

Matematica con Python

Progetto di costruzione di materiali per un laboratorio di matematica con Python e ipython-notebook.

L'idea è di realizzare dei materiali che diano degli stimoli e degli spunti, non dei materiali "finiti" ma materiali grezzi che richiedano agli alunni degli interventi per completare, correggere, ampliare, adattare ai propri interessi, stravolgere.

Non una scatola di montaggio con tutti i pezzi prefabbricati della matematica, pezzi solo da individuare, mettere al posto giusto, incollare e colorare, ma una scatola con materiale grezzo che permetta di pasticciare e sperimentare idee, progetti, fantasie. Non un tour nella matematica progettato con logica e coerenza, ma delle pedule, uno zaino e un sacco a pelo per esplorare la matematica, magari perdendosi nei suoi meandri e nelle sue trappole.

Non un luogo dove trovare risposte, ma dove inventare domande.

I materiali didattici presenti in questo lavoro sono da considerare sempre "in costruzione", anche questi sono spunti, suggestioni, suggerimenti da esplorare e da modificare non sono ricette costruite "con i dosaggi esatti degli esperti" da applicare pari pari.

Tutto quanto viene qui descritto è stato appreso gratuitamente da altri e viene messo a disposizione gratuitamente sotto la licenza Creative Commons CC-BY-SA alle seguenti condizioni:

- Ciunque può usarlo per qualunque scopo, modificarlo e diffonderlo citando la fonte.
- La licenza non può essere cambiata.

Territori da esplorare

In questo laboratorio non ci poniamo il problema di imparare a risolvere esercizi, ma di esplorare alcune regioni della matematica.

Lo facciamo con l'uso di strumenti automatici di calcolo. Questo permette di spostare l'attenzione dai problemi relativi all'*apprendimento* dei meccanismi legati alla matematica alla *costruzione* e all'*esplorazione* dei meccanismi stessi.

Non siamo noi a dover eseguire gli algoritmi (i matematici hanno inventato apposta i computer per questo) possiamo concentrarci nell'inventare algoritmi e capirne caratteristiche e limiti.

Di seguito elenco alcuni argomenti e riporto per ognuno un esempio di codice Python per dare una prima suggestione sulle possibilità offerte e sulle complicazioni aggiunte dal linguaggio.

Oggetti elementari: numeri e parole

Prima di affrontare argomenti più complessi dobbiamo prendere confidenza con gli oggetti elementari della matematica (e del linguaggio):

- numeri interi,
- numeri con la virgola,
- numeri binari,
- valori booleani,
- parole - stringhe.

In realtà le parole non sono considerate oggetti elementari della matematica, ma saper usare le parole è fondamentale anche per esprimere i concetti matematici.

Con questi oggetti si possono costruire espressioni ed esplorare le proprietà delle operazioni.

```
In [17]: 2**100
```

```
Out[17]: 1267650600228229401496703205376L
```

```
In [18]: 0.2 + 0.3
```

```
Out[18]: 0.5
```

```
In [19]: True and False
```

```
Out[19]: False
```

```
In [20]: 'ambaraba' + 'cici' + 'cocò'
```

```
Out[20]: 'ambarabacicicoc\xc3\xb2'
```

Approfondimenti sui numeri

Numeri complessi

Python mette a disposizione come oggetti elementari anche i numeri complessi. Se interessati, è possibile quindi esplorare anche la matematica del campo complesso.

```
In [21]: (4+3j)+(2-5j)
```

```
Out[21]: (6-2j)
```

La libreria math

Le operazioni predefinite in Python sono relativamente poche. La libreria math fornisce tutte le funzioni che servono a un matematico.

```
In [22]: import math
```

```
math.sin(math.pi/4)
```

Out[22]: 0.7071067811865475

Funzioni

Le funzioni sono un trucco dei matematici per dare un nome ad alcune espressioni. Il linguaggio di programmazione ci fornisce dei meccanismi per creare funzioni con uno o più variabili (parametri) e calcolarne il valore con con diversi argomenti.

```
In [23]: def quadrato(x):  
         return x*x  
         quadrato(12)
```

