmo di generazione dei numeri primi noto come "Crivello di Eratostene" :



http://it.wikipedia.org/wiki/Crivello_di_Eratostene

5

 **SCAI**
SuperComputing Applications and Innovation

Introduzione a **NumPy**: laboratorio 2








NumPy: Laboratorio 2

- Usando lo *slicing* di array, calcolare la derivata numerica della funzione $\sin(x)$ tra 0 e 2π ; calcolare poi la massima distanza tra la derivata numerica e quella analitica e la media della stessa distanza sull'intera griglia.
hint: se abbiamo una griglia sufficientemente fitta in cui è valutata la funzione $\sin(x)$, la sua derivata è ben approssimata dal rapporto incrementale
- Calcolare l'integrale Monte-Carlo unidimensionale di una funzione data (ad es $f(x)=1+2x$)

$$\int_a^b f(x) dx \approx \frac{b-a}{n} \sum_{i=1}^n f(x_i)$$
 dove i punti x_i sono numeri random uniformemente distribuiti nell'intervallo a, b . Costruire uno script Python che implementi tale integrazione con un *loop* esplicito e tramite vettorizzazione con NumPy. Infine, testare il cpu time delle due porzioni del programma tramite la funzione `time.clock()`

7


NumPy: Laboratorio 2 (2)

- Dato un gioco nel quale si vince se la somma di 4 dadi è minore di 10, determinare se conviene giocarci qualora si vinca 10 volte il valore della posta
hint: usare `np.random.randint` per creare un array bidimensionale - con shape (4, n) - di interi casuali compresi tra 1 e 6
- leggere il file `angoli.txt` (timestep, angolo) con NumPy e costruire la distribuzione degli angoli tra 0 – 360 gradi a step di 5 gradi. Infine, stampare a video il grafico della distribuzione
 hint: usare ad esempio

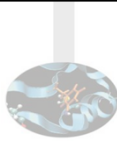

```
cont = int(angolo)/step
dist[cont] += 1
```

[plot da IPython: usare il magic command %pylab e poi plot(x,y)]

8



SCAI
SuperComputing Applications and Innovation



NumPy: Laboratorio 2 (3)

5. Leggere il file di dati `populations.txt`, contenente le popolazioni di lepri e linci (e carote) in Canada del Nord dal 1900 al 1920. Calcolare:

- La media e la deviazione standard di ogni specie nel periodo
- In quale anno ogni specie ha la massima popolazione
- Quale specie ha la massima popolazione per ogni anno (*hint*: `np.argsort` e *fancy indexing*)
- In quali anni almeno una delle popolazioni è superiore a 50.000
- I due anni, per ogni specie, in cui la popolazione è più bassa

9