

PAUL DEITEL  
HARVEY DEITEL



# Introduzione a Python®

Per l'informatica e la data science

A cura di Pietro Codara e Carlo Mereghetti



**MyLab** Codice per accedere  
alla piattaforma



# **Introduzione a Python®**



# Introduzione a Python®

Per l'informatica e la data science

Paul Deitel  
Harvey Deitel

A cura di Pietro Codara e Carlo Mereghetti

  Pearson

 Pearson

© 2021 Pearson Italia, Milano - Torino

*Authorized translation from the English language edition, entitled INTRO TO PYTHON FOR COMPUTER SCIENCE AND DATA SCIENCE: LEARNING TO PROGRAM WITH AI, BIG DATA AND THE CLOUD, 1<sup>st</sup> Edition by PAUL DEITEL; HARVEY DEITEL; HARVEY DEITEL, published by Pearson Education, Inc, publishing as Pearson, Copyright © 2020.*

*All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.*

*Italian language edition published by Pearson Italia S.p.A., Copyright © 2021.*

Per i passi antologici, per le citazioni, per le riproduzioni grafiche, cartografiche e fotografiche appartenenti alla proprietà di terzi, inseriti in quest'opera, l'editore è a disposizione degli aventi diritto non potuti reperire nonché per eventuali non volute omissioni e/o errori di attribuzione nei riferimenti.

È vietata la riproduzione, anche parziale o ad uso interno didattico, con qualsiasi mezzo, non autorizzata.

Le fotocopie per uso personale del lettore possono essere effettuate nei limiti del 15% di ciascun volume dietro pagamento alla SIAE del compenso previsto dall'art. 68, commi 4 e 5, della legge 22 aprile 1941, n. 633.

Le riproduzioni effettuate per finalità di carattere professionale, economico o commerciale o comunque per uso diverso da quello personale possono essere effettuate a seguito di specifica autorizzazione rilasciata da CLEARedi, Corso di Porta Romana 108, 20122 Milano, e-mail autorizzazioni@clearedi.org e sito web www.clearedi.org.

I nostri libri sono ecosostenibili: la carta è prodotta sostenendo il ciclo naturale e per ogni albero tagliato ne viene piantato un altro; il cellofan è realizzato con plastiche da recupero ambientale o riciclate; gli inchiostri sono naturali e atossici; i libri sono prodotti in Italia e l'impatto del trasporto è ridotto al minimo.

Curatori per l'edizione italiana: Pietro Codara, Carlo Mereghetti

Traduzione: Diego Valota

Realizzazione editoriale: Giulia Maselli e Maria Mantero

Grafica di copertina: Simone Tartaglia

Immagine di copertina: © Denel/Shutterstock

Stampa: Arti Grafiche Battaia – Zibido San Giacomo (MI)

Seaborn © Copyright 2012-2020, Michael Waskom. Created using Sphinx 2.3.1.

Matplotlib © Copyright 2002 - 2012 John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the Matplotlib development team; 2012 - 2021 The Matplotlib development team. Created using Sphinx 3.1.1. Doc version v3.3.4.

Tutti i marchi citati nel testo sono di proprietà dei loro detentori.

9788891915924

Printed in Italy

1<sup>a</sup> edizione: aprile 2021

Ristampa  
00 01 02 03 04

Anno  
21 22 23 24 25

#### LIBRI DI TESTO E SUPPORTI DIDATTICI

Il sistema di gestione per la qualità della Casa Editrice è certificato in conformità alla norma UNI EN ISO 9001:2015 per l'attività di progettazione, realizzazione e commercializzazione di: • prodotti editoriali scolastici, dizionari lessicografici, prodotti per l'editoria di varia ed università • materiali didattici multimediali off-line • corsi di formazione e specializzazione in aula, a distanza, e-learning.

Member of CISQ Federation



CERTIFIED MANAGEMENT SYSTEM  
ISO 9001

*In memoria di Marvin Minsky,  
uno dei padri fondatori dell'intelligenza artificiale.*

*È stato un privilegio essere tuo studente in due corsi  
sull'intelligenza artificiale al M.I.T.  
Hai ispirato i tuoi studenti a pensare oltre i limiti.*

*Harvey Deitel*





# Sommario

Prefazione	XIX
Prima di cominciare	XXXV
Pearson MyLab	XXXIX

<b>Capitolo 1</b>	<b>Introduzione ai computer e a Python</b>	<b>1</b>
1.1	<b>Introduzione</b>	<b>1</b>
1.2	<b>Hardware e software</b>	<b>2</b>
	1.2.1 Legge di Moore	3
	1.2.2 Organizzazione di un computer	3
1.3	<b>Gerarchia dei dati</b>	<b>5</b>
1.4	<b>Linguaggi macchina, assembly e di alto livello</b>	<b>7</b>
1.5	<b>Introduzione alla tecnologia a oggetti</b>	<b>9</b>
1.6	<b>Sistemi operativi</b>	<b>11</b>
1.7	<b>Python</b>	<b>13</b>
1.8	<b>Ecco le librerie!</b>	<b>15</b>
	1.8.1 Libreria Standard di Python	15
	1.8.2 Librerie per la data science	16
1.9	<b>Altri linguaggi di programmazione popolari</b>	<b>17</b>
1.10	<b>Test guidato: usare IPython e i notebook Jupyter</b>	<b>18</b>
	1.10.1 Usare IPython in modo interattivo come calcolatrice	18
	1.10.2 Eseguire un programma Python usando l'interprete IPython	19
	1.10.3 Scrivere ed eseguire codice in un notebook Jupyter	21
1.11	<b>Internet e World Wide Web</b>	<b>25</b>
	1.11.1 Internet: una rete di reti	25
	1.11.2 World Wide Web: come rendere Internet user-friendly	25
	1.11.3 Il cloud	26
	1.11.4 Internet delle cose	26

<b>1.12</b>	<b>Tecnologie software</b>	<b>27</b>
<b>1.13</b>	<b>Quanto sono grandi i big data?</b>	<b>28</b>
1.13.1	Analisi dei big data	32
1.13.2	Big data e data science stanno facendo la differenza: casi d'uso	32
<b>1.14</b>	<b>Caso di studio: Un'applicazione mobile</b>	<b>34</b>
<b>1.15</b>	<b>Introduzione alla data science: intelligenza artificiale all'intersezione tra informatica e data science</b>	<b>35</b>
	<b>Esercizi</b>	<b>37</b>
<b>Capitolo 2</b>	<b>Introduzione alla programmazione Python</b>	<b>41</b>
<b>2.1</b>	<b>Introduzione</b>	<b>41</b>
<b>2.2</b>	<b>Variabili e istruzioni di assegnamento</b>	<b>41</b>
<b>2.3</b>	<b>Operazioni aritmetiche</b>	<b>43</b>
<b>2.4</b>	<b>Funzione print e un'introduzione alle stringhe tra virgolette singole e doppie</b>	<b>47</b>
<b>2.5</b>	<b>Stringhe tra virgolette triple</b>	<b>49</b>
<b>2.6</b>	<b>Acquisire input dall'utente</b>	<b>51</b>
<b>2.7</b>	<b>Prendere decisioni: l'istruzione if e gli operatori di confronto</b>	<b>52</b>
<b>2.8</b>	<b>Oggetti e tipizzazione dinamica</b>	<b>57</b>
<b>2.9</b>	<b>Introduzione alla data science: statistiche descrittive basilari</b>	<b>58</b>
<b>2.10</b>	<b>Riepilogo</b>	<b>61</b>
	<b>Esercizi</b>	<b>61</b>
<b>Capitolo 3</b>	<b>Istruzioni di controllo e sviluppo dei programmi</b>	<b>65</b>
<b>3.1</b>	<b>Introduzione</b>	<b>65</b>
<b>3.2</b>	<b>Algoritmi</b>	<b>65</b>
<b>3.3</b>	<b>Pseudocodice</b>	<b>66</b>
<b>3.4</b>	<b>Istruzioni di controllo</b>	<b>67</b>
<b>3.5</b>	<b>Istruzione if</b>	<b>69</b>
<b>3.6</b>	<b>Istruzioni if...else e if...elif...else</b>	<b>71</b>
<b>3.7</b>	<b>Istruzione while</b>	<b>75</b>
<b>3.8</b>	<b>Istruzione for</b>	<b>77</b>
3.8.1	Iterabili, liste e iteratori	78
3.8.2	Funzione integrata range	78
<b>3.9</b>	<b>Assegnamenti estesi</b>	<b>79</b>
<b>3.10</b>	<b>Sviluppo del programma: ripetizioni controllate da sequenze</b>	<b>80</b>
3.10.1	Dichiarazione dei requisiti	80
3.10.2	Pseudocodice dell'algoritmo	80