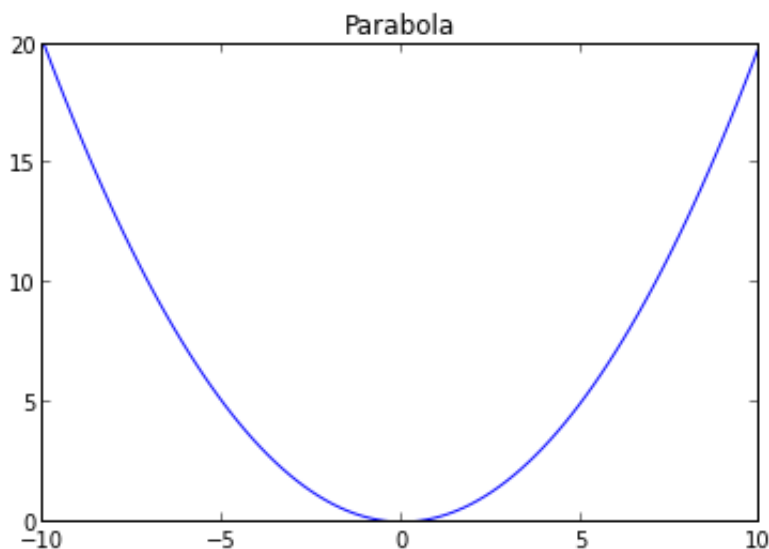
Out[23]: 144

Grafici

Usando apposite librerie è facile tracciare il grafico di funzioni nel piano o nello spazio.

```
In [24]: x = np.linspace(-10, 10, 100)  
         y1 = 0.2 * x**2  
         ax = plt.axes()  
         ax.plot(x, y1, c='blue')  
         ax.set_title("Parabola")  
         # modify all the axes elements in-place  
         #XKCDify(ax, expand_axes=True)
```

Out[24]: <matplotlib.text.Text at 0x2c3b8d0>



Successioni

Anche le successioni forniscono un interessante ambito di esplorazione e ricerca.

```
In [25]: for numero in range(10):  
         print(numero)
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

Algoritmi

La richiesta fatta all'inizio dello studio della matematica è quella di saper eseguire con ordine e precisione degli algoritmi inventati da altri. Ora un esecutore, ordinato e preciso, lo abbiamo a disposizione, possiamo concentrarci sull'attività (più interessante? senz'altro più "matematica") descrivere, progettare e realizzare algoritmi.

```
In [26]: def MCD(a, b):  
         if a == b:  
             return a  
         if a > b:  
             return MCD(a-b, b)  
         return MCD(a, b-a)
```

```
MCD(36, 28)
```

```
Out[26]: 4
```

Insiemi, vettori

A volte oggetti elementari non sono sufficienti per i problemi della matematica. Abbiamo bisogno di aggregare i dati in sequenze e poter lavorare con insiemi o con insiemi ordinati di oggetti.

Python ci mette a disposizione le normali operazioni con gli insiemi, mentre le operazioni di base con le sequenze sono:

- l'indicizzazione: estrarre un elemento dato il suo indice;
- l'affettamento: estrarre una sottosequenza.

Insiemi

I set sono strutture di dati che corrispondono agli insiemi matematici.

```
In [27]: A = set('abcdefghi')
print('A =', A)
B = set('aeiou')
print('B =', B)
print('A U B =', A|B)

('A =', set(['a', 'c', 'b', 'e', 'd', 'g', 'f', 'i', 'h']))
('B =', set(['a', 'i', 'e', 'u', 'o']))
('A U B =', set(['a', 'c', 'b', 'e', 'd', 'g', 'f', 'i', 'h', 'o', 'u']))
```

Stringhe

Le stringhe sono sequenze immutabili di caratteri.

```
In [28]: parola = 'Abracadabra'
print(parola[4])
print(parola[2:5])

c
rac
```

Liste

L'oggetto lista è una sequenza ordinata di oggetti. Le liste sono quindi degli oggetti definiti in modo ricorsivo. Gli elementi di una lista possono essere oggetti qualunque quindi anche altre liste che possono contenere altre liste che...

```
In [29]: valori = [2, 5, 7, 3]
print(sum(valori))

17
```

Nuovi oggetti matematici

Python permette non solo di operare con oggetti predefiniti, ma anche di creare dei nuovi oggetti con i loro attributi e i loro metodi. E definendo i metodi adatti è anche possibile usare questi nuovi oggetti all'interno di espressioni.

Punti

È possibile trattare un punto come oggetto matematico?

Si possono fare operazioni tra punti?

Che senso potrebbe avere sommare due punti o moltiplicarli?

```
In [30]: class Punto():
          def __init__(self, xcoord, ycoord):
              self.xcoord = xcoord
              self.ycoord = ycoord
          def __str__(self):
              return '{0}; {1}'.format(self.xcoord, self.ycoord)
          def __add__(self, other):
              return Punto(self.xcoord + other.xcoord,
                           self.ycoord + other.ycoord)

a = Punto(2, 3)
b = Punto(4, 5)
print(a + b)
```

(6; 8)

Frazioni

Come è possibile rappresentare una frazione?

Come eseguire le operazioni con le frazioni?

In []:

Vettori

Un vettore è un oggetto matematico dotato di modulo, direzione e verso.

Come può essere rappresentato matematicamente?

Che operazioni possono aver senso con i vettori?

In []:

In questo capitolo abbiamo visto:

Quali sono alcuni territori della matematica che si possono esplorare anche con l'aiuto di un linguaggio di programmazione.

Alcuni esempi:

- Numeri e parole
- Approfondimenti sui numeri
- Funzioni
- Grafici di funzioni
- Successioni
- Algoritmi
- Insiemi, vettori
- Nuovi oggetti