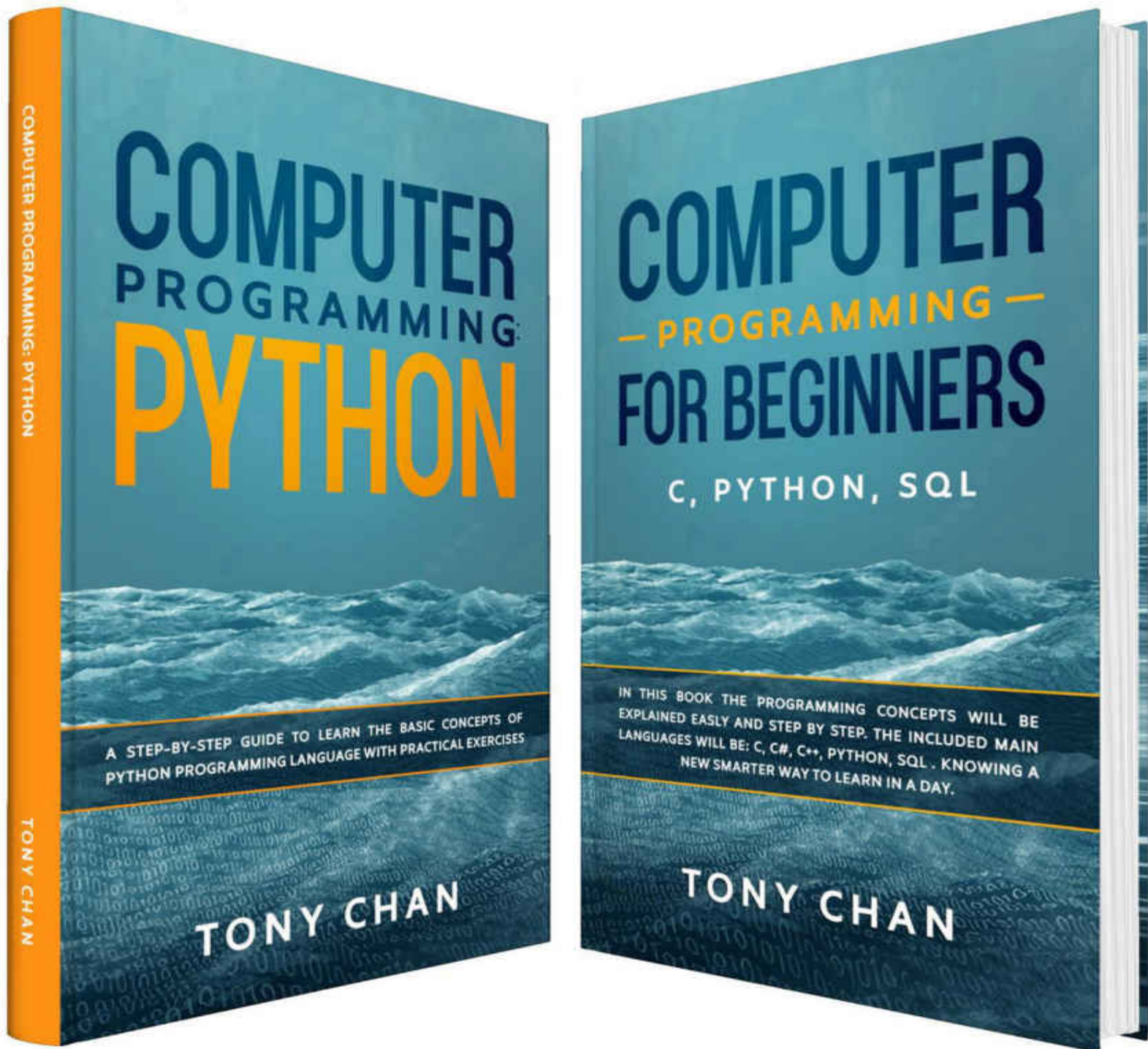


# Programmazione per Principianti: 2 libri in 1

Una guida passo-passo per imparare Python, C, SQL

in ITALIANO



# **Programmazione per principianti: 2 libri in 1**

*Una guida passo-passo per imparare C, SQL e Python*

## **LIBRO 1: Programmazione del Computer per principianti: C, Python, SQL**

*In questo libro, i concetti di programmazione saranno spiegati facilmente e passo dopo passo.  
Include i linguaggi principali: C, Python e SQL. Conoscerai un nuovo metodo per imparare in un  
giorno.*

*e*

## **LIBRO 2: Programmazione del Computer: PYTHON**

*Una guida passo-passo per apprendere i concetti di base del linguaggio di programmazione Python  
con esercizi pratici.*

***Di Tony Chan***

**© Copyright 2020 - Tutti i diritti riservati.**

Il contenuto in questo libro non può essere riprodotto, duplicato o trasmesso senza il diretto permesso scritto dell'autore o dell'editore.

In nessuna circostanza sarà attribuita alcuna colpa o responsabilità legale all'editore, o autore, per eventuali danni, riparazioni o perdite monetarie dovute alle informazioni contenute in questo libro.

**Avviso legale:**

Questo libro è protetto da copyright. Questo libro è solo per uso personale. Non è possibile modificare, distribuire, vendere, utilizzare, citare o parafrasare alcuna parte o il contenuto di questo libro senza il consenso dell'autore o dell'editore.

**Avviso di esclusione di responsabilità:**

Si prega di notare che le informazioni contenute in questo documento sono solo a scopo educativo e d'intrattenimento. È stato compiuto ogni sforzo

per presentare informazioni accurate, aggiornate e affidabili e complete. Nessuna garanzia di alcun tipo è dichiarata o implicita. I lettori riconoscono che l'autore non s'impegna a fornire consulenza legale, finanziaria, medica o professionale. Il contenuto di questo libro è stato derivato da varie fonti. Consultare un professionista autorizzato prima di tentare qualsiasi tecnica descritta in questo libro.

Leggendo questo documento, il lettore accetta che in nessun caso l'autore è responsabile per eventuali perdite, dirette o indirette, subite a seguito dell'uso delle informazioni contenute in questo documento, incluse omissioni o inesattezze.

# **LIBRO 1: Programmazione del Computer per Principianti: C, Python, SQL**

## **Sommario**

[introduzione](#)

## Capitolo 1: Le basi della programmazione

Perché vuoi programmare?

Scelta del linguaggio di codifica

Cerca corsi online

Concentrati maggiormente sul pensiero computazionale

Prova alcuni giochi di codifica

Guarda alcuni video sulla codifica

Realizza alcuni dei tuoi progetti

Celebrate le vostre piccole vittorie

Googlare Alcuni dei messaggi di errore

Conoscere un modo più intelligente per imparare a programmare in un giorno

Buone combinazioni linguistiche

Perché questi linguaggi di programmazione sono importanti da imparare?

## Capitolo 2: I diversi sistemi operativi

Sistema operativo Windows

Le applicazioni

Lavorare con Windows

Spegnimento del computer

I vantaggi dell'utilizzo di Windows

Sistema operativo Linux

Livello di esecuzione

I vantaggi di Linux

Sistema operativo Mac

[Il molo](#)

[I menu](#)

[Uno sguardo agli utenti](#)

[I riflettori](#)

[Il dashboard](#)

[I vantaggi del sistema operativo Mac](#)

## [Capitolo 3: I linguaggi C.](#)

[Uno sguardo a C](#)

[Ottenere l'IDE](#)

[Scrivere un codice fittizio](#)

[Scrittura di loop](#)

[Loop Do / While](#)

[Il ciclo annidato](#)

[Le variabili](#)

[Le funzioni C.](#)

[Uno sguardo al C ++](#)

[Creazione del nostro primo codice in C ++](#)

[Le variabili C ++](#)

[Gli array C ++](#)

[La dichiarazione Switch](#)

[In che modo C # è diverso](#)

[I tipi di dati C #](#)

[Creazione di una classe](#)

[Creazione di oggetti](#)

[I costruttori C #](#)

[I cicli C #](#)

## [Capitolo 4: Come programmare in Python](#)

[I vantaggi di Python](#)

[Creare la tua classe](#)

[Scrittura delle variabili](#)

[I loop Python](#)

[L'importanza delle dichiarazioni condizionali](#)

[Le eredità di Python](#)

## [Capitolo 5: Programmazione SQL](#)

[I vantaggi di SQL](#)

[Comandi comuni in SQL](#)

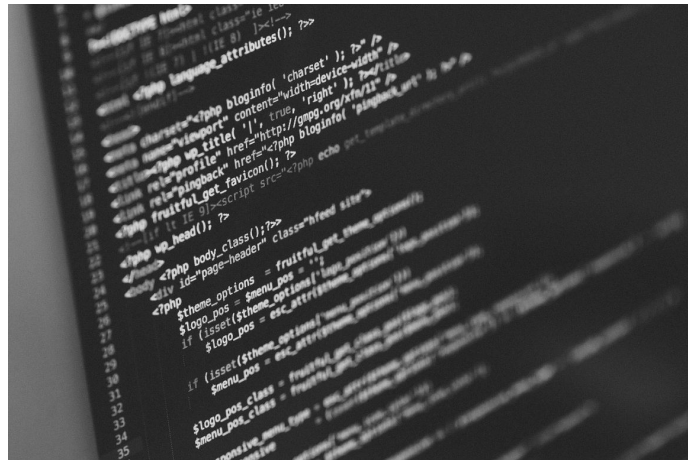
[Come creare una tabella](#)

[Modifica delle tabelle](#)

[Fare una query.](#)

[Conclusione](#)





# **LIBRO 2: Programmazione del Computer: PYTHON**

## **Sommario**

### [introduzione](#)

### [Capitolo 1: Introduzione a Python](#)

#### [1.1 Passaggi per imparare Python](#)

#### [1.2 Fondamenti di Python](#)

## **Capitolo 2: Gli attributi irresistibili di Python**

[2.1 Caratteristiche uniche della programmazione Python](#)

[2.2 Versioni di Python](#)