	3.10.3	Codifica dell'algoritmo in Python	81
	3.10.4	Introduzione alle stringhe formattate	82
<b>3.11</b>		<b>Sviluppo del programma: ripetizioni controllate da sentinelle</b>	<b>82</b>
<b>3.12</b>		<b>Sviluppo del programma: istruzioni di controllo annidate</b>	<b>86</b>
<b>3.13</b>		<b>Funzione integrata range: un approfondimento</b>	<b>90</b>
<b>3.14</b>		<b>Usare il tipo Decimal per valori monetari</b>	<b>91</b>
<b>3.15</b>		<b>Istruzioni break e continue</b>	<b>94</b>
<b>3.16</b>		<b>Operatori booleani and, or e not</b>	<b>95</b>
<b>3.17</b>		<b>Introduzione alla data science: misure di tendenza centrale (media, mediana e moda)</b>	<b>98</b>
<b>3.18</b>		<b>Riepilogo</b>	<b>99</b>
		<b>Esercizi</b>	<b>100</b>
<b>Capitolo 4</b>		<b>Funzioni</b>	<b>107</b>
	<b>4.1</b>	<b>Introduzione</b>	<b>107</b>
	<b>4.2</b>	<b>Definire le funzioni</b>	<b>107</b>
	<b>4.3</b>	<b>Funzioni con parametri multipli</b>	<b>110</b>
	<b>4.4</b>	<b>Generazione di numeri casuali</b>	<b>112</b>
	<b>4.5</b>	<b>Caso di studio: Un gioco di fortuna</b>	<b>115</b>
	<b>4.6</b>	<b>Libreria Standard di Python</b>	<b>118</b>
	<b>4.7</b>	<b>Funzioni del modulo math</b>	<b>119</b>
	<b>4.8</b>	<b>Esplorare con il completamento automatico di IPython</b>	<b>120</b>
	<b>4.9</b>	<b>Parametri con valori predefiniti</b>	<b>121</b>
	<b>4.10</b>	<b>Argomenti denominati</b>	<b>122</b>
	<b>4.11</b>	<b>Liste arbitrarie di argomenti</b>	<b>122</b>
	<b>4.12</b>	<b>Metodi: funzioni che appartengono a oggetti</b>	<b>124</b>
	<b>4.13</b>	<b>Regole di visibilità</b>	<b>124</b>
	<b>4.14</b>	<b>import: un approfondimento</b>	<b>126</b>
	<b>4.15</b>	<b>Passaggio di argomenti a una funzione: un approfondimento</b>	<b>128</b>
	<b>4.16</b>	<b>Pila delle chiamate a funzione</b>	<b>130</b>
	<b>4.17</b>	<b>Programmazione in stile funzionale</b>	<b>131</b>
	<b>4.18</b>	<b>Introduzione alla data science: misure di dispersione</b>	<b>133</b>
	<b>4.19</b>	<b>Riepilogo</b>	<b>135</b>
		<b>Esercizi</b>	<b>135</b>
<b>Capitolo 5</b>		<b>Sequenze: liste e tuple</b>	<b>139</b>
	<b>5.1</b>	<b>Introduzione</b>	<b>139</b>

<b>5.2</b>	<b>Liste</b>	<b>140</b>
<b>5.3</b>	<b>Tuple</b>	<b>144</b>
<b>5.4</b>	<b>Spacchettare sequenze</b>	<b>146</b>
<b>5.5</b>	<b>Porzioni di sequenze</b>	<b>149</b>
<b>5.6</b>	<b>Istruzione del</b>	<b>152</b>
<b>5.7</b>	<b>Passare le liste come argomenti alle funzioni</b>	<b>154</b>
<b>5.8</b>	<b>Ordinare liste</b>	<b>155</b>
<b>5.9</b>	<b>Ricerca nelle sequenze</b>	<b>157</b>
<b>5.10</b>	<b>Altri metodi delle liste</b>	<b>158</b>
<b>5.11</b>	<b>Simulare le pile con le liste</b>	<b>161</b>
<b>5.12</b>	<b>Comprensione di lista</b>	<b>162</b>
<b>5.13</b>	<b>Espressioni generatrici</b>	<b>164</b>
<b>5.14</b>	<b>Funzioni filter, map e reduce</b>	<b>165</b>
<b>5.15</b>	<b>Altre funzioni per l'elaborazione di sequenze</b>	<b>168</b>
<b>5.16</b>	<b>Liste bidimensionali</b>	<b>170</b>
<b>5.17</b>	<b>Introduzione alla data science: simulazione e visualizzazioni statiche</b>	<b>173</b>
	5.17.1 Grafici per 600, 60.000 e 6.000.000 di lanci di dado	173
	5.17.2 Visualizzare frequenze e percentuali del lancio di un dado	175
<b>5.18</b>	<b>Riepilogo</b>	<b>181</b>
	<b>Esercizi</b>	<b>182</b>
<b>Capitolo 6</b>	<b>Dizionari e insiemi</b>	<b>191</b>
<b>6.1</b>	<b>Introduzione</b>	<b>191</b>
<b>6.2</b>	<b>Dizionari</b>	<b>191</b>
	6.2.1 Creazione di un dizionario	192
	6.2.2 Iterare su un dizionario	193
	6.2.3 Operazioni elementari sui dizionari	194
	6.2.4 Metodi keys e values dei dizionari	196
	6.2.5 Confronti nei dizionari	197
	6.2.6 Esempio: dizionario dei voti degli studenti	198
	6.2.7 Esempio: conteggio delle parole	199
	6.2.8 Metodo update dei dizionari	201
	6.2.9 Comprensione dei dizionari	201
<b>6.3</b>	<b>Insiemi</b>	<b>202</b>
	6.3.1 Confrontare insiemi	204
	6.3.2 Operazioni matematiche tra insiemi	206
	6.3.3 Operatori e metodi per insiemi mutabili	207
	6.3.4 Comprensione degli insiemi	209

<b>6.4</b>	<b>Introduzione alla data science: visualizzazioni dinamiche</b>	<b>209</b>
6.4.1	Come funzionano le visualizzazioni dinamiche	209
6.4.2	Implementare una visualizzazione dinamica	212
<b>6.5</b>	<b>Riepilogo</b>	<b>214</b>
	<b>Esercizi</b>	<b>215</b>
<b>Capitolo 7</b>	<b>Programmazione orientata ai vettori con NumPy</b>	<b>219</b>
<b>7.1</b>	<b>Introduzione</b>	<b>219</b>
<b>7.2</b>	<b>Creare array a partire da dati esistenti</b>	<b>220</b>
<b>7.3</b>	<b>Attributi degli array</b>	<b>221</b>
<b>7.4</b>	<b>Riempire gli array con valori specifici</b>	<b>223</b>
<b>7.5</b>	<b>Creare array a partire da intervalli</b>	<b>224</b>
<b>7.6</b>	<b>Prestazioni di liste e array: introduzione di %timeit</b>	<b>225</b>
<b>7.7</b>	<b>Operatori degli array</b>	<b>227</b>
<b>7.8</b>	<b>Metodi di calcolo di NumPy</b>	<b>229</b>
<b>7.9</b>	<b>Funzioni universali</b>	<b>231</b>
<b>7.10</b>	<b>Indicizzazione e porzionamento</b>	<b>233</b>
<b>7.11</b>	<b>Viste: copie superficiali</b>	<b>235</b>
<b>7.12</b>	<b>Copie profonde</b>	<b>237</b>
<b>7.13</b>	<b>Cambi di forma e trasposizioni</b>	<b>237</b>
<b>7.14</b>	<b>Introduzione alla data science: Series e DataFrame di Pandas</b>	<b>240</b>
7.14.1	Series	241
7.14.2	DataFrame	245
<b>7.15</b>	<b>Riepilogo</b>	<b>253</b>
	<b>Esercizi</b>	<b>254</b>
<b>Capitolo 8</b>	<b>Stringhe: un approfondimento</b>	<b>261</b>
<b>8.1</b>	<b>Introduzione</b>	<b>261</b>
<b>8.2</b>	<b>Formattazione delle stringhe</b>	<b>262</b>
8.2.1	Tipi di presentazione	262
8.2.2	Larghezza di campo e allineamento	264
8.2.3	Formattazione di numeri	265
8.2.4	Metodo format delle stringhe	266
<b>8.3</b>	<b>Concatenare e ripetere stringhe</b>	<b>267</b>
<b>8.4</b>	<b>Togliere gli spazi dalle stringhe</b>	<b>267</b>
<b>8.5</b>	<b>Invertire caratteri minuscoli e maiuscoli</b>	<b>268</b>
<b>8.6</b>	<b>Operatori di confronto per stringhe</b>	<b>269</b>
<b>8.7</b>	<b>Ricerca delle sottostringhe</b>	<b>270</b>

