PAUL DEITEL
HARVEY DEITEL

python™

Introduzione a Python®

Per l'informatica e la data science

A cura di Pietro Codara e Carlo Mereghetti

MyLab Codice per accedere alla piattaforma

Pearson

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

ECO LIBRO Pearson

Pearson

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, 1st Edition by PAUL DEITEL; HARVEY DEITEL; HARVEY DEITEL, published by Pearson Education, Inc, publishing as Pearson, Copyright © 2020.

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)

9788891915924

Printed in Italy

1ª edizione: aprile 2021

Ristampa
00 01 02 03 04

Anno
21 22 23 24 25

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 inspirato i tuoi studenti a pensare oltre i limiti.

Harvey Deitel

Sommario

Prefazione

"C'è oro in quelle colline laggiù!"[1]

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" connetterà 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.[2] Attualmente Python è particolarmente popolare nell'educazione e nel calcolo scientifico[3] e ha recentemente sorpassato R come linguaggio più usato nella data science.[4, 5, 6]

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.[7] 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.[8] 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".[9]
- **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 arbitrali 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:[10] 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*.
- Mostreremo 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.

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".[11]
- 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, oggigiorno, 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.[12]

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.[13] Segnaleremo 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` (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)**[14] e il **StatLib Datasets Archive** (con più di 100 dataset) della **Carnegie Mellon University**.[15]

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 possegga l'equivalente di 100 miliardi di processori paralleli.[16] 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[17]

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.