[2.3 Applicazioni di Python](#)

[2.4 Installazione di Python](#)

## **Capitolo 3: Compilatori, IDE e EDITOR di testo**

[3.1 Compilatori per la programmazione Python](#)

[3.2 IDE di Python](#)

[3.3 Interpreti Python](#)

[3.4 Editor di testo Python](#)

## **Capitolo 4: Confronto di Python con altri linguaggi**

[4.1 Confronto tra Python e C ++](#)

[4.2 Confronto tra Python e C #](#)

[4.3 Confronto tra Python e Java](#)

[4.4 Confronto tra Python e JavaScript](#)

[4.5 Confronto tra Python e Go Programming language](#)

[4.6 Confronto tra Python e Node.js](#)

[4.7 Confronto tra Python e PHP](#)

## **Capitolo 5: Variabili, operatori e tipi di dati di Python**

[5.1 Variabili di Python](#)

[5.2 Operatori in Python](#)

[5.3 Tipi di dati in Python](#)

## **Capitolo 6: Espressioni regolari, dichiarazioni di espressioni, cicli in Python**

[6.1 Espressioni regolari di Python](#)

[6.2 Espressioni di istruzioni di Python](#)

[6.3 Cicli in Python](#)

## **Capitolo 7: Funzioni Python e gestione dei file**

[7.1 Funzioni in Python](#)

[7.2 Gestione dei file di Python](#)

## **Capitolo 8: Concetti OOP in Python**

[8.1 Concetti di base OOP](#)

[8.2 Classe e oggetti Python](#)

[8.3 Python Constructor](#)

[8.4 Ereditarietà di Python](#)

## **Capitolo 9: Moduli, eccezioni e array in Python**

[9.1 Moduli in Python](#)

[9.2 Le eccezioni nella programmazione Python](#)

[9.3 Array Python](#)

## **Capitolo 10: Librerie in Python**

[10.1 Librerie generali Python](#)

[10.2 Librerie Python per la scienza dei dati](#)

## **Capitolo 11: Il futuro di Python**

[11.1 Opportunità di carriera associate a Python](#)

[11.2 Crescente popolarità di Python](#)

### 11.3 Fattori alla base della crescita di Python nel mondo moderno

#### Conclusione

#### Riferimenti

# ***Programmazione del Computer per Principianti: C, Python, SQL***

## **Introduzione**

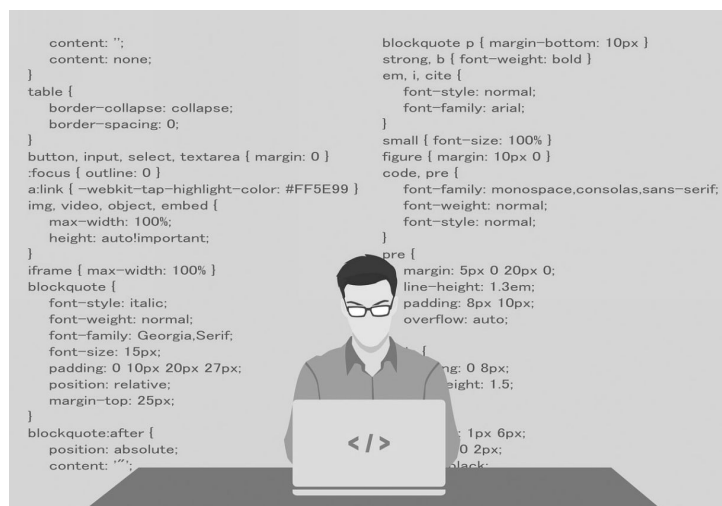
Congratulazioni per l'acquisto di: Computer Programming for Beginners: C, Python, SQL.

I capitoli seguenti parleranno di tutto ciò che sarà necessario conoscere per iniziare ad affacciarsi nel meraviglioso mondo della programmazione. C'è tanto da imparare e allo stesso tempo ci si apriranno tante possibilità, ma purtroppo molti sono spaventati dalla mole di cose da studiare e ovviamente immaginano curve di apprendimento molto lunghe. Con l'aiuto di questa guida, ci prenderemo il tempo per imparare non solo le basi della programmazione per computer, ma anche di più su come iniziare a lavorare su cinque diversi linguaggi di codifica in meno di una settimana, è un obiettivo ambizioso ma varrà la pena provarci.

Partiremo entrando più in dettaglio sulla programmazione del computer e su tutte le cose belle che saremo in grado di fare con essa. Esploreremo suggerimenti che ci semplificheranno la codifica, consigli su come utilizzare in cooperazione alcuni dei migliori linguaggi e persino suggerimenti per assicurarci di capire il modo migliore per imparare a programmare in pochissimo tempo.

Nel secondo capitolo ci prenderemo un momento per parlare dei tre principali sistemi operativi: Mac Os di Apple, Windows e Linux. Tutti e tre hanno vantaggi che li rendono un'ottima scelta per lavorarci, ma prima di capire quale sia il migliore per noi è importante sapere di più su ciò che vorrai ottenere con il tuo progetto e anche sul tuo obiettivo finale. Discuteremo su come questi sistemi operativi lavorano e cosa saremo in grado di fare con ciascuno di essi, inoltre faremo anche una panoramica su ciò che il futuro ha in serbo per noi, cioè sui nuovi software in progetto o che stanno per essere immessi nel mercato a breve.

Poi sarà il momento di entrare in alcune parti divertenti della programmazione e di cosa potremo fare in questa guida. Daremo prima uno sguardo al C, o meglio a: C, C ++ e C #. Questi sono tutti ottimi linguaggi imperativi di natura procedurale che condividono molte somiglianze, ma ci sono anche alcune grandi differenze, di cui parleremo in questa sezione. Vedremo come scrivere codici in ognuno, come gestire le variabili, le funzioni e altro in modo da poter iniziare a scrivere i nostri primi software in pochissimo tempo.



Dopo aver avuto la possibilità di imparare a utilizzare il C, è il momento di passare a lavorare con l'eccellente Python. Questo è un ottimo linguaggio su cui molti programmatori amano dedicare il loro tempo perché non solo è pensato per essere intuitivo, ma è rivolto sia al principiante, sia allo sviluppatore più esperto. Questo lo rende l'opzione perfetta per imparare in un giorno o meno e, con tutte le potenziali applicazioni, sarai in grado di farlo funzionare per le tue esigenze di codifica, non importa quali siano. Vedremo alcuni dei modi più comuni in cui puoi lavorare con Python, incluso quanto sia interessante come linguaggio OOP, cioè orientato agli oggetti, diverso dalla programmazione procedurale e strutturata, infatti nel primo si utilizzano gli oggetti software che essenzialmente comunicano tra di loro, mentre il secondo è un insieme di procedure o funzioni, nel terzo le procedure sono strutturate in parti diverse. Vista l'importanza di Python, in questa raccolta il secondo libro sarà completamente dedicato ad esso, in modo da poterlo studiare in maniera più approfondita in tutte le sue sfumature.

Infine, l'ultimo linguaggio su cui ci concentreremo è SQL. Questo è un po' diverso dagli altri perché si concentra principalmente sull'aiutarci a creare, riuscire ad analizzare e altro ancora, i nostri database. Il codice è tutto sommato abbastanza semplice da imparare e potremo farlo in poco tempo, quindi, in ogni caso, potrebbe essere una buona idea dargli un'occhiata.

Ci sono così tanti fantastici linguaggi di programmazione là fuori, ma i cinque in cui ci siamo concentrati saranno sicuramente la scelta migliore al fine di realizzare praticamente qualsiasi progetto che abbiamo pensato. Arrivato a questo punto sarai sicuramente pronto per saperne di più sulla programmazione per computer e su come partire in pochissimo tempo, prepariamoci per questo piccolo grande viaggio.

Esistono molti altri libri relativi a questo argomento sul mercato, grazie ancora per aver scelto questo!

È stato fatto ogni sforzo per garantire che sia pieno di quante più informazioni utili possibili, buon divertimento!

## **Capitolo 1: Le basi della programmazione**

Se stai cercando dei modi per rendere le tue lezioni di programmazione più facili, sicuramente fare il primo passo in questo nuovo mondo apparirà come un insormontabile montagna. E in molti casi, sembrerà un compito davvero gravoso. C'è un grande segreto che dovremo conoscere. Ci saranno un sacco di opzioni economiche e persino gratuite che sarai in grado di utilizzare per insegnare a te stesso le tecniche di codifica che vorresti utilizzare per rendere questo processo di apprendimento divertente ed eccitante.





Che tu sia un adulto affamato di conoscenza, in cerca di un grande cambiamento di carriera, o anche uno studente, che vuole davvero imparare un po' di più sull'ultimo linguaggio macchina uscito per spingersi verso il futuro, o un hobbista che vuole solo soddisfare la sua avidità di conoscenza su come funzionano questi servizi e il software che utilizziamo, allora la programmazione sarà uno dei modi migliori per aiutarti a portare avanti tutto questo. La buona notizia è che tutto ciò di cui abbiamo veramente bisogno per iniziare il nostro viaggio di programmazione è: innanzitutto un computer e poi l'accesso a Internet. In alcuni casi è anche possibile utilizzare solo uno smartphone. Prima di immergerci in parte del lavoro, probabilmente dovremmo iniziare con alcune nozioni sulle basi della programmazione, per aiutarci a iniziare.

## **Perché vuoi programmare?**

La prima cosa su cui dobbiamo dedicare un po' di tempo è esaminare i motivi, i veri motivi per cui sei interessato a imparare a programmare.

In questo caso è necessario essere onesti con se stessi. È questo il nostro tentativo di cambiare carriera? Abbiamo intenzione di acquisirne solo una parte per ottenere una promozione sul lavoro? Abbiamo intenzione di creare una nuova app o solo di far sfoggio delle nuove abilità acquisite?