<b>8.8</b>	<b>Rimpiazzare sottostringhe</b>	<b>271</b>
<b>8.9</b>	<b>Suddividere e unire le stringhe</b>	<b>272</b>
<b>8.10</b>	<b>Metodi per controllare caratteri</b>	<b>274</b>
<b>8.11</b>	<b>Stringhe raw</b>	<b>275</b>
<b>8.12</b>	<b>Introduzione alle espressioni regolari</b>	<b>276</b>
	8.12.1 Modulo re e funzione fullmatch	276
	8.12.2 Rimpiazzare sottostringhe e suddividere stringhe	280
	8.12.3 Altre funzioni di ricerca e accesso alle corrispondenze	281
<b>8.13</b>	<b>Introduzione alla data science: Pandas, espressioni regolari e data munging</b>	<b>284</b>
<b>8.14</b>	<b>Riepilogo</b>	<b>288</b>
	<b>Esercizi</b>	<b>289</b>
<b>Capitolo 9</b>	<b>File ed eccezioni</b>	<b>295</b>
<b>9.1</b>	<b>Introduzione</b>	<b>295</b>
<b>9.2</b>	<b>File</b>	<b>296</b>
<b>9.3</b>	<b>Elaborazione dei file di testo</b>	<b>297</b>
	9.3.1 Scrivere in un file di testo: introduzione all'istruzione with	297
	9.3.2 Leggere dati da un file di testo	298
<b>9.4</b>	<b>Aggiornare file di testo</b>	<b>300</b>
<b>9.5</b>	<b>Serializzazione con JSON</b>	<b>302</b>
<b>9.6</b>	<b>Focus sulla sicurezza: serializzazione e deserializzazione con pickle</b>	<b>305</b>
<b>9.7</b>	<b>Note aggiuntive sui file</b>	<b>305</b>
<b>9.8</b>	<b>Gestione delle eccezioni</b>	<b>306</b>
	9.8.1 Divisione per zero e input non valido	306
	9.8.2 Istruzioni try	307
	9.8.3 Catturare eccezioni multiple con una sola clausola except	309
	9.8.4 Quali eccezioni vengono sollevate da una funzione o da un metodo?	310
	9.8.5 Quale codice andrebbe messo in una suite try?	310
<b>9.9</b>	<b>Clausola finally</b>	<b>310</b>
<b>9.10</b>	<b>Sollevare un'eccezione esplicitamente</b>	<b>313</b>
<b>9.11</b>	<b>(Opzionale) Svolgimento dello stack e traceback</b>	<b>313</b>
<b>9.12</b>	<b>Introduzione alla data science: lavorare con i file CSV</b>	<b>315</b>
	9.12.1 Il modulo csv della Libreria Standard di Python	315
	9.12.2 Leggere file CSV e inserirli nell'oggetto DataFrame di Pandas	318
	9.12.3 Lettura del Titanic Disaster Dataset	319
	9.12.4 Semplice analisi dati con il Titanic Disaster Dataset	320
	9.12.5 Istogramma delle età dei passeggeri	321

<b>9.13</b>	<b>Riepilogo</b>	<b>322</b>
	<b>Esercizi</b>	<b>322</b>
<b>Capitolo 10</b>	<b>Programmazione orientata agli oggetti</b>	<b>329</b>
<b>10.1</b>	<b>Introduzione</b>	<b>329</b>
<b>10.2</b>	<b>Classe personalizzata per un conto bancario</b>	<b>331</b>
	10.2.1 Test guidato: classe Account	331
	10.2.2 Definizione della classe Account	333
	10.2.3 Composizione: riferimenti agli oggetti come membri delle classi	334
<b>10.3</b>	<b>Controllo degli accessi sugli attributi</b>	<b>336</b>
<b>10.4</b>	<b>Proprietà per l'accesso ai dati</b>	<b>336</b>
	10.4.1 Test guidato: classe Time	337
	10.4.2 Definizione della classe Time	338
	10.4.3 Note di progettazione della classe Time	342
<b>10.5</b>	<b>Simulare attributi "privati"</b>	<b>343</b>
<b>10.6</b>	<b>Caso di studio: Simulare mescolatura e distribuzione delle carte</b>	<b>345</b>
	10.6.1 Test guidato: classi Card e DeckOfCards	345
	10.6.2 Classe Card: introduzione degli attributi di classe	347
	10.6.3 Classe DeckOfCards	349
	10.6.4 Visualizzare immagini delle carte con Matplotlib	350
<b>10.7</b>	<b>Ereditarietà: classi base e sottoclassi</b>	<b>353</b>
<b>10.8</b>	<b>Costruzione di una gerarchia di ereditarietà: introduzione al polimorfismo</b>	<b>355</b>
	10.8.1 Classe base CommissionEmployee	355
	10.8.2 Sottoclasse SalariedCommissionEmployee	358
	10.8.3 Elaborazione polimorfica di CommissionEmployee e SalariedCommissionEmployee	362
	10.8.4 Nota sulla programmazione orientata agli oggetti e su quella basata sugli oggetti	362
<b>10.9</b>	<b>Duck typing e polimorfismo</b>	<b>363</b>
<b>10.10</b>	<b>Sovrascrittura degli operatori</b>	<b>364</b>
	10.10.1 Test guidato: classe Complex	365
	10.10.2 Definizione della classe Complex	366
<b>10.11</b>	<b>Gerarchia delle classi delle eccezioni ed eccezioni personalizzate</b>	<b>368</b>
<b>10.12</b>	<b>Tuple denominate</b>	<b>369</b>
<b>10.13</b>	<b>Una breve introduzione alle nuove classi di dati di Python 3.7</b>	<b>370</b>
	10.13.1 Creazione della classe di dati Card	371
	10.13.2 Utilizzare la classe di dati Card	374
	10.13.3 Vantaggi delle classi di dati rispetto alle tuple denominate	376
	10.13.4 Vantaggi delle classi di dati rispetto alle classi tradizionali	376

<b>10.14</b>	<b>Testing di unità con docstring e doctest</b>	<b>376</b>
<b>10.15</b>	<b>Spazi dei nomi e visibilità</b>	<b>381</b>
<b>10.16</b>	<b>Introduzione alla data science: serie temporali e regressione lineare semplice</b>	<b>384</b>
<b>10.17</b>	<b>Riepilogo</b>	<b>392</b>
	<b>Esercizi</b>	<b>392</b>
<b>Capitolo 11</b>	<b>Pensare come un informatico: ricorsione, ricerca, ordinamento e notazione O grande</b>	<b>401</b>
<b>11.1</b>	<b>Introduzione</b>	<b>401</b>
<b>11.2</b>	<b>Fattoriali</b>	<b>402</b>
<b>11.3</b>	<b>Esempio di ricorsione per il fattoriale</b>	<b>402</b>
<b>11.4</b>	<b>Serie ricorsiva di Fibonacci</b>	<b>405</b>
<b>11.5</b>	<b>Confronto tra ricorsione e iterazione</b>	<b>408</b>
<b>11.6</b>	<b>Ricerca e ordinamento</b>	<b>408</b>
<b>11.7</b>	<b>Ricerca lineare</b>	<b>409</b>
<b>11.8</b>	<b>Efficienza degli algoritmi: O grande</b>	<b>410</b>
<b>11.9</b>	<b>Ricerca binaria</b>	<b>412</b>
	11.9.1 Implementazione della ricerca binaria	413
	11.9.2 O grande della ricerca binaria	415
<b>11.10</b>	<b>Algoritmi di ordinamento</b>	<b>416</b>
<b>11.11</b>	<b>Ordinamento per selezione</b>	<b>416</b>
	11.11.1 Implementazione dell'ordinamento per selezione	416
	11.11.2 Funzione di utilità <code>print_pass</code>	418
	11.11.3 O grande dell'ordinamento per selezione	419
<b>11.12</b>	<b>Ordinamento per inserimento</b>	<b>419</b>
	11.12.1 Implementazione dell'ordinamento per inserimento	419
	11.12.2 O grande dell'ordinamento per inserimento	421
<b>11.13</b>	<b>Ordinamento per fusione</b>	<b>421</b>
	11.13.1 Implementazione dell'ordinamento per fusione	422
	11.13.2 O grande dell'ordinamento per fusione	426
<b>11.14</b>	<b>Riepilogo dell'efficienza degli algoritmi di ordinamento e ricerca di questo capitolo</b>	<b>426</b>
<b>11.15</b>	<b>Visualizzare gli algoritmi</b>	<b>427</b>
	11.15.1 Funzioni generatrici	429
	11.15.2 Implementazione dell'ordinamento per selezione animato	430
<b>11.16</b>	<b>Riepilogo</b>	<b>435</b>
	<b>Esercizi</b>	<b>435</b>



<b>Capitolo 12</b>	<b>Natural Language Processing (NLP)</b>	<b>ONLINE</b>
<b>Capitolo 13</b>	<b>Data Mining Twitter</b>	<b>ONLINE</b>
<b>Capitolo 14</b>	<b>IBM Watson and Cognitive Computing</b>	<b>ONLINE</b>
<b>Capitolo 15</b>	<b>Machine learning: classificazione, regressione e clustering</b>	<b>449</b>
<b>15.1</b>	<b>Introduzione al machine learning</b>	<b>449</b>
15.1.1	Scikit-learn	450
15.1.2	Tipologie di machine learning	451
15.1.3	Dataset inclusi in Scikit-learn	453
15.1.4	Passi da seguire in un tipico caso di studio nella data science	453
<b>15.2</b>	<b>Caso di studio: Classificazione con k-nearest neighbors e il dataset Digits, Parte 1</b>	<b>454</b>
15.2.1	Algoritmo k-nearest neighbors	455
15.2.2	Caricare il dataset	456
15.2.3	Visualizzare i dati	460
15.2.4	Dividere i dati per l'addestramento e il testing	462
15.2.5	Creare il modello	463
15.2.6	Addestrare il modello	463
15.2.7	Fare previsioni sulle classi di cifre	464
<b>15.3</b>	<b>Caso di studio: Classificazione con k-nearest neighbors e il dataset Digits, Parte 2</b>	<b>465</b>
15.3.1	Metriche per la precisione del modello	466
15.3.2	Convalida incrociata k-fold	469
15.3.3	Eseguire diversi modelli per trovare il migliore	470
15.3.4	Regolazione degli iperparametri	472
<b>15.4</b>	<b>Caso di studio: Serie temporali e regressione lineare semplice</b>	<b>473</b>
<b>15.5</b>	<b>Caso di studio: Regressione lineare multipla sul dataset California Housing</b>	<b>477</b>
15.5.1	Caricare il dataset	478
15.5.2	Esplorare i dati con Pandas	480
15.5.3	Visualizzare le feature	481
15.5.4	Dividere i dati per l'addestramento e per il testing	485
15.5.5	Addestrare il modello	486
15.5.6	Testare il modello	487
15.5.7	Visualizzare i prezzi attesi e quelli previsti	487
15.5.8	Metriche per i modelli di regressione	488
15.5.9	Scegliere il modello migliore	489
<b>15.6</b>	<b>Caso di studio: Machine learning non supervisionato. Parte 1: ridurre le dimensioni</b>	<b>490</b>

<b>15.7</b>	<b>Caso di studio: Machine learning non supervisionato.</b>	
	<b>Parte 2: clustering k-means</b>	<b>493</b>
15.7.1	Caricare il dataset Iris	495
15.7.2	Esplorare il dataset Iris: statistiche descrittive con Pandas	497
15.7.3	Visualizzare il dataset con pairplot di Seaborn	498
15.7.4	Utilizzare lo stimatore KMeans	501
15.7.5	Riduzione della dimensionalità con l'analisi delle componenti principali	503
15.7.6	Scegliere il miglior stimatore per il clustering	505
<b>15.8</b>	<b>Riepilogo</b>	<b>506</b>
	<b>Esercizi</b>	<b>507</b>
<b>Capitolo 16</b>	<b>Deep learning</b>	<b>515</b>
<b>16.1</b>	<b>Introduzione</b>	<b>515</b>
16.1.1	Applicazioni del deep learning	517
16.1.2	Dimostrazioni del deep learning	518
16.1.3	Risorse su Keras	518
<b>16.2</b>	<b>Dataset integrati in Keras</b>	<b>518</b>
<b>16.3</b>	<b>Ambienti personalizzati di Anaconda</b>	<b>519</b>
<b>16.4</b>	<b>Reti neurali</b>	<b>520</b>
<b>16.5</b>	<b>Tensori</b>	<b>522</b>
<b>16.6</b>	<b>Reti neurali convoluzionali per la visione e classificazione multiclasse sul dataset MNIST</b>	<b>523</b>
16.6.1	Caricare il dataset MNIST	525
16.6.2	Esplorazione dei dati	525
16.6.3	Preparazione dei dati	527
16.6.4	Creare la rete neurale	529
16.6.5	Addestrare e valutare il modello	537
16.6.6	Salvare e caricare un modello	542
<b>16.7</b>	<b>Visualizzare l'addestramento della rete neurale con TensorBoard</b>	<b>542</b>
<b>16.8</b>	<b>ConvnetJS: addestramento e visualizzazione per il deep learning basati sul browser</b>	<b>545</b>
<b>16.9</b>	<b>Reti neurali ricorrenti per le sequenze; analisi del sentiment con il dataset IMDb</b>	<b>546</b>
16.9.1	Caricare il dataset delle recensioni cinematografiche di IMDb	547
16.9.2	Esplorazione dei dati	547
16.9.3	Preparazione dei dati	550
16.9.4	Creare la rete neurale	551
16.9.5	Addestrare e valutare il modello	553
<b>16.10</b>	<b>Regolazione dei modelli di deep learning</b>	<b>554</b>

---

<b>16.11</b>	<b>Modelli preaddestrati di convnet su ImageNet</b>	<b>555</b>
<b>16.12</b>	<b>Apprendimento con rinforzo</b>	<b>556</b>
16.12.1	Deep q-learning	557
16.12.2	OpenAI Gym	557
<b>16.13</b>	<b>Riepilogo</b>	<b>558</b>
	<b>Esercizi</b>	<b>558</b>
<b>Capitolo 17</b>	<b>Big Data: Hadoop, Spark, NoSQL and IoT</b>	<b>ONLINE</b>
	<b>Indice analitico</b>	<b>567</b>



# Prefazione

“C’è oro in quelle colline laggiù!”<sup>1</sup>

Da molti decenni, forti tendenze sono all’opera: l’hardware dei computer sta rapidamente diventando sempre più veloce, piccolo ed economico; la banda (cioè la capacità di trasportare informazioni) di Internet sta velocemente aumentando, rimanendo conveniente; anche il software di qualità è diventato largamente disponibile e gratuito, o quasi, grazie al movimento “open source”. Molto presto l’“Internet delle cose” conetterà decine di miliardi di dispositivi di ogni tipo immaginabile. Tutto ciò genererà un’enorme quantità di dati, che cresceranno sempre di più e arriveranno sempre più velocemente.

Non molti anni fa, se ci avessero detto che avremmo scritto un libro di introduzione alla programmazione per i primi anni di università con un elefante colorato in copertina, simbolo di qualcosa di grande, come i big data, la nostra reazione sarebbe stata “Hmmm?”. Se ci avessero detto che avremmo incluso anche l’IA (intelligenza artificiale), avremmo obiettato: “Veramente? Non è una cosa troppo avanzata per un programmatore principiante?”.

Se ci avessero detto che avremmo aggiunto anche “data science” nel titolo, avremmo commentato: “Ma i dati non fanno già parte degli argomenti trattati dall’informatica? Perché dovremmo aver bisogno di una diversa disciplina accademica?” Ebbene, nella programmazione moderna le ultime innovazioni hanno tutte a che fare con i dati: data science, analisi dei dati, big data, database relazionali (SQL), database NoSQL e NewSQL.

Quindi eccoci qui! Benvenuti a *Introduzione a Python*.

In questo libro apprenderete sperimentando con le tecnologie informatiche attualmente più efficaci e all’avanguardia. Come vedrete, useremo una miscela di argomenti di informatica e data science appropriata a corsi introduttivi delle due discipline e ad altri corsi universitari correlati. Programmerete in Python, il linguaggio più in crescita tra quelli più diffusi al mondo. In questa prefazione vi descriviamo “l’anima” del libro.

I programmatori professionisti apprezzano molto il linguaggio Python per la sua potenza espressiva, la sua leggibilità, la sua concisione e l’interattività. Inoltre, essi amano il mondo dello sviluppo software open source, grazie al quale vengono creati software riutilizzabili in un ampio spettro di applicazioni.

Sia che voi siate un docente, uno studente principiante o un programmatore esperto, questo libro ha molto da offrirvi. Python è un eccellente linguaggio sia per i principianti che per lo sviluppo di applicazioni industriali. Per chi è nuovo alla programmazione, stabiliremo delle solide basi nei primi capitoli.

Ci auguriamo che *Introduzione a Python* sia per voi educativo, divertente e stimolante. Per noi è stato un piacere lavorare a questo progetto.

## Il linguaggio Python nei corsi di informatica e di data science

Molte università americane di alto livello sono passate a Python come linguaggio d’elezione per l’insegnamento nei corsi introduttivi di informatica, “otto tra i primi 10 dipartimenti di informatica (80%) e 27 tra i primi 39

---

1. Autore sconosciuto, spesso attribuita erroneamente a Mark Twain.

(69%)” usano Python.<sup>2</sup> Attualmente Python è particolarmente popolare nell’educazione e nel calcolo scientifico<sup>3</sup> e ha recentemente sorpassato R come linguaggio più usato nella data science.<sup>4, 5, 6</sup>

## Architettura modulare

Ci aspettiamo che i corsi di laurea triennale in informatica si evolveranno per includere una componente di data science. Questo libro è pronto per agevolare nuovi percorsi di tale tipo, oltre a essere adatto ai corsi introduttivi di data science che hanno una componente di programmazione Python.

L’**architettura modulare** del libro (lo potete vedere consultando il Sommario) ci permette di andare incontro alle esigenze dell’utenza informatica, della data science e di altre aree. I docenti potranno adattarlo comodamente a un’ampia offerta di corsi rivolti a **studenti di diverse discipline**.