La risposta ci servirà per capire quale, tra i cinque linguaggi macchina qui proposti, sia il migliore da padroneggiare per te, per impegno e in termini di denaro e tempo che questo obiettivo sta andando richiedere. Ad esempio, se hai il sogno di creare il tuo sistema operativo o un nuovo programma software in grado di competere con PhotoShop, ti consigliamo di utilizzare un linguaggio più potente come il C ++ per aiutarti ad affrontare alcuni degli argomenti più complessi che sono necessari.

D'altra parte, se hai una carriera professionale e vorresti essere in grado di passare ad un ambito più tecnico, allora dovresti conoscere meglio qualcosa di Raspberry, e qui Python andrebbe benissimo. Essenzialmente in base al nostro obiettivo o progetto, ci orienteremo sul linguaggio migliore per noi.

## **Scelta del linguaggio**

Una volta che hai avuto l'idea del motivo per cui vorresti programmare, l'obiettivo da perseguire, sarai più facilmente in grado di individuare il linguaggio che dovresti favorire rispetto agli altri. Non ne esiste uno migliore di altri, o che sia più user-friendly rispetto alla concorrenza. Ad esempio, opzioni come CSS e HTML di solito vengono considerate la base per entrare nella programmazione, ma in realtà ci aiutano solo a creare alcuni semplici siti, anzi, per la precisione, la struttura grafica di pagine internet per lo più visuali, niente a che vedere con l'e-commerce per capirci.

Se, al contrario, desideri lavorare con un linguaggio di codifica in grado di gestire database e fornirti siti Web interattivi che includeranno cose come sistemi di pagamento, allora lavorare con Python, SQL, PHP e JavaScript renderà il compito fattibile. Ma se desideri creare un'app per dispositivi mobili, lavorare con C o Swift sarà la scelta da seguire. Lo stesso vale per le app iOS, mentre le app Android funzioneranno meglio con Kotlin e Java. Gli sviluppatori di videogiochi apprezzeranno C# o Unity per dare vita ad alcune delle loro idee.

Se non sei ancora sicuro del tipo di linguaggio di codifica con cui vorresti lavorare, la maggior parte dei professionisti consiglierà di imparare a lavorare con JavaScript, C# o Python. Questo perché hanno la più ampia flessibilità e utilità in generale. Per aiutarti a prendere questa decisione, devi considerare ciò che speri di ottenere quando lavori ai tuoi progetti.

## **Cerca corsi online**

Per tutti quelli che hanno intenzione di cavarsela da soli, magari anche risparmiando, esistono diversi corsi online, completamente gratuiti. Ovviamente non avremo la possibilità di essere seguiti passo passo e di scambiarsi dubbi con gli insegnanti. La quantità dell'offerta su internet è tale che probabilmente tenderà a confonderci e impedirci di scegliere il miglior corso per noi.

Probabilmente passerai diversi giorni a cercare il linguaggio di programmazione perfetto. Se il fattore tempo non è un problema, allora sarai libero di trovare il codice perfetto per i tuoi scopi.

## **Concentrati maggiormente sul pensiero computazionale**

Un'altra cosa su cui dobbiamo considerare di lavorare è come pensare in modo più computazionale. Invece di concentrare la nostra attenzione sull'apprendimento di un solo linguaggio di programmazione specifico, puoi anche imparare un po' di più su come risolvere i problemi in maniera che il computer li interpreti al meglio. Inizialmente bisognerà soffermarsi su questo. Dovremo anche migliorare le nostre abilità sul riconoscimento di schemi, astrazioni e algoritmi.

Ci sarà anche un po' di gergo su cui devi dedicare il tuo tempo, come loop, dichiarazioni condizionali ed eccezioni, che troveremo molto spesso nel nostro viaggio, indipendentemente dal linguaggio di codifica con cui decidi di lavorare. Più sarai in grado di comprendere e passare *attraverso* queste ideologie, maggiormente facile sarà per te imparare la lingua che vorresti, così, ogni ulteriore codice con cui dovrai lavorare, diventerà sempre più d'immediata comprensione.

## **Prova alcuni giochi di codifica**

Ammettiamolo, non importa quanto sei motivato per il lavoro che dovrai fare con il linguaggio di programmazione scelto. Nessuno vuole passare un'eternità ad analizzare righe dopo righe di codice su un libro o su uno schermo. A volte sarà difficile e noioso, e se non hai avuto molto tempo per lavorare con la codifica, dopo un po' sembrerà incomprensibile.

Molti programmatori sanno che è molto più facile vedere gli esempi del codice con cui stanno lavorando attivamente, e questo li aiuta a scoprire davvero che questi concetti astratti hanno più senso in generale di quanto non facessero prima. Questa sarà parte della comodità che si trova con i tutorial di codifica interattivi che siamo in grado di trovare sul web e ci assicureranno che siamo in grado di fare davvero le cose.

Ad esempio, uno dei giochi o tutorial interattivi con cui siamo in grado di lavorare sarà conosciuto come FreeCodeCamp. Questo sarà in grado di scomporre alcuni concetti di codifica con cui siamo in grado di lavorare trasformandoli in blocchi più piccoli che verranno visualizzati nel browser Web che ci accompagnerà nel nostro percorso. Ci aiuterà anche ad abbinare ciascuno di questi concetti con un esercizio appropriato che dobbiamo decifrare prima di passare all'argomento seguente. Piuttosto che provare semplicemente a saltare ad alcune delle cose divertenti senza imparare questo metodo, sarà meglio conoscere il codice e come eseguirne il debug secondo necessità.



I giochi di programmazione saranno anche uno dei metodi divertenti che siamo in grado di utilizzare per bagnarci i piedi quando si tratta di programmazione. Se hai un po' di tempo da perdere e vuoi esercitarti nella programmazione, potresti lavorare con Hour of Code. Oppure ci sono molte scuole che lavoreranno con il programma Minecraft: Educational Edition, al fine di aiutare i bambini, apprendere alcune delle basi della programmazione, con blocchi di codifica e persino alcuni funzionano con JavaScript. Ci sono molte opzioni con cui siamo in grado di lavorare quando è il momento di saperne di più sulla codifica e trasformare il lavoro in qualcosa che sarà utile per le tue esigenze.

**Guarda alcuni video sulla codifica**

Un'altra opzione con cui siamo in grado di lavorare quando si tratta di imparare a programmare è guardare qualcun altro che ci mostra come portare a termine il tutto. Sarai in grado di trovare tutti i tipi di video educativi su praticamente qualsiasi argomento relativo alla programmazione che funzioni per le tue esigenze. Puoi trovarne molti su YouTube e altro, e dipenderà solo da ciò che vorresti imparare lungo la strada. Quando lavoriamo con i video giusti, sarai in grado di risolvere gli errori che devi correggere, imparare come eseguire diverse parti del processo di codifica e molto altro ancora.

**Realizza alcuni dei tuoi progetti**



Indipendentemente da quanti seminari di codifica svolgi o dal numero di certificati con cui lavorerai, la tua capacità di destreggiarti nel codice migliorerà in base ai tipi di programmi che vorrai creare. Sebbene non sia necessario realizzare un progetto personale super ambizioso, comunque dovrà essere qualcosa che ti appassiona e con cui vorresti lavorare a lungo termine per migliorarlo costantemente e, se necessario, aggiornarlo nei mesi a venire.

La cosa migliore qui è iniziare in piccolo con il lavoro che stai facendo, ma puoi pensare in grande. Il progetto che sceglierai coinvolgerà al meglio alcune abilità che già fanno parte della tua conoscenza di base. Devi anche avere una sorta di piano in atto per le abilità e le caratteristiche future che potresti voler apprendere per trasformare quella visione in una realtà che puoi utilizzare in questo progetto e altro ancora.

Se non sarai in grado di applicare le tue capacità di programmazione a più di un progetto, probabilmente non verrai notato. Ciò significa che dobbiamo andare oltre, e fare un po' di pianificazione e pratica. Quando avrai alcuni progetti alle spalle, non solo con la pratica migliorerai, ma sarai anche visto come un professionista nel lavoro che stai facendo.

**Celebrate le vostre piccole vittorie**

Il nostro “lavoro” di programmatore sarà costellato di dettagli a cui, ogni volta, dovremo prestare grande attenzione. Per questo dovremo celebrare i nostri piccoli obbiettivi superati, sarà molto importante. Ci vorrà molta pratica per far funzionare ogni elemento insieme con gli altri e, come se non bastasse, senza alcun tipo di errore. Se, per qualche motivo non ti prenderai cura dei dettagli, a prima vista trascurabili, come ad esempio chiudere il tag HTML, passerai molto tempo a eseguire il debug di un semplice errore nella sintassi, piuttosto che dedicare tempo a scrivere codice in modo più intelligente e complesso.

Dovrai anche assicurarti di essere estremamente paziente. Non ti aspettare di codificare il prossimo Fortnite o un altro progetto grande nonché di successo, con poche lezioni sul tema o solo poche ore di studio. Creare un gioco di previsione dei numeri che sia semplice ma accurato, o una codifica priva di errori di un modulo RSVP per un sito Web sarà effettivamente un risultato davvero significativo per qualcuno che sia un principiante.

Anche se stai lavorando con alcune di queste opzioni meno importanti, all'inizio, non dovresti sentirti male se rimani bloccato. Non usarlo come una scusa per arrendersi. E dovresti sapere quando è il momento di prendertela comoda per superare la fase di esaurimento che sicuramente ci sarà, così come cercare sempre assistenza ovunque sia possibile, e riprovare finché il programma non funzionerà bene.

## **Googlare i messaggi di errore**

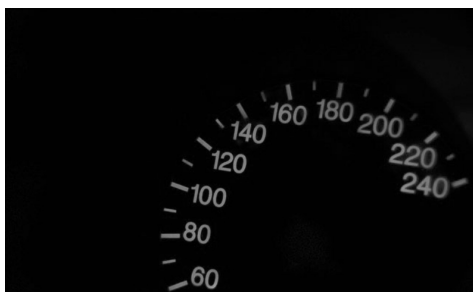
Questo è in realtà qualcosa che può aiutarti indipendentemente dal tipo di linguaggio di programmazione con cui lavorerai. Se stai eseguendo la codifica e non sei sicuro del motivo per cui il codice è buggato e perché non funziona nel modo desiderato, puoi sempre eseguire e trovare alcune delle soluzioni che renderanno tutto più semplice online. È probabile che altri principianti avranno lo stesso tipo di errore lungo la strada e se copi e incolli semplicemente il messaggio di errore in un motore di ricerca qui, dovresti essere in grado di trovare alcune informazioni su quell'errore.