I Capitoli 1-11 coprono i temi tradizionali di un corso introduttivo alla programmazione. I capitoli 1-10 contengono anche un breve paragrafo *facoltativo* “**Introduzione alla data science**”, dove introdurremo l’intelligenza artificiale, le statistiche descrittive di base, le misure di tendenza centrale e di dispersione, la simulazione, le visualizzazioni statiche e dinamiche, l’elaborazione dei file CSV, l’esplorazione e la manipolazione dei dati con la libreria Pandas, le serie temporali e la regressione lineare semplice. Tutto questo vi preparerà ai casi di studio sulla data science, sull’IA, sui big data e sul cloud che troverete nei Capitoli 12-17, dove vi daremo l’opportunità di utilizzare **dataset del mondo reale**.

Dopo aver coperto i Capitoli 1-5 e alcune parti fondamentali dei Capitoli 6-7, sarete in grado di affrontare una buona parte dei **casi di studio sulla data science, l’IA e i big data** che sono presentati nei Capitoli 12-17, in base a ciò che è più appropriato per il vostro corso di programmazione:

- I corsi di informatica lavoreranno di più sui Capitoli 1-11 e su alcuni paragrafi “Introduzione alla data science” presenti nei Capitoli 1-10. I docenti di informatica includeranno alcuni casi di studio dei Capitoli 12-17.
- I corsi di data science lavoreranno meno sui Capitoli 1-11, affrontando la maggior parte dei paragrafi “Introduzione alla data science” presenti nei Capitoli 1-10 e tutti, o quasi, i casi di studio dei Capitoli 12-17.

Il paragrafo “Dipendenze tra i capitoli” di questa Prefazione aiuterà i docenti a pianificare i loro corsi sfruttando l’architettura modulare del libro.

I Capitoli 12-17 sono ricchi di argomenti interessanti, potenti e innovativi. Vengono presentate implementazioni di casi di studio su argomenti come il machine learning supervisionato, il machine learning non supervisionato, il deep learning e l’apprendimento con rinforzo (negli esercizi), il Natural Language Processing, il data mining, i big data e altro ancora. Durante questo percorso, acquisirete un’**ampia competenza** dei termini e dei concetti della data science, passando da brevi definizioni all’utilizzo dei concetti appresi in programmi piccoli, medi e grandi. Dare un’occhiata al dettagliato Indice analitico del libro vi darà un’idea dell’ampiezza degli argomenti coperti.

## A chi è destinato questo libro

L’architettura modulare di questo libro lo rende adatto a diversi gruppi di lettori:

- **Studenti dei corsi di laurea in informatica.** Prima di tutto, il nostro libro è un’introduzione solida e moderna al linguaggio Python. Le raccomandazioni di ACM/IEEE (le due principali associazioni di informatici e ingegneri) per i corsi di laurea in informatica elencano cinque tipi di percorsi: ingegneria

2. Guo, Philip. “Python Is Now the Most Popular Introductory Teaching Language at Top U.S. Universities”, ACM, 7 luglio 2014, <https://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-u-s-universities/fulltext>.

3. <https://www.oreilly.com/ideas/5-things-to-watch-in-python-in-2017>.

4. <https://www.kdnuggets.com/2017/08/python-overtakes-r-leader-analytics-data-science.html>.

5. <https://www.r-bloggers.com/data-science-job-report-2017-r-passes-sas-but-python-leaves-them-both-behind/>.

6. <https://www.oreilly.com/ideas/5-things-to-watch-in-python-in-2017>.

informatica, informatica, sistemi informativi, tecnologie dell'informazione e ingegneria del software.<sup>7</sup> Il libro è appropriato per ognuno di essi.

- **Corsi di laurea triennale in data science.** Questo libro è utile anche in molti corsi di data science, in particolare per quelli introduttivi, dato che segue le raccomandazioni per **integrare tutte le aree fondamentali di ogni corso**. In un percorso di data science, questo libro può essere usato come libro di testo principale per il primo corso di informatica o di data science, per poi essere usato come libro di riferimento su Python per il resto del percorso.
- **Corsi delle lauree magistrali in data science.** Questo libro può essere usato come libro di testo principale per il primo corso, per poi essere usato come libro di riferimento su Python nei successivi corsi sulla data science.
- **Altri corsi di laurea.** Questo libro può essere usato in corsi di servizio per studenti di corsi di laurea diversi da informatica o data science.
- **Istituti tecnici superiori.** Questa tipologia di scuole tenderà a offrire corsi per preparare gli studenti a programmi di studio sulla data science. Questo libro può essere adatto allo scopo.
- **Scuole medie superiori.** Così come si iniziò a insegnare informatica in risposta all'interesse degli studenti, molti docenti stanno ora iniziando a insegnare Python e a introdurre la data science.<sup>8</sup> Secondo un recente articolo apparso su LinkedIn, “la data science dovrebbe essere insegnata alle scuole superiori”, dove il “percorso dovrebbe rispecchiare le tipologie di carriere che i nostri ragazzi affronteranno, concentrandosi direttamente verso la direzione presa dai lavori e dalle tecnologie”.<sup>9</sup>
- **Corsi di aggiornamento aziendali e corsi professionalizzanti.**

## Caratteristiche fondamentali

### Semplicità, brevità, attualità

- **Semplicità.** In ogni aspetto del libro ci siamo sforzati di essere **semplici e chiari**. Per esempio, quando esistono molte librerie per svolgere un determinato compito, abbiamo sempre scelto la più semplice.
- **Brevità.** La maggior parte dei 538 esempi del libro è breve, con poche linee di codice, per avere un immediato riscontro in IPython. Esempi più lunghi sono stati necessari in circa 40 script per poter fornire casi di studio completi.
- **Attualità.** Abbiamo letto moltissimi libri e testi professionali sulla programmazione Python e sulla data science, oltre a guardare circa 15.000 articoli, video, post su blog, articoli di ricerca, post su forum e parti di documentazioni. Questo ci ha permesso di “avere il polso” delle comunità che gravitano attorno a Python, all'informatica, alla data science, all'IA, ai big data e al cloud, e di creare così 1.566 esempi, esercizi e progetti (EEP) di estrema attualità.

### Riscontro immediato, esplorazione, scoperta e pedagogia sperimentale con IPython

- La maniera ideale per imparare da questo libro è di leggere ed eseguire gli esempi di codice in parallelo. In tutto il libro useremo l'**interprete IPython**, che mette a disposizione una modalità interattiva facile e con riscontro immediato, così da permettere una veloce esplorazione, scoperta e sperimentazione di Python e delle sue numerose librerie.
- La maggior parte del codice viene presentata in **piccole sessioni interattive di IPython**. Ogni frammento di codice che scriverete verrà letto immediatamente da IPython, che lo valuterà e ne visualizzerà i risultati. Questo **riscontro immediato** permette di mantenere viva l'attenzione, di incoraggiare l'apprendimento, di facilitare la prototipizzazione rapida e di velocizzare il processo di sviluppo del software.

7. <https://www.acm.org/education/curricula-recommendations>.

8. <http://datascience.la/introduction-to-data-science-for-high-school-students/>.

9. <https://www.linkedin.com/pulse/data-science-should-taught-high-school-rebecca-croucher/>.

- Nei nostri libri utilizziamo sempre un **approccio didattico diretto al codice**, concentrandoci su *programmi completi e funzionanti con esempi di input e output*. La “magia” di IPython è data dal fatto che “rende vivi” i frammenti di codice all’inserimento di ogni singola riga. Tutto ciò incentiva l’apprendimento e incoraggia l’esplorazione.
- Usare IPython è un buon modo per imparare i messaggi associati agli errori più comuni. Faremo errori intenzionalmente per mostrarvi cosa succede. Quando vi diremo che c’è un errore, provate a vedere cosa accade.
- Utilizziamo la stessa filosofia di riscontro immediato nei **557 esercizi di autovalutazione** (ideali nelle “flipped classroom”) e in molti dei 471 esercizi e progetti di fine capitolo presenti nel libro.

### Fondamenti di programmazione in Python

- Per prima cosa questo è un libro di testo introduttivo a Python. Forniremo un’ampia copertura dei fondamenti della programmazione in Python e in generale.
- Discuteremo i modelli di programmazione in Python: **procedurale, funzionale e orientata agli oggetti**.
- Daremo particolare importanza alla **risoluzione dei problemi** e allo **sviluppo di algoritmi**.
- Utilizzeremo le migliori pratiche per **preparare gli studenti al lavoro**.
- Utilizzeremo la **programmazione funzionale** in maniera appropriata attraverso tutto il libro. Una tabella nel Capitolo 4 elenca le caratteristiche fondamentali della programmazione funzionale in Python, indicando i capitoli in cui verranno spiegate.