Questo renderà molto più facile per te capire cosa sta succedendo con il tuo codice e ti assicurerà di essere in grado di apprendere e risolvere il problema allo stesso tempo. Se stai ancora riscontrando problemi quando si tratta di questo, sarai quindi in grado di pubblicare la tua domanda su alcuni dei forum con cui vuoi lavorare, inclusi GitHub, Stack Overflow e altro.

**Conoscere un modo più intelligente per imparare a  
programmare in un giorno**

Mentre tutti i suggerimenti di cui abbiamo parlato sopra saranno ottimi per aiutarti a saperne di più sulla codifica, è importante imparare il trucco numero uno per ridurre questa codifica in pochissimo tempo. Se sei davvero impegnato e utilizzi alcuni dei codici e delle informazioni sui codici in questa guida, puoi persino impararli in un giorno!

Il trucco qui è sporcarsi le mani e iniziare effettivamente a scrivere codice. Puoi passare tutto il tempo che desideri a leggere libri, testi e siti Web su questi codici, ma finché non apri effettivamente il compilatore e inizi a digitare tutto, rimarrai indietro. Più codici puoi digitare e fare pratica, più velocemente imparerai.



Può sembrare che questo trucco sia troppo semplice, ma è davvero tutto ciò di cui hai bisogno per assicurarti che la codifica ti venga facilmente e che vedrai i migliori risultati in pochissimo tempo. Scegli semplicemente quale lingua vorresti imparare da questa guida, prenditi il tempo di leggere quella sezione, quindi esci davvero e inizia a digitare e praticare alcuni di quei codici, e sarai pronto per partire.

## **Buone combinazioni linguistiche**

Quando stai imparando un nuovo codice, è una buona idea rapportarlo e confrontarlo con un altro e capirne le differenze e le possibili integrazioni. Essere in grado di combinare due o più lingue e utilizzarle contemporaneamente può semplificare notevolmente il lavoro di codifica. Daremo uno sguardo ad alcune delle migliori combinazioni che troverai quando sarà il momento di riunire due codici contemporaneamente.

Per cominciare, possiamo guardare SQL e Python. SQL è un linguaggio progettato per interagire con i database ed è standardizzato per quelli basati sul modello relazionale. Può aiutare a creare, modificare, regolare, eliminare e altro ancora i numerosi database su cui facciamo affidamento per la nostra organizzazione. Consideriamo che combinarlo insieme a Python apporta al codice un livello ancora superiore.

Python viene fornito con uno standard utilizzato con le interfacce di database. Questo standard è l'API DB di Python. E' utilizzato per aiutare a definire un'interfaccia comune, inoltre servirà a accedere ad alcuni dei database relazionali che SQL aiuta a eseguire. In altre parole, il codice che usiamo in Python per la comunicazione al database finirà allo stesso modo, indipendentemente dal database o modulo del database che decidiamo di utilizzare. Quindi, anche se useremo molto SQL sui database, il linguaggio Python può intervenire e fare la differenza.

Anche il linguaggio C # può essere utilizzato bene con SQL. Questo perché è progettato per aggiungere un po' di potenza quando è il momento di completare le ricerche su qualsiasi database che cerchiamo di creare con SQL nel nostro lavoro. Questo renderà semplicemente le cose un po' più facili e aggiungerà ad alcune delle funzionalità che siamo in grado di gestire questo tipo di linguaggio.

## **Perché questi linguaggi di programmazione sono importanti da imparare?**

Ci sono tanti motivi per cui dovremo dedicare del tempo all'apprendimento di un linguaggio di programmazione. E possiamo passare tutto il giorno a parlare dei vantaggi di imparare a programmare. Questo può aiutarci a saperne di più sui nostri computer, a trovare un nuovo percorso professionale e altro ancora. Ma la domanda principale che vogliamo esaminare è perché ognuna di queste lingue è così importante, da sola, da imparare.

I primi sono i linguaggi C, C# e C++. Sono considerati alcuni dei linguaggi più potenti, quelli che saranno perfetti per i grandi progetti o quelli che avranno molte cose che colpiscono contemporaneamente, come grandi database, siti Web, giochi e altro ancora. Questi a volte sono visti come più difficili da imparare e come se avessero più parti su cui lavorare, ma spesso, per i tuoi codici di livello superiore, ne varrà la pena.

Poi c'è il linguaggio Python. Questo è considerato un linguaggio generico che può essere utilizzato per quasi tutti i tipi di codifica o programmi che desideri. È stato progettato pensando ai principianti, quindi sebbene abbia un sacco di potenza e funzionalità dietro di esso, è anche abbastanza facile da imparare per chiunque. Questa è spesso la lingua consigliata ai principianti a causa di queste caratteristiche.

Il prossimo nell'elenco è il linguaggio Java. Questo è in grado di aiutarti con una serie di programmi che vorresti conoscere. Ma è particolarmente noto per aiutare nella creazione di siti Web. Sebbene uno qualsiasi di questi linguaggi, ad eccezione di SQL, possa essere utilizzato per creare siti Web e gestirli, nessuno è in grado di gestirlo come si vede con Java. Questa lingua è impostata per rendere questi siti Web di alta classe e di grande effetto e aggiunge anche un elemento di interazione a questo.

E infine, raggiungiamo il linguaggio SQL. Se sei un'azienda che si occupa di database, il linguaggio SQL è quello con cui devi passare un po' di tempo. Questo linguaggio è stato progettato per soddisfare tutte le esigenze del database, dalla creazione alla modifica, eliminazione, aggiunta e molto altro ancora. Le altre lingue a volte possono aiutare con questo, ma sarà davvero l'opzione SQL a farlo funzionare bene.

Ci sono molte cose interessanti su cui possiamo dedicare del tempo quando si tratta di imparare di più sulla programmazione. Dedicheremo un po' di tempo alla ricerca di alcune delle migliori opzioni che ci aiuteranno a migliorare con parte del lavoro che stiamo cercando di fare nella codifica. Immergiamoci subito per iniziare!

## **Capitolo 2: I diversi sistemi operativi**

La prossima cosa su cui dobbiamo dedicare un po 'di tempo sono i sistemi operativi con cui potremo lavorare per la nostra programmazione. Ci sono tre principali sistemi operativi su cui possiamo dedicare il nostro tempo per aiutare con la programmazione, inclusi i sistemi operativi Windows, Linux e Mac. Diamo un'occhiata ad alcuni dei diversi vantaggi e funzionalità che saremo in grado di vedere con ciascuno di questi sistemi operativi.

## **Sistema operativo Windows**

Il primo sistema operativo con cui lavoreremo è il sistema operativo Windows, progettato dalla società Microsoft. Questo è un sistema che ti permetterà di accedere a molti dei computer progettati e si troverà precaricato su molti dei nuovi personal computer con cui vorrai lavorare, il che fa parte del motivo per cui questo è il più popolare sistema operativo in tutto il mondo.

Questo sistema operativo ci consentirà di completare un'ampia varietà di attività che dovrai svolgere quotidianamente sul tuo computer. Ad esempio, è possibile utilizzare Windows per connettersi online, controllare la posta elettronica, modificare alcune immagini digitali, ascoltare musica, giocare e molto altro ancora lungo il percorso. Windows verrà utilizzato anche in molti uffici perché fornirà gli strumenti di produttività di cui abbiamo bisogno, inclusi fogli di calcolo, elaboratori di testi e calendari.



Microsoft ha rilasciato la prima versione di Windows a metà degli anni '80. Ci saranno molte versioni più recenti di Windows da quel momento, ma la più recente includerà Windows 10, che è stato rilasciato nel 2015. Altre versioni popolari includeranno Windows Vista, Windows XP, Windows 8, e Windows 7.

## **Le applicazioni**

Un'applicazione, che sarà spesso conosciuta come un programma in questo sistema operativo, sarà un tipo di software che ti permetterà di completare alcune attività sul tuo computer. Windows verrà fornito con molte applicazioni integrate che siamo in grado di utilizzare. Quindi, se desideri lavorare con Microsoft Edge, puoi aprirlo e iniziare a navigare in Internet.

L'apertura di queste applicazioni sarà in grado di aiutarci a realizzare quella particolare azione. È possibile utilizzare il mouse e quindi fare clic sul pulsante Start. Quindi sei in grado di selezionare l'applicazione desiderata. Se non vedi quella che desideri, puoi fare clic su Tutti i programmi per visualizzare un elenco completo delle applicazioni disponibili sul tuo sistema.

L'applicazione apparirà in una nuova finestra. Tieni presente che non ci sarà un menu Start in Windows 8, quindi di solito dovrai aprire queste applicazioni nella schermata Start. Fare clic su questo pulsante per andare alla schermata Start, quindi fare clic sull'app per aprirla.

## **Lavorare con Windows**

Ogni volta che sei pronto per aprire un'applicazione, una cartella o un file, apparirà in una nuova finestra. È quindi possibile avere più di un elemento aperto contemporaneamente in finestre diverse. Lo userai per la maggior parte del tempo, quindi è importante sapere davvero come passare da una finestra aperta all'altra, come spostare e ridimensionare le finestre e come chiuderle quando hai finito di lavorarci.