### 538 esempi, 471 esercizi e progetti (EEP)

- Gli studenti useranno un approccio concreto per imparare da un’ampia selezione di **esempi, esercizi e progetti (EEP)** tratti dalle applicazioni dell’informatica, della data science e di molti altri campi nel **mondo reale**.
- I **538 esempi** partono da singoli frammenti di codice fino ad arrivare a casi di studio completi di informatica, data science, intelligenza artificiale o big data.
- I **471 esercizi e progetti** sono estensioni naturali degli esempi contenuti nei capitoli. Ogni capitolo si chiude con un considerevole numero di esercizi a copertura di un’ampia varietà di tematiche. Questo aiuterà i docenti a ritagliare i loro corsi sulle particolari richieste della propria platea e a variare i compiti dei corsi a ogni semestre.
- Gli EEP daranno un’introduzione coinvolgente, impegnativa e interessante alla programmazione in Python, alla IA concreta, all’informatica e alla data science.
- Gli studenti affronteranno sfide emozionanti e interessanti con tecnologie di **IA, big data e cloud** come **NLP, data mining, machine learning, deep learning, Hadoop, MapReduce, Spark, IBM Watson**, con le librerie fondamentali per la data science (**NumPy, Pandas, SciPy, NLTK, TextBlob, spaCy, BeautifulSoup, Textatistic, Tweepy, Scikit-learn, Keras**) e con le librerie di visualizzazione (**Matplotlib, Seaborn, Folium**).
- I nostri EEP vi incoraggeranno a pensare al futuro. Scrivendo la Prefazione abbiamo avuto questa idea (anche se non è presente nel testo, ci sono altri progetti altrettanto stimolanti): grazie al **deep learning**, all’**Internet delle cose** e alla grande quantità di telecamere puntate sugli eventi sportivi, diventerà possibile mantenere *statistiche automatiche*, rivedere i dettagli di ogni partita e risolvere i problemi della moviola in campo in maniera immediata. Gli appassionati non dovranno più sopportare errori arbitrari o ritardi tipici dei moderni eventi sportivi. Ecco un’altra idea: potremmo usare queste tecnologie per eliminare gli arbitri. Perché no? Stiamo continuamente affidando le nostre vite ad altre tecnologie basate sul deep learning, come i **chirurghi robotici** o le **macchine a guida autonoma!**
- I **progetti negli esercizi** vi incoraggeranno ad approfondire ciò che avrete imparato e a effettuare ricerche su tecnologie che non abbiamo trattato. Questi progetti sono spesso ad ampio raggio e potrebbero richiedere ricerche e sforzi implementativi considerevoli.



- Vi incoraggiamo a guardare le tantissime **demo** e gli esempi di codice **open source** (disponibili su siti come **GitHub**) per trarre ispirazione per ulteriori **progetti e ricerche per tesi**.

### 557 esercizi e risposte di autovalutazione

- La maggior parte dei paragrafi termina con una media di tre **esercizi di autovalutazione**.
- Le **tipologie di autovalutazione** vi permetteranno di testare la vostra comprensione dei concetti studiati.
- Le **autovalutazioni interattive con IPython** vi daranno la possibilità di rinforzare le tecniche di programmazione appena imparate.
- Per un apprendimento rapido, gli esercizi di autovalutazione sono seguiti immediatamente dalle loro risposte.

### Preferire spiegazioni in linguaggio corrente al linguaggio matematico

- Gli argomenti di data science possono avere un alto contenuto matematico. Dato che questo libro verrà usato nei primi corsi di informatica o di data science, dove gli studenti ancora non hanno profonde conoscenze matematiche, abbiamo evitato la matematica più complessa, che può essere trattata in corsi più avanzati.
- Abbiamo catturato la sostanza concettuale della matematica per metterla al lavoro nei nostri esempi, esercizi e progetti. L'abbiamo fatto usando le **librerie di Python** che nascondono le complessità matematiche, come **statistics**, **NumPy**, **SciPy**, **Pandas** e molte altre. In questo modo, per gli studenti, sarà facile beneficiare di tecniche matematiche come la **regressione lineare** senza dover necessariamente conoscere la matematica che ci sta dietro. Negli esempi sul **machine learning** e il **deep learning**, ci concentreremo sulla creazione di oggetti che svolgeranno le operazioni matematiche per noi “dietro le quinte”. Questo è uno dei principi della **programmazione basata sugli oggetti**. È come guidare una macchina in maniera sicura verso la propria destinazione senza conoscere la matematica, l'ingegneria e la scienza necessarie alla costruzione dei motori, delle trasmissioni, del volante o dei sistemi di frenata.

### Visualizzazioni

- **67 visualizzazioni a colori, statiche, dinamiche, animate e interattive, bidimensionali o tridimensionali** (tabelle, grafici, immagini, animazioni ecc.) vi aiuteranno nella comprensione dei concetti.
- Ci concentreremo su visualizzazioni di alto livello prodotte da **Matplotlib**, **Seaborn**, **Pandas** e **Folium** (per le **mappe interattive**).
- Useremo le visualizzazioni come strumento pedagogico. Per esempio, renderemo “viva” la **legge dei grandi numeri** attraverso un **lancio di dadi simulato** e un grafico a barre dinamico. All'aumentare dei lanci, vedrete le percentuali associate alle facce dei dadi avvicinarsi gradualmente a 16,667% (1/6), mentre le altezze delle barre che rappresentano le percentuali si equalizzeranno.
- Dovrete imparare a conoscere i vostri dati. Un modo per farlo è quello di guardare semplicemente ai dati grezzi. Così facendo, anche per modeste quantità di dati, potreste rapidamente perdervi nei dettagli. Le visualizzazioni sono cruciali nell'esplorazione dei big data, dove possiamo avere miliardi o più di oggetti, per **esplorarli** e per **comunicare i risultati delle ricerche in maniera riproducibile**. Secondo un modo di dire, un'immagine vale più di mille parole:<sup>10</sup> nei **big data**, una visualizzazione potrebbe valere quanto miliardi di oggetti, o anche più, in un database.
- Vi capiterà di avere bisogno di “volare in alto come un aereo” per avere una visione “ad ampio raggio” dei dati. Le **statistiche descrittive** vi potranno aiutare, ma possono essere fuorvianti. Il quartetto di Anscombe, che studierete negli esercizi, dimostra attraverso le visualizzazioni che dataset *molto diversi* possono avere statistiche descrittive *quasi identiche*.
- Mostriamo il codice delle visualizzazioni e delle animazioni, così che possiate poi crearne di vostre. Metteremo a disposizione anche i file sorgenti e i notebook Jupyter delle animazioni, così che possiate

10. [https://en.wikipedia.org/wiki/A\\_picture\\_is\\_worth\\_a\\_thousand\\_words](https://en.wikipedia.org/wiki/A_picture_is_worth_a_thousand_words).

personalizzarne comodamente il codice e i parametri, per rieseguirle e vedere gli effetti delle vostre modifiche.

- In molti esercizi vi verrà chiesto di creare le vostre visualizzazioni.

### Esperienze con i dati

- La proposta di percorso di studio in data science recita che “Le **esperienze con i dati** devono giocare un ruolo fondamentale in tutti i corsi”.<sup>11</sup>
- Negli esercizi, esempi e progetti del libro, lavorerete con molti **dataset del mondo reale** e con diverse **fonti di dati**. Esiste un’ampia varietà di **dataset open e gratuiti** disponibili online per i vostri esperimenti. Alcuni dei siti a cui ci riferiremo elencano centinaia o migliaia di dataset. Vi incoraggiamo a esplorarli.
- Abbiamo analizzato centinaia di programmi didattici, rintracciato le **preferenze dei docenti sui vari dataset** e ricercato i dataset più popolari negli studi di **machine learning supervisionato, machine learning non supervisionato e deep learning**. Molte delle librerie che userete includono dataset per le vostre sperimentazioni.
- Imparerete i passi necessari per ottenere i dati e prepararli per l’analisi, per analizzare i dati usando diverse tecniche, per regolare i vostri modelli e per comunicare i vostri risultati in maniera efficace, specialmente con l’utilizzo delle visualizzazioni.

### Pensare come uno sviluppatore

- Lavorerete con l’**ottica di uno sviluppatore**, usando siti come **GitHub** e **StackOverflow** e facendo molte ricerche online. I nostri **paragrafi “Introduzione alla data science”**, uniti ai casi di studio nei Capitoli 12-17, forniranno una ricca esperienza di utilizzo dei dati.
- **GitHub** è una sede eccellente per **trovare codice sorgente open source** da incorporare nei vostri progetti (e per contribuire con il vostro codice alla **comunità open source**). Inoltre, è un elemento fondamentale nell’arsenale di uno sviluppatore software, grazie agli **strumenti di controllo della versione**, che aiutano gli sviluppatori a gestire i propri progetti.
- Vi incoraggiamo a studiare il codice degli sviluppatori su siti come GitHub.
- Per prepararvi a una carriera nell’informatica o nella data science, userete una straordinaria gamma di **librerie Python** e di data science gratuite e open source, e di **dataset reali** gratuiti e open forniti da governi, aziende e università.