Per aiutarci, dobbiamo lavorare con le parti di una finestra. Vedrai che ci saranno i tre tipi di pulsanti che si trovano nell'angolo in alto a destra di tutte le finestre con cui lavori. Questi pulsanti saranno lì per rendere più facile eseguire alcune funzioni diverse. Questi includeranno:

1. È possibile fare clic sul pulsante Riduci a icona per nascondere la finestra. Questa finestra verrà ridotta a icona sulla barra delle applicazioni. È quindi possibile fare clic sull'icona di quella finestra sulla barra delle applicazioni e verrà visualizzata di nuovo la finestra.
2. Fare clic sul pulsante Ingrandisci in modo che la finestra occupi l'intero schermo per semplificare il lavoro.
3. Se hai lo schermo ingrandito, scoprirai che il pulsante Ingrandisci verrà temporaneamente modificato nel pulsante Ripristina. È possibile fare clic su questo pulsante per riportare la finestra alla dimensione originale che dovrebbe essere.
4. Quindi il pulsante finale sarà il pulsante Chiudi. Questo ti aiuta a chiudere la finestra e andare avanti.

È anche possibile spostare una finestra con cui stiamo lavorando. Per aiutarci a spostare una finestra. Per fare ciò, faremo clic e quindi trascineremo la parte superiore della finestra in modo che possa essere spostata nella posizione corretta sullo schermo. Quando hai finito con questo, puoi rilasciare il mouse. Quindi la finestra si troverà nella nuova posizione in cui vorresti che fosse.

Un'altra cosa su cui lavorare è ridimensionare la finestra. Puoi passare il mouse in modo che si trovi su qualsiasi angolo della finestra, quindi fare clic e trascinare il mouse per ingrandire o rimpicciolire la finestra. Quando hai finito, puoi rilasciare il mouse. La finestra può quindi essere ridimensionata alla dimensione che vorremmo che fosse.

## **Spegnimento del computer**

Quando hai finito di lavorare con il tuo computer su cui è installato il sistema Windows, è importante eseguire il processo di spegnimento nel modo corretto. La procedura con cui puoi lavorare varia in base alla versione di Windows con cui stai lavorando. Se stai lavorando con Windows 10, fai clic sul pulsante Start, quindi sull'icona di alimentazione, quindi fai clic su in modo da poter spegnere.

Se stai utilizzando una versione precedente di questo, come Windows 8, dovrai fare clic sul pulsante Start per accedere alla schermata Start, fare clic sull'icona di alimentazione vicino all'angolo in alto a destra e selezionare Spegni. Se non vedi questo tipo di opzione, significa che dovrai eseguire l'aggiornamento a Windows 8.1 per farlo funzionare. Ma se stai lavorando con Windows 7 o una versione precedente, puoi fare clic sul pulsante Start e quindi selezionare l'opzione per Arresta il sistema e il computer si spegnerà.

## **I vantaggi dell'utilizzo di Windows**

Ci sono alcuni vantaggi che saremo in grado di vedere quando si tratta di lavorare con il sistema Windows. Il primo vantaggio è che Windows è facile da usare. Scoprirai che puoi imparare a usarlo quasi istantaneamente e non ci sarà affatto una grande curva di apprendimento che ne consegue. Anche il software sarà facile da ottenere. La popolarità di questo sistema sarà dovuta alla disponibilità che ha sul mercato del software. Il software sarà ovunque e saremo in grado di usarlo per usi personali e aziendali.

Windows sarà anche l'hardware che i produttori sceglieranno. Questo si troverà sulla maggior parte dei nuovi computer con cui scegli di lavorare, il che ci permette di sapere che è forte e facile da lavorare. E se ti piacciono i giochi e hai molta varietà quando si tratta del tuo sistema operativo, allora il sistema Windows sarà il migliore con cui lavorare. Poiché questo sistema operativo è robusto e in grado di supportare qualsiasi tipo di gioco che un giocatore vorrebbe utilizzare, questa è una scelta popolare da seguire se questo è il tuo obiettivo principale.

E il vantaggio finale che esamineremo è che Windows sarà in grado di adattarsi anche agli altri prodotti disponibili da Microsoft. questo garantirà che saremo in grado di vedere alcuni dei vantaggi che desideriamo in pochissimo tempo. Sarai in grado di lavorare con l'opzione Internet di Microsoft, Word, Excel e altri prodotti simili, che possono davvero semplificarti la vita.

## **Sistema operativo Linux**

Il prossimo sistema operativo in cui ci immergeremo è il sistema operativo Linux. Questo potrebbe non essere famoso come il Mac e i sistemi operativi Windows, ma ha molte caratteristiche che lo rendono particolarmente adatto per aiutare con la codifica e la programmazione, quindi vale la pena dedicare del tempo per saperne di più.

Per cominciare, il sistema operativo Linux sarà una delle versioni popolari che siamo in grado di trovare con Unix. Sarà open source poiché il codice che usiamo per esso è disponibile gratuitamente, rendendolo libero da usare. È progettato con la compatibilità che abbiamo imparato a conoscere e ad amare, e persino a aspettarci, da UNIX, e anche le funzionalità che vedremo con questo saranno simili a quelle di UNIX.

La prima cosa che dobbiamo esplorare quando si tratta del sistema Linux sono i tre componenti principali che lo compongono. Questi includeranno:

1. Il kernel: questo sarà il cuore di Linux. Fondamentalmente sarà la parte responsabile delle attività principali di questo sistema operativo. Consisterà in una varietà di moduli e interagirà direttamente con qualsiasi hardware sottostante. Il kernel ci fornirà l'astrazione necessaria per nascondere i dettagli hardware di basso livello per i programmi applicativi o il sistema.
2. Libreria di sistema: questa sarà una funzione o un programma speciale che sarà in grado di utilizzare i programmi applicativi o le utilità di sistema che possono accedere alle funzionalità del kernel. Questi aiuteranno a svolgere la maggior parte delle funzionalità del sistema operativo e non avranno bisogno dei diritti di accesso del codice del modello del kernel.
3. Utilità di sistema: questa parte finale riguarderà i programmi che possono essere responsabili di svolgere attività specializzate e anche individuali lungo il percorso.

## **Livello di esecuzione**

La seconda cosa che dobbiamo esplorare è il livello di esecuzione in questo sistema. Il livello di esecuzione sarà una modalità o uno stato che si trova in questo sistema operativo. Ce ne saranno sette e verranno rappresentati con un numero intero a una cifra. Sarà una delle cose più importanti che dobbiamo imparare in Linux. E il kernel di cui abbiamo parlato prima è in grado di differenziare e supportare tutti i livelli. I secondi livelli di esecuzione che dobbiamo esplorare quando siamo in Linux includeranno:

1. Il livello di esecuzione 0: questo significa che il sistema sarà nel suo stato di trasporto, senza attività in corso.
2. Livello di esecuzione 1: significa che il sistema è in modalità per un singolo utente. Questo verrà usato raramente.
3. Livello di esecuzione 2: questo significa che il sistema è in modalità per più di un utente, ma non sarà presente alcun file system di rete. Anche questo spesso non viene utilizzato.
4. Livello di esecuzione 3: significa che il sistema viene utilizzato in modalità riga di comando con più utenti. Questo sarà il livello di esecuzione predefinito che troveremo sulla maggior parte dei sistemi che utilizzano Linux.
5. Livello di esecuzione 4: questo significa che siamo in una modalità che verrà definita dall'utente. Questo è un altro livello che non viene utilizzato più di tanto.
6. Livello di esecuzione 5: significa che il sistema sarà in modalità per più utenti, ma la GUI è a posto o l'interfaccia utente grafica. Questo sarà anche un altro livello di esecuzione standard per molti di questi sistemi con cui lavoriamo.
7. Livello di esecuzione 6: questo verrà utilizzato ogni volta che il nostro utente desidera poter riavviare il sistema per aiutare con un riavvio quando le cose non funzionano come vorremmo.

## **I vantaggi di Linux**

Dobbiamo anche dedicare un po' di tempo a dare un'occhiata ad alcuni dei vantaggi che arriveranno con il sistema operativo Linux. Questo ci aiuterà a garantire che stiamo utilizzando il sistema operativo nel modo giusto e che siamo in grado di vedere alcuni dei migliori risultati lungo il percorso. Ci sono molti vantaggi che vedremo quando si tratta di questo sistema operativo, tra cui;

1. Il modello operativo è coerente. Indipendentemente dalla versione o distribuzione che stai utilizzando con Linux, che tu sia un piccolo dispositivo incorporato o un supercomputer, scoprirai che l'operazione con questo sistema sarà la stessa, qualunque cosa accada. Ciò significa che lavorerai con la stessa gestione dei processi, la sintassi della riga di comando, l'amministrazione di rete e le applicazioni in generale.
2. Scalabilità: sei in grado di prendere il sistema operativo Linux e ridimensionarlo quanto vuoi. Ciò significa che è in grado di funzionare su qualsiasi cosa, dall'orologio sul tuo braccio a un supercomputer e fino ad alcuni dei cluster di computer distribuiti in tutto il mondo. Questo ci aiuterà a vedere davvero alcuni dei diversi modi in cui siamo in grado di lavorare con questo programma.
3. Open source e ottimizzato dalla community. Poiché la natura di Linux è disponibile gratuitamente e open source, potresti essere preoccupato per il supporto, la correzione di bug e miglioramenti futuri. La buona notizia è che la comunità attorno a Linux sarà in grado di mantenerlo attivo e con più di 22 milioni di righe di codice nel solo kernel, scoprirai che esiste già una grande comunità, con circa 5000 singoli sviluppatori dietro le quinte a partire dal 2016, aiutando a sviluppare il sistema Linux. E tutto questo sviluppo ti aiuterà a trarre vantaggio quando lavori sul sistema Linux.



4. Rete di piena unzione. Nel corso degli anni, Linux ha sviluppato un solido set di funzionalità di rete, inclusi molti strumenti per il networking per fornire e gestire bridging, routing, DNS, DHCP, risoluzione dei problemi di rete e monitoraggio della rete e reti virtuali.
5. Gestione dei pacchetti: il vantaggio finale che vedremo è come Linux è in grado di gestire i pacchetti che passano attraverso. Il sistema di gestione dei pacchetti Linux ci consentirà di installare qualsiasi nuova applicazione e servizio che desideriamo con pochi comandi di base per aiutarci a fare tutto.