### Casi di studio sull’intelligenza artificiale

- Per quale motivo questo libro non ha un capitolo dedicato all’intelligenza artificiale? Nei casi di studio dei Capitoli 12-16 presenteremo tematiche di **intelligenza artificiale** (intersezione fondamentale tra informatica e data science), come l’**elaborazione del linguaggio naturale**, l’**estrazione dati da Twitter per svolgere l’analisi del sentiment**, il **calcolo cognitivo con Watson di IBM**, il **machine learning supervisionato e non supervisionato**, il **deep learning** e l’**apprendimento con rinforzo** (negli esercizi). Il Capitolo 17 (online) presenterà le infrastrutture hardware e software per i big data, quelle che permettono agli informatici e agli esperti di data science di implementare soluzioni all’avanguardia basate sull’IA.

### Informatica

- L’elaborazione dei fondamenti di Python nei Capitoli 1-10 vi permetterà di imparare a pensare come un informatico. Il Capitolo 11 vi darà una prospettiva ancora più avanzata sui temi classici dell’informatica. In tale capitolo si evidenzieranno i problemi riguardanti le prestazioni.

---

11. “Curriculum Guidelines for Undergraduate Programs in Data Science”, <http://www.annualreviews.org/doi/full/10.1146/annurev-statistics-060116-053930> (p. 18).

## Collezioni integrate: liste, tuple, insiemi e dizionari

- Per molti sviluppatori, oggi, ha poco senso costruire strutture dati *personalizzate*. Queste sono l'argomento di corsi avanzati d'informatica, mentre il nostro obiettivo è strettamente limitato ai corsi introduttivi di informatica o di data science. Nel libro ci sono **due capitoli** che contengono un accurato trattamento delle **strutture dati integrate in Python: liste, tuple, dizionari e insiemi**, con le quali si possono svolgere la maggior parte dei compiti.

## Programmazione orientata ai vettori con NumPy e gli oggetti Series e DataFrame di Pandas

- Useremo un approccio innovativo in questo libro, concentrandoci su tre strutture dati fondamentali prese da librerie open source: i vettori di NumPy e gli oggetti `Series` e `DataFrame` di Pandas. Queste librerie sono usate in maniera massiccia nella data science, nell'informatica, nell'intelligenza artificiale e nei big data. NumPy offre prestazioni superiori di due ordini di grandezza rispetto alle liste integrate in Python.
- Nel Capitolo 7 tratteremo estensivamente i vettori di NumPy. Molte librerie, come Pandas, sono costruite a partire da NumPy. I **paragrafi "Introduzione alla data science"** nei Capitoli 7-9 introducono gli oggetti `Series` e `DataFrame` di Pandas, i quali, insieme ai vettori di NumPy, verranno poi usati nei capitoli successivi.

## Elaborazione e serializzazione dei file

- Il Capitolo 9 presenta l'**elaborazione dei file di testo**, per poi passare alla serializzazione degli oggetti nel diffuso formato **JSON (JavaScript Object Notation)**. JSON è un formato di scambio dati molto comune, che incontreremo frequentemente nei capitoli sulla data science, spesso all'interno di librerie che ne nascondono i dettagli per semplicità.
- Molte librerie per la data science mettono a disposizione funzionalità integrate di elaborazione file, così da poter caricare i dataset nei vostri programmi Python. Oltre ai file in testo puro, elaboreremo file nel **formato CSV**, usando il modulo `csv` della Libreria Standard di Python e anche funzionalità della libreria Pandas.

## Programmazione basata sugli oggetti

- In tutto il codice Python che abbiamo studiato per preparare questo libro, raramente abbiamo trovato *classi personalizzate*, che sono comunemente utilizzate all'interno delle potenti librerie usate dai programmatori Python.
- Vogliamo sottolineare l'utilizzo dell'enorme quantità di preziose classi che la **comunità open source di Python** ha impacchettato in librerie diventate standard industriali. Dovrete conoscere le diverse librerie esistenti, scegliendo quelle necessarie per le vostre applicazioni, creando oggetti da classi esistenti (tipicamente in una o due righe di codice) e utilizzando queste classi per i vostri scopi. Tutto ciò prende il nome di **programmazione basata sugli oggetti** e fa parte dell'attrattività di Python, perché permette di **costruire impressionanti applicazioni in maniera veloce e concisa**.
- Grazie a questo approccio sarete in grado di usare il machine learning, il deep learning, l'apprendimento con rinforzo (negli esercizi) e altre tecnologie di IA per risolvere un'ampia gamma di problemi interessanti, incluse le sfide del **calcolo cognitivo** come il **riconoscimento vocale** e la **visione artificiale**. In passato, seguendo solo un corso di programmazione introduttiva, non sareste stati in grado di affrontare questi compiti.

## Programmazione orientata agli oggetti

- Per gli studenti di informatica, lo sviluppo di classi *personalizzate* è un'abilità fondamentale della **programmazione orientata agli oggetti**, così come l'ereditarietà, il polimorfismo e il duck typing. Li tratteremo nel Capitolo 10.
- Il nostro trattamento della programmazione orientata agli oggetti è modulare, quindi i docenti potranno scegliere una copertura di base o intermedia.

- Nel Capitolo 10 viene presentato il testing di unità con `doctest` e una divertente simulazione di mescolatura e distribuzione di carte da gioco.
- I sei capitoli sulla data science, l'IA, i big data e il cloud richiederanno solamente alcune brevi definizioni di classi personalizzate. I docenti che decideranno di non usare il Capitolo 10, potranno far copiare queste definizioni ai propri studenti.

### Privatezza

- Negli esercizi, vedrete le sempre più severe leggi sulla privatezza dei dati, come la **HIPAA (*Health Insurance Portability and Accountability Act*)** negli Stati Uniti e la **GDPR (*General Data Protection Regulation*)** nell'Unione Europea (vedi Capitolo 17, online). Un aspetto fondamentale della privacy è la protezione delle **informazioni di identificazione personale (PII)**, mentre una delle problematiche primarie dei big data è la facilità con cui si possono incrociare fatti riguardanti le persone guardando in diversi database. Vedremo problemi legati alla privacy in diversi punti del libro.

### Sicurezza

- La sicurezza è cruciale per il rispetto della privacy. Vedremo alcune problematiche relative alla sicurezza specifiche di Python.
- L'IA e i big data presentano questioni etiche uniche per la privacy e la sicurezza. Negli esercizi, verrà chiesto agli studenti di cercare il **progetto di sicurezza OWASP** (<http://www.pythonsecurity.org/>), il **rilevamento di anomalie**, la **blockchain** (la tecnologia che sta dietro alle criptovalute come i Bitcoin o Ethereum) e altro ancora.

### Etica

- Dilemma etico: supponiamo che grazie all'analisi dei big data e all'IA si preveda ci sia una grossa probabilità che una persona senza precedenti criminali stia per commettere un grave crimine. Questa persona dovrebbe essere arrestata? Negli esercizi farete ricerche su questo e su altri problemi etici, come i *deep fake* (immagini e video generati dall'IA che però appaiono reali), i *pregiudizi* nel machine learning e l'*editing genetico CRISPR*. Gli studenti ricercheranno anche i problemi etici e di privacy riguardanti l'IA e gli **assistenti intelligenti**, come **Watson di IBM**, **Alexa di Amazon**, **Siri di Apple**, **Cortana di Microsoft** e l'**assistente di Google**. Per esempio, recentemente, un giudice americano ha ordinato ad Amazon di consegnare le registrazioni di Alexa per poterle usare in un processo.<sup>12</sup>

### Riproducibilità

- Nella scienza in generale e nella data science in particolare, ci deve essere la possibilità di riprodurre i risultati degli esperimenti e degli studi per poterli comunicare in maniera efficace. I **notebook Jupyter** sono il mezzo preferito per questo scopo.
- Mettiamo a disposizione diversi notebook Jupyter per aiutarvi a soddisfare le richieste di riproducibilità dei percorsi di studio in data science.
- Nel libro, parleremo della *riproducibilità* nel contesto delle tecniche di programmazione e di software come Jupyter e **Docker**.

### Trasparenza

- Nella proposta di percorso formativo in data science si parla di trasparenza dei dati. Uno degli aspetti della trasparenza è dato dalla disponibilità dei dati. Molti governi e organizzazioni aderiscono ai principi delle infrastrutture **open** per i dati, permettendo a chiunque di accedere ai loro dati.<sup>13</sup> Segneremo un'ampia gamma di dataset resi disponibili da queste entità.

---

12. <https://techcrunch.com/2018/11/14/amazon-echo-recordings-judge-murder-case>.

13. [https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Big%20data%20The%20next%20frontier%20for%20innovation/MGI\\_big\\_data\\_full\\_report.ashx](https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Big%20data%20The%20next%20frontier%20for%20innovation/MGI_big_data_full_report.ashx) (pagina 56).

- Determinare la correttezza dei dati e conoscere le loro origini (pensate, per esempio, alle “fake news”), sono altri aspetti della trasparenza. Molti dei dataset che andremo a usare sono inclusi nelle librerie che presenteremo, come **Scikit-learn** per il machine learning e **Keras** per il deep learning. Vi indicheremo anche diversi **archivi di dataset**, come il **Machine Learning Repository** (con più di 450 dataset) della **University of California Irvine (UCI)**<sup>14</sup> e il **StatLib Datasets Archive** (con più di 100 dataset) della **Carnegie Mellon University**.<sup>15</sup>

## Prestazioni

- Utilizzeremo lo **strumento di profilazione timeit** in molti esempi ed esercizi, per confrontare le prestazioni dei diversi approcci che useremo per affrontare un determinato compito. Faremo altre osservazioni sulle prestazioni quando vedremo le espressioni generatrici, i vettori di NumPy rispetto alle liste di Python, i modelli di machine learning, quelli di deep learning e il calcolo distribuito con Hadoop e Spark.

## Big data e parallelismo

- Le applicazioni per computer, in generale, vanno bene per fare una cosa alla volta. Le applicazioni più moderne hanno necessità di fare molte cose in parallelo. Si crede che la mente umana posseda l'equivalente di 100 miliardi di processori paralleli.<sup>16</sup> Per anni abbiamo scritto di parallelismo a livello dei programmi, una materia complessa e soggetta a errori.
- In questo libro, invece di farvi scrivere codice per la parallelizzazione, userete librerie come Keras costruite su TensorFlow, o strumenti per i big data come Hadoop e Spark per parallelizzare le operazioni. In questa epoca di big data e IA, le enormi richieste di elaborazione delle applicazioni che lavorano sui dati si devono avvantaggiare del vero parallelismo messo a disposizione dai **processori multicore**, dai **processori grafici (graphics processing unit, GPU)**, dalle **unità di elaborazione tensoriale (tensor processing unit, TPU)** e dagli enormi **cluster di computer nel cloud**. Alcune operazioni sui big data potrebbero necessitare di migliaia di processori che lavorano in parallelo per poter analizzare un'enorme quantità di dati in un tempo ragionevole. Sequenzializzare un tale processo non è un'opzione praticabile, ci vorrebbe troppo tempo.

## Dipendenze tra i capitoli

Se siete un docente che deve pianificare un corso, o un professionista che deve decidere quali capitoli leggere, in questo paragrafo vi aiuteremo a fare le scelte più adeguate. Leggete il **Sommario** per familiarizzare velocemente con il contenuto del libro. Seguire i capitoli nell'ordine in cui sono presentati è l'ideale. Tuttavia, la maggior parte del contenuto nei paragrafi “Introduzione alla data science” che si trovano alla fine dei Capitoli 1-10 e nei casi di studio dei Capitoli 12-17 richiede solamente la lettura dei Capitoli 1-5 e di una piccola parte dei Capitoli 6-10 come vedremo qui di seguito.

## Parte 1: Fondamenti di Python

**Raccomandiamo che tutti i corsi affrontino il contenuto dei Capitoli 1-5.**

- **Capitolo 1, “Introduzione ai computer e a Python”**. Introduce i concetti di base per poter affrontare la programmazione in Python nei Capitoli 2-11 e i casi di studio su big data, intelligenza artificiale e cloud contenuti nei Capitoli 12-17. Il capitolo contiene anche **test guidati su IPython e notebook Jupyter**.
- **Capitolo 2, “Introduzione alla programmazione Python”**. Presenta i fondamenti della programmazione in Python, con esempi di codice per illustrare le funzionalità essenziali.
- **Capitolo 3, “Istruzioni di controllo e sviluppo dei programmi”**. Presenta le istruzioni di controllo di Python, focalizzandosi sullo **sviluppo degli algoritmi e la risoluzione dei problemi**, e introducendo infine l'**elaborazione basilare delle liste**.

14. <https://archive.ics.uci.edu/ml/datasets.php>.

15. <http://lib.stat.cmu.edu/datasets/>.

16. <https://www.technologyreview.com/s/532291/fmri-data-reveals-the-number-of-parallel-processes-running-in-the-brain/>.

- **Capitolo 4, “Funzioni”**. Introduce la costruzione dei programmi con l’utilizzo di funzioni già esistenti e di quelle personalizzate; verranno presentate **tecniche di simulazione** con la **generazione casuale dei numeri** e si introdurranno le **basi dell’elaborazione sulle tuple**.
- Il **Capitolo 5, “Sequenze: liste e tuple”**. Presenta con maggior dettaglio le collezioni di liste e tuple integrate in Python; nel capitolo si comincia anche a presentare la **programmazione in stile funzionale**.

## Parte 2: Strutture dati, stringhe e file in Python<sup>17</sup>

Qui di seguito riassumiamo le dipendenze tra i Capitoli 6-9, assumendo che abbiate letto i Capitoli 1-5.

- **Capitolo 6, “Dizionari e insiemi”**. Il paragrafo “Introduzione alla data science” non dipende dal contenuto del Capitolo 6.
- **Capitolo 7, “Programmazione orientata ai vettori con NumPy”**. Il paragrafo “Introduzione alla data science” richiede la conoscenza dei dizionari (Capitolo 6) e dei vettori (Capitolo 7).
- **Capitolo 8, “Stringhe: un approfondimento”**. Il paragrafo “Introduzione alla data science” richiede la conoscenza delle stringhe raw e delle espressioni regolari (Paragrafi 8.11-8-12), e degli oggetti `Series` e `DataFrame` di Pandas contenuti nel Paragrafo 7.14.
- **Capitolo 9, “File ed eccezioni”**. Per la **serializzazione JSON** è utile aver capito i fondamenti dei dizionari (Paragrafo 6.2). Inoltre, il paragrafo “Introduzione alla data science” richiede la conoscenza della funzione integrata `open` e dell’istruzione `with` (Paragrafo 9.3), in aggiunta alle funzionalità degli oggetti `DataFrame` contenute nel Paragrafo 7.14.

## Parte 3: Argomenti avanzati su Python

Qui di seguito riassumiamo le dipendenze tra i Capitoli 10-11, assumendo che abbiate letto i Capitoli 1-5.

- **Capitolo 10, “Programmazione orientata agli oggetti”**. Il paragrafo “Introduzione alla data science” richiede la conoscenza delle funzionalità degli oggetti `DataFrame` descritte nel Paragrafo 7.14. I docenti che vogliono affrontare solo le **classi e gli oggetti** possono usare i Paragrafi 10.1-10.6. I docenti che vogliono affrontare anche argomenti più avanzati, come l’**ereditarietà**, il **polimorfismo** e il **duck typing**, possono usare i Paragrafi 10.7-10.9. I Paragrafi 10.10-10.15 mettono a disposizione ulteriori prospettive.
- **Capitolo 11, “Pensare come un informatico: ricorsione, ricerca, ordinamento e notazione O grande”**. Richiede il saper operare con gli elementi dei vettori (Capitolo 7), il comando magico `%timeit` (Paragrafo 7.6), il metodo `join` delle stringhe (Paragrafo 8.9) e `FuncAnimation` di Matplotlib dal Paragrafo 6.4.

## Parte 4: Casi di studio su IA, cloud e big data

Qui di seguito riassumiamo le dipendenze tra i Capitoli 12-17, assumendo che abbiate letto i Capitoli 1-5. La maggior parte di questi capitoli richiede anche le basi sui dizionari contenute nel Paragrafo 6.2.

- **Capitolo 12 (online), “Natural Language Processing (NLP)”**. Utilizza le funzionalità degli oggetti `DataFrame` di Pandas del Paragrafo 7.14.
- **Capitolo 13 (online), “Data Mining Twitter”**. Utilizza le funzionalità degli oggetti `DataFrame` di Pandas (Paragrafo 7.14), il metodo `join` delle stringhe (Paragrafo 8.9), le basi di JSON (Paragrafo 9.5), `TextBlob` (Paragrafo 12.2) e le nuvole di parole (Paragrafo 12.3). Molti esempi richiedono la definizione di una classe usando l’ereditarietà (Capitolo 10), ma i lettori possono semplicemente replicare le definizioni delle classi che mettiamo a disposizione senza leggere il Capitolo 10.
- **Capitolo 14 (online), “IBM Watson and Cognitive Computing”**. Utilizza la funzione integrata `open` e l’istruzione `with` (Paragrafo 9.3).
- **Capitolo 15, “Machine learning: classificazione, regressione e clustering”**. Utilizza le basi dei vettori di NumPy e il metodo `unique` (Capitolo 7), le funzionalità degli oggetti `DataFrame` di Pandas (Paragrafo 7.14) e la funzione `subplots` di Matplotlib (Paragrafo 10.6).

17. Avremmo potuto includere il Capitolo 5 nella Parte 2. L’abbiamo inserito nella Parte 1 perché rientra nell’insieme di capitoli che dovrebbe essere affrontato in tutti i corsi.