## **Sistema operativo Mac**

Il terzo tipo di sistema operativo su cui passeremo un po 'di tempo è il sistema operativo Mac. Questo sarà il sistema operativo utilizzato per aiutare a eseguire qualsiasi computer Mac e Apple. Sarà il sistema più semplice disponibile e consentirà al computer Mac di funzionare. Tutti i programmi, cartelle e file presenti sul tuo computer saranno gestiti da questo sistema operativo, così come la connettività Internet, il consumo della batteria e molto altro ancora nel processo.

Quando lavori con il tuo computer Mac, scoprirai che ci sono molti modi diversi in cui puoi interagire con questo sistema operativo. Alcuni dei formati che incontrerai includeranno:

1. Quando stai visualizzando il desktop del computer.
2. Quando impieghi del tempo a sfogliare i file sul tuo computer.

3. Quando si utilizza la funzione Spotlight, o ricerca, sul computer.
4. Quando si utilizza il Dock o la barra delle applicazioni sul computer.
5. Poiché il sistema sta lavorando per eseguire le tue applicazioni.

E questi sono solo alcuni dei punti di partenza di quando lavorerai con il sistema operativo Mac. In realtà è sempre in esecuzione in background, assicurandoti che il computer funzioni nel modo che desideri e che cosa puoi fare per far funzionare le cose nel modo che desideri.

Per iniziare con alcuni più dettagli del sistema Mac è la casa di questo sistema o il desktop. Questo può essere pensato come il desktop fisico o un luogo in cui è possibile distribuire il lavoro che si sta tentando di svolgere durante l'esecuzione di un'attività. Ad esempio, è possibile inserire un'immagine da Internet su questo desktop, apportare alcune modifiche e quindi trascinarla in un'e-mail per inviarla a qualcun altro. Ricorda, anche se questo desktop sarà un ottimo posto per fare il tuo lavoro, non è un posto a lungo termine per archiviare i tuoi file perché diventerà disordinato, quindi devi avere i file archiviati in cartelle e altro per mantenerlo organizzato e facile da lavorare.

## **La barra Dock**

Il prossimo sulla lista sarà il Dock. Quando guardi lo schermo del tuo computer Mac, vedrai ciò che è noto come Dock. Questa è fondamentalmente una barra che avrà tutte le scorciatoie di cui hai bisogno per alcune delle applicazioni che usi di più. Se non lo vedi già lì, puoi spostare il cursore in modo che vada sulla parte inferiore dello schermo, e poi questi si apriranno da soli. Le icone che si trovano su questo Dock saranno scorciatoie e potrai rimuovere o aggiungere le icone senza avere alcun effetto sulle applicazioni stesse.

Per aggiungere un nuovo collegamento dell'applicazione a questo dock, devi solo lavorare con il Launchpad, che sarà una panoramica completa delle applicazioni disponibili sul tuo computer. Nella maggior parte dei casi, sarai in grado di trovare questo Launchpad nella parte sinistra del Dock. Devi solo fare clic su questo e quindi visualizzare le applicazioni disponibili, prima di trascinare un'icona nel dock per creare un nuovo collegamento, se lo desideri.

Un altro modo in cui possiamo eseguire il processo di aggiunta di alcune scorciatoie che vorremmo al Dock è trovare il file dell'applicazione (che si troverà nella cartella Applicazioni), quindi trascinarlo nella Dock. Per far sì che ciò accada, tuttavia, dobbiamo assicurarci di avere una certa familiarità con la navigazione tra i file quando esaminiamo tutto questo, quindi se non hai questa familiarità, allora devi lavorare con il Launchpad di cui abbiamo parlato circa prima.

È anche possibile esaminare e provare a rimuovere alcuni degli elementi che si trovano automaticamente sul nostro dock. Per fare ciò, dobbiamo solo trascinarlo fuori e tenerlo sul desktop invece che sul dock. Dopo circa un secondo, vedrai apparire un'etichetta che dice "rimuovi". Puoi rilasciare l'icona, quindi il collegamento dell'applicazione di cui stiamo parlando verrà rimosso dal Dock.

## **I menu**

Ora passeremo alla parte superiore del nostro schermo, dove potremo vedere alcuni menu. Questi menu cambieranno a seconda del tipo di applicazione che intendi utilizzare in quel momento. Se c'è qualcosa che vuoi fare con l'applicazione in cui ti trovi in quel momento, è meglio dare un'occhiata a quelle azioni nei menu.

In tutto questo possiamo anche trovare alcuni extra di menu. Siamo in grado di trovarli nella parte superiore dello schermo in cui ci troviamo. A destra di dove siamo stati in grado di trovare i menu, vedremo anche alcuni altri simboli. queste piccole icone verranno utilizzate per rendere più semplice la modifica delle nostre impostazioni su Mac. Invece di dover seguire e seguire il processo per aprire le nostre Preferenze di Sistema, siamo in grado di modificare le impostazioni con gli extra del Menu.

Per cambiare quale di questi extra sarà visibile, è sufficiente fare clic sulle preferenze di sistema. Quindi fai clic sul pannello per le preferenze e cerca la casella di controllo che ti consentirà di scegliere se desideri che quel particolare extra per il menu sia visibile o meno in quel momento.

## **Uno sguardo agli utenti**

La prossima cosa che possiamo dare un'occhiata a questo sistema operativo sono gli utenti. Questo sistema operativo ci offrirà un modo davvero carino per passare da un account all'altro per gli utenti sullo stesso computer. Sarai in grado di guardare vicino all'angolo in alto a destra dello schermo e vedere il tuo nome lì. Fare clic su di esso e quindi visualizzare un elenco degli utenti che si trovano su quel computer. Se sei l'unico che utilizza quel computer, sarai l'unico utente elencato lì.

Se condividi il computer con altri, noterai che ci saranno molti nomi diversi che verranno visualizzati anche quando fai clic su questo. Quando vuoi passare a un nuovo utente, devi solo fare clic sul suo nome e quindi accedere al suo account. Se è richiesta una password, sarà necessario concentrarsi anche su questa e aggiungerla.

## **Spotlight**

Un'altra cosa su cui possiamo dedicare un po' del nostro tempo è Spotlight. Se guardi nell'angolo in alto a destra dello schermo su cui stai lavorando, vedrai che c'è una lente d'ingrandimento più piccola. Questa sarà la funzione di ricerca per questo sistema operativo: Spotlight.

## **La dashboard**

Questo sistema operativo avrà anche una grande cosa con cui siamo in grado di lavorare, nota come Dashboard. È qui che siamo in grado di eseguire alcune mini-applicazioni note come widget. Questo può permetterci di avere molta libertà potenziale con il lavoro che vogliamo fare in questo sistema operativo, e se prevedi di avere del lavoro che richiede widget, allora questa è la posizione in cui sarai in grado di trovare loro.

## **I vantaggi del sistema operativo Mac**

La prossima cosa che dobbiamo esaminare sono alcuni dei vantaggi che derivano da questo tipo di sistema operativo. Ci sono molte opzioni che possiamo esaminare quando è il momento di concentrarci su questi sistemi operativi e abbiamo già esaminato alcune di esse. Allora perché dovremmo dedicare il nostro tempo a lavorare con il sistema operativo Mac rispetto ad alcuni degli altri? Alcuni dei vantaggi che vedremo quando si tratta di lavorare con questo tipo di sistema operativo includono:

1. L'interfaccia è semplice ma ha molta potenza. Questo ci offrirà un approccio molto semplice quando si tratta di informatica e, se sei abituato a lavorare con un iPad o un iPhone, scoprirai che l'interfaccia ti sarà familiare. L'interfaccia è stata progettata per essere facile da usare per chiunque, rendendola più attraente per coloro che desiderano la semplicità per portare a termine il lavoro. Ricorda, tuttavia, che quasi tutti questi sistemi operativi non saranno intuitivi per impostazione predefinita. Ciò che lo distinguerà, tuttavia, è che sarà in anticipo e pulito nel complesso?
2. Viene fornito con meno problemi di sicurezza e virus in generale: uno dei maggiori vantaggi che vedremo con questa opzione è che viene fornito con meno problemi di sicurezza e virus rispetto ad altri sistemi operativi. In passato, si credeva che i computer Apple non potessero contrarre virus, ma da allora è stato considerato falso. Anche se questo sistema operativo può gestire occasionalmente bug e virus, il numero di questi incidenti è diminuito e non è così diffuso come lo sono con altre opzioni. E richiede molta meno manutenzione per funzionare rispetto a Windows, quindi questo è un altro vantaggio di cui godere.

3. L'hardware e il sistema operativo si integreranno senza problemi. Poiché Apple è un produttore di dispositivi e uno sviluppatore di software, sono in grado di far funzionare bene l'hardware e il sistema operativo. Ciò garantirà che il computer funzioni nel modo in cui desideri, senza molti problemi lungo il percorso.
4. Viene fornito con alcune app di produttività precaricate. Questo sistema operativo verrà fornito con alcune delle app di cui hai bisogno per iniziare subito a lavorare. Ad esempio, molti dei computer Mac con cui decidi di lavorare arriveranno con Keynote, Numbers e Pages già presenti e pronti per l'uso.
5. Viene fornito con una funzionalità multitasking efficace e senza precedenti: il multitasking sarà un grande vantaggio di questo sistema. Questi computer avranno le specifiche hardware progettate per gestire il multitasking di cui abbiamo bisogno per portare a termine. Si assicurerà inoltre che siamo in grado di gestire tutte le operazioni che vogliamo con un flusso fluido e reattivo e il sistema sarà in grado di rispondere ad alcuni dei tasti di scelta rapida incorporati, i gesti del mouse e track-pad per passare da app o finestre in esecuzione in background.

Sebbene ci saranno molti vantaggi che arriveranno lavorando su questo sistema operativo, ci sono alcuni motivi per cui le persone sceglieranno di non utilizzare affatto questa opzione. Ad esempio, questi computer Mac potrebbero funzionare bene, ma saranno molto costosi. Se hai un budget limitato, questa non è l'opzione migliore tra cui scegliere. Quando si tratta di alcuni degli aggiornamenti dell'hardware che si desidera utilizzare, questi sistemi saranno poco flessibili e spesso sarà necessario acquistare un altro computer costoso per farlo accadere.

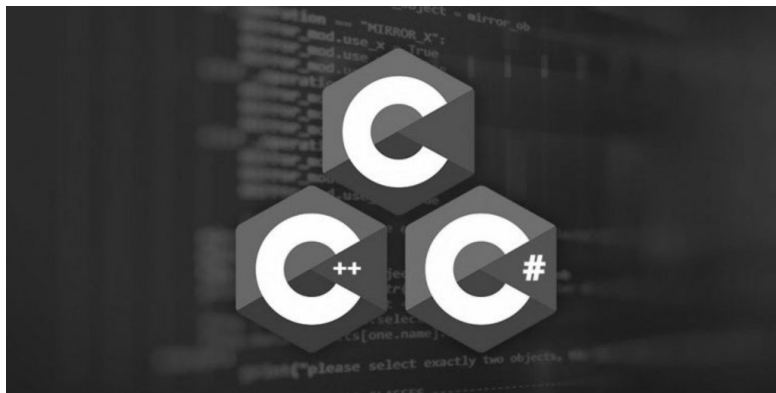
Mancano anche titoli di gioco e un'esperienza di gioco avanzata quando si lavora con uno di questi computer. Per i giocatori accaniti, lavorare con Windows fornirà più opzioni di quelle che vedremo concentrandoci sul sistema operativo Mac. Ci sono una serie di ragioni per questo, ma se desideri passare un po' di tempo a giocare, allora devi essere consapevole di questo. E l'ecosistema di app che vedremo con questo è ancora un po' poco brillante e può allontanare alcune persone da esso nel complesso.

Come possiamo vedere, ci sono molte ottime opzioni che possiamo utilizzare quando è il momento di gestire alcune delle scelte nei sistemi operativi. La scelta del sistema operativo giusto a volte può essere una questione di preferenza, ea volte varia maggiormente in base al tipo di codici che speri di scrivere, al linguaggio di codifica che scegli di utilizzare e al risultato finale che desideri ottenere . Scegliere il sistema operativo giusto dall'inizio sarà così importante per assicurarti di essere davvero in grado di eseguire la codifica che desideri.

### **Capitolo 3: I linguaggi C.**



Ora è il momento per noi di dare un'occhiata ad alcune delle programmazioni che siamo in grado di fare con le competenze che abbiamo. Inizieremo con alcuni dei linguaggi C disponibili, inclusi C, C ++ e C #. Diamo un'occhiata a come siamo in grado di lavorare, come sono simili e come sono anche un po 'diverse.



## **Uno sguardo a C**

L'obiettivo principale che vedremo con il linguaggio C è che avrebbe dovuto essere un linguaggio efficiente ma che sarebbe anche in grado di formarsi attorno a una comunità per avere una fratellanza. La comunità intorno a questo sapeva che avevano bisogno di mantenere le cose senza intoppi e, per farlo, non potevano semplicemente inserire le informazioni in un terminale da sole. Ma avevano bisogno di comunicare anche dopo la codifica. Il risultato di questa idea è stato il linguaggio C. Questo è un linguaggio strutturato e di uso generale, noto come linguaggio di programmazione orientato alla procedura. Non è stato progettato per funzionare con un tipo specifico di applicazione per l'avvio, ma funziona anche per applicazioni aziendali e usi scientifici.

## **Ottenere l'IDE**

La prima cosa che dobbiamo fare prima di usare C è configurare l'IDE. Questo è noto come ambiente di sviluppo integrato ed è così importante perché ci aiuterà a scrivere, collegare, modificare, compilare e persino eseguire i codici che vorremmo. L'IDE ci aiuterà ad assicurarci che il linguaggio C funzioni.

Il miglior IDE da utilizzare è l'IDE Code Blocks. Questo verrà fornito con le funzionalità necessarie lungo il percorso. Non dovrai passare del tempo online o trovare un altro compilatore con le parti giuste perché questo IDE lo avrà già dall'inizio. Puoi trovare questo particolare IDE visitando il sito web <http://www.codeblocks.org>.

## **Scrivere un codice fittizio**

Dobbiamo anche dedicare del tempo ad esaminare alcune delle basi per scrivere codici in questo linguaggio. Ciò renderà le cose un po 'più semplici e ci assicurerà di essere in grado di capire davvero come funziona questo processo e come possiamo lavorare su alcuni dei nostri codici in seguito. Questo è un ottimo modo per assicurarci di poter scrivere e salvare codici nel linguaggio C per dopo.

Per iniziare, dobbiamo entrare nell'IDE Code Bocks che abbiamo appena aggiunto al nostro computer, oppure puoi usare qualsiasi altro IDE che desideri. Da lì, fai clic sul pulsante Nuovo in modo da poter aprire un file vuoto. Puoi quindi digitare un codice a tua scelta nell'editor in modo da poter digitare il codice e salvarlo. Un semplice codice con cui vogliamo lavorare, tra cui:

```
main() {}
```

Ora, quando arriviamo a questo punto, dobbiamo salvare il nostro file del sorgente. Questo può essere fatto facilmente quando facciamo clic sul pulsante Salva. Puoi scegliere di salvare questo file predefinito per il tuo computer o dedicare un momento a scegliere la cartella in cui devono finire i file di codice u. In ogni caso, ricorda la posizione in modo da poterla trovare in seguito.

Un'altra cosa da considerare è il nome del file che vuoi salvare. Deve essere qualcosa di facile in modo che tu possa ricordarlo e trovarlo effettivamente in un secondo momento, se necessario. Lo chiameremo `u201cdummy.u201d` in modo che possiamo essere tutti sulla stessa pagina, ma puoi sicuramente scegliere qualcosa che è un po 'più facile in seguito.

Mentre siamo in argomento, potresti notare quando provi a eseguire questa parte del codice che non verrà inviata al compilatore per noi. Questo perché stiamo lavorando su un codice fittizio, solo per fare pratica e non su un codice vero e proprio. Tutti i codici che passiamo il tempo a scrivere in C devono avere una funzione principale, insieme a qualcos'altro, semplicemente perché è qui che inizierà l'esecuzione di tutti i nostri programmi. Dobbiamo solo aggiungere la funzione principale che vogliamo usare all'interno di quelle parentesi, e siamo a posto.

Ancora una volta, poiché questo sarà un codice sorgente noto come fittizio e uno in cui ci stiamo solo bagnando i piedi e vedendo come consentirà il lavoro, non abbiamo aggiunto il codice necessario a quelle parentesi. Quando eseguiamo questo codice, non ti ritroverai affatto con l'output. Man mano che acquisisci familiarità con la lingua, scoprirai che è possibile aggiungere e rimuovere parti diverse per adattarle a ciò che devi fare lungo il percorso.

## **Scrittura di loop**

I loop sono una bella aggiunta da aggiungere al tuo codice. Sono progettati per accettare una grande quantità di codice e adattarlo a poche righe alla volta. Ad esempio, se desideri scrivere una tabella di moltiplicazione, sarà difficile passare attraverso ogni riga di codice. Ma con un ciclo, possiamo creare un'enorme tabella di moltiplicazione, non importa quanto grande vorremmo che fosse, con solo poche righe di codice.

Esistono diversi tipi di loop con cui siamo in grado di lavorare. La prima opzione su cui ci concentreremo è il ciclo while. Questo ci consentirà di specificare la condizione che farebbe ripetere il ciclo. Dovrebbero esserci anche alcune istruzioni che si ripeteranno nel codice e alcune condizioni che interromperanno il ciclo nel complesso. Se non aggiungi nell'ultima parte, il ciclo si bloccherà e non sarai in grado di fermarlo affatto.

Per aiutarci un po 'qui, daremo un'occhiata al ciclo while. Il codice seguente è un buon esempio di come possiamo lavorare con questo tipo di ciclo:

```
#include <stdio.h>
```

```
Int main()
```

```
{
```

```
    Int x;
```

```
    x = 1;
```

```
    while(x <= 10)
```

```
    {
```

```
        Printf ("% d \ n", x);
```

```
        x ++;
```

```
    }
```

```
    Return(0);
```

```
}
```

La condizione di cui abbiamo bisogno per il ciclo verrà mostrata nella riga 8, ed è impostata in modo che quando il valore della nostra variabile rimane inferiore o uguale al numero dieci, l'istruzione si ripeterà. Ma quando arriviamo alla riga 11, mostra che il valore di x sarà incrementale, quindi aumenterà solo una volta per ogni lop che eseguiamo. Una volta che x è maggiore di dieci, ad esempio undici, il codice vedrà che la condizione non è più vera e smetterà di seguire il ciclo con cui stiamo lavorando.

Sarai in grado di utilizzare questa sintassi per aiutarci a creare tutti i cicli while di cui avremmo bisogno per il nostro codice. Non devi fermarti alle dieci, per esempio. Puoi fare in modo che il codice vada fino a 100 o qualche altro numero, se lo desideri. Devi solo assicurarti che il codice sia impostato per gestire questo. Quindi, con la condizione corretta, il ciclo while continuerà fino a quando la condizione non sarà più soddisfatta.

## **Loop Do / While**

Il secondo tipo di ciclo che siamo in grado di gestire in C è il ciclo do / while. Sarà simile a quello che abbiamo visto con il ciclo while, ma in questo è un po' capovolto. Con il ciclo while, il codice verificherà se la condizione è vera prima ancora di eseguire il codice. Con il ciclo do / while, eseguiamo il codice e verificheremo se la condizione è ancora vera prima di eseguirlo di nuovo. Se stai cercando di eseguire il ciclo almeno una volta, il ciclo do / while è l'opzione migliore. Un buon esempio del ciclo do / while in azione sarà:

```
#include <stdio.h>
```

```

Int main()
{
    Char ch;
    Ch = 'A';

    Do
    {
        Putchar (ch);
        Ch ++;
    }
    While (ch! = 'Z');

    Putchar ('\n ');

    Return(0);
}

```

Noterai che il ciclo do / while inizierà con la riga 9 con l'istruzione "do". questo non ha una condizione perché le istruzioni verranno eseguite una dopo l'altra. Quindi puoi vedere che la condizione while apparirà sulla tua 14a riga. Il ciclo si baserà sulla variabile ch come condizione, e viene prima inizializzato alla riga 7 prima di essere manipolato alla riga 12. Prenditi un momento per compilare e poi esegui questo codice nel tuo compilatore.

## **Il ciclo annidato**

Possiamo anche guardare il ciclo annidato. Ci sono momenti in cui gli altri due cicli saranno utili per le nostre esigenze, ma ci sono momenti in cui avremo bisogno di andare un po' oltre e lavorare con quello che è noto come un ciclo annidato. All'inizio sembrerà complicato, ma sarà più facile di altri cicli, in base a ciò che stai cercando di fare in questo linguaggio di programmazione.

Fondamentalmente, il ciclo annidato sarà quando uniamo insieme due degli altri cicli in uno. Quindi questo ciclo annidato non verrà eseguito finché entrambe le parti non saranno state completate. Un esempio del codice che possiamo usare per questo includerà:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
Int main()
```

```
{
```

```
    Int column, r;
```

```
    Srand ((unsigned)time(NULL));    /*Seed Randomizer*/
```

```
    For(column= 0; column<10; column++)
```

```
    {
```

```
        r = rand ();
```

```
        printf("%2d\t",r% 100);
```

```
    }
```

```
    Putchar('\n');
```



```
    Return(0);  
}
```

Il codice elencato sopra sarà uno che sta per produrre dieci numeri selezionati casualmente. Puoi impiegare un po 'di tempo per costruirlo e quindi eseguire il codice per vedere che dall'output finirà per essere una riga di dieci valori che sono stati selezionati casualmente. La spaziatura tra le variabili verrà eseguita quando useremo il carattere '\ t' che è stato inserito alla riga 14.

## **Le variabili**

In molte delle lingue che imparerai in questa guida, le variabili saranno importanti. Fondamentalmente, questa variabile sarà un nome che viene dato all'area di archiviazione che il programma che creiamo è in grado di manipolare. Ogni variabile in C arriverà con un tipo specifico, che determinerà il layout e la dimensione della memoria della variabile, l'intervallo di valori che stiamo consentendo al programma di memorizzare in quella memoria e l'insieme di operazioni che lasceremo essere applicato a quella variabile.

Scoprirai che il nome della variabile sarà composto da cifre, lettere e dal carattere di sottolineatura. Deve iniziare con un trattino basso o una lettera. E dobbiamo ricordare che in C, dobbiamo preoccuparci della distinzione tra maiuscole e minuscole, quindi fai attenzione alle lettere minuscole e maiuscole con cui stai lavorando quando assegni il nome alle variabili.

Ci sono alcuni tipi di variabili che siamo in grado di lavorare all'interno della nostra codifica. Alcuni dei tipi più comuni di variabili su cui possiamo concentrarci che aiuteranno con il nostro linguaggio C includeranno:

1. Char: questa sarà una variabile che è un singolo ottetto o un byte, ovvero una generica sequenza di otto bit.
2. Int: definisce qualunque dato che rappresenti un sottoinsieme dell'insieme matematico dei numeri interi.
3. Float: questo sarà un valore a virgola mobile con precisione singola.
4. Double: questo sarà un valore a virgola mobile con doppia precisione.
5. Void: questo rappresenterà l'assenza di un tipo.

La programmazione in C è unica in quanto ci darà il tempo di definire alcuni altri tipi di variabili, se lo desideriamo. Ma le cinque opzioni che abbiamo sopra sono alcuni dei modi più comuni in cui vedrai questo gioco.

Possiamo anche esaminare e dichiarare una variabile in C. Una dichiarazione di variabile ci fornirà una certa garanzia che ci sarà una variabile specifica di un dato tipo e nome in modo che il compilatore sia effettivamente in grado di andare avanti e finire fino alla sua compilazione senza richiedere i dettagli completi sulla variabile. La definizione della variabile avrà un significato, ma viene visualizzato solo al momento della compilazione e il compilatore avrà bisogno della definizione della variabile effettiva quando si collega il programma.

Una dichiarazione di variabile sarà la più utile quando si lavora con molti file e si è in grado di definire la variabile in uno dei file che saranno disponibili nel momento in cui si collega il programma. Dovrai utilizzare la parola chiave `extern` per dichiarare la variabile in qualsiasi punto. Sebbene tu possa esaminare e dichiarare la stessa variabile in più di una parte del tuo programma, puoi definirla solo una volta in un blocco di codice, funzione o file.

Possiamo dare un'occhiata a un esempio di come funzionerà. In questo, avremo le variabili dichiarate in alto, ma le definiremo e quindi inizializzeremo quelle variabili all'interno della funzione principale che è importante per noi. Il codice che vedremo per questo include:

```
#include <stdio.h>
```

```
// Variable declaration:
```

```
extern int a, b;
```

```
extern int c;
```

```
extern float f;
```

```
int main () {  
  
    /* variable definition: */  
    int a, b;  
    int c;  
    float f;  
  
    /* actual initialization */  
    a = 10;  
    b = 20;  
  
    c = a + b;  
    printf("value of c : %d \n", c);  
  
    f = 70.0/3.0;  
    printf("value of f : %f \n", f);  
  
    return 0;  
}
```

## **Le funzioni C.**

Anche se le funzioni saranno importanti in tutti i linguaggi che esploriamo, ci vorrà del tempo per vedere come funzionano nel linguaggio C. Una funzione sarà un gruppo di affermazioni che si uniranno ed eseguiranno un compito. Ogni programma in C verrà fornito con, come minimo, una funzione. Questa sarà la funzione `main()`, e anche i programmi più banali e piccoli in questo linguaggio aggiungeranno alcune altre funzioni da definire.

Puoi dividere il tuo codice in funzioni separate. Il modo in cui lo farai - dipende da te, ma vuoi saggiamente dividerlo in modo che la funzione sia in grado di svolgere i compiti che desideri. La dichiarazione di una funzione fornirà al compilatore molte informazioni, incluso il nome della funzione, il tipo restituito e i parametri. E poi, la definizione di una funzione ci fornirà il corpo effettivo della funzione.

La libreria standard C ci fornirà molte delle funzioni integrate che il nostro programma è in grado di richiamare. Ad esempio, se desideri concatenare due stringhe, dovresti usare la funzione `strcat()`, ma se desideri copiare una posizione della memoria in un'altra posizione, allora dovresti usare qualcosa come `memcpy()`.

Dobbiamo quindi vedere come potremmo definire la nostra funzione. La forma generale di una definizione di funzione in questo tipo di linguaggio sarà simile alla seguente:

```
return_type function_name( parameter list) {  
    body of the function  
}
```

Una definizione di una funzione in C prenderà in considerazione il corpo della funzione e un'intestazione della funzione. Alcune delle parti che troveremo nella funzione con cui stiamo lavorando includeranno:

1. The return type: una funzione probabilmente restituirà un valore. Return\_type sarà il tipo di dati del valore che la funzione restituirà. Alcune funzioni eseguiranno le operazioni corrette senza restituire un valore. In questo caso, o con il codice che abbiamo fatto sopra, return\_type sarà la parola chiave del void.
2. Function Name: questo sarà il nome che assegnerai effettivamente alla funzione. Il nome della funzione e l'elenco dei parametri si uniranno per costituire anche la firma della funzione.
3. Parameters: il parametro è più di un segnaposto qui. Quando una funzione è stata invocata, è possibile trasferire il valore al parametro. Questo valore sarà conosciuto come il parametro effettivo o l'argomento. L'elenco dei parametri farà riferimento al numero, all'ordine e al tipo dei parametri di una funzione. Questi sono opzionali ed è possibile avere una funzione nel codice che non contiene parametri.
4. Function body: manterrà una raccolta di istruzioni che definiranno ciò che la funzione è in grado di fare.

## **Uno sguardo al C ++**

Il linguaggio della rete che dobbiamo esaminare è il linguaggio C ++. Questo ha un sacco di storia alle spalle ed è uno dei primi linguaggi di programmazione là fuori. Poiché questo linguaggio è in circolazione da molto tempo, avrà molti programmatori a cui piace usarlo, il che potrà essere utile quando inizierai ad utilizzarlo.

Il C ++ sarà uno di quei linguaggi che ci aiuteranno a scrivere molte delle applicazioni e dei codici che speriamo di progettare durante il processo. Simile a molti degli altri linguaggi di programmazione di cui potresti aver sentito parlare in passato, e ad altri di cui parleremo in questa guida, il C ++ sarà visto come un linguaggio di programmazione orientato agli oggetti, o OOP. Ciò significa che questo linguaggio di programmazione ruoterà attorno a oggetti e classi, piuttosto che ad azioni. Questo renderà il processo di scrittura dei nostri codici un po 'più semplice e garantirà che non ci imbattiamo in altri problemi con cose che si perdono o sono difficili da trovare.

Ci sono molti motivi per cui ai programmatori piace lavorare con il linguaggio C ++ e alcuni di questi includeranno:

1. Dispone di un'ampia biblioteca. Questo ci aiuta a estrarre le funzioni, le variabili e altre parti di codifica che dobbiamo usare, senza dover lottare per trovare la parte giusta al momento giusto.
2. Si integra alla perfezione con righe di codice esterne, cioè con il C ++ potremmo usare dell'altro codice per i nostri programmi. Ci saranno

dei momenti nella tua codifica in cui avrai bisogno di accoppiare C ++ con un altro linguaggio, e questo sarà facile da fare.

3. Funzionerà per la maggior parte dei progetti su cui vuoi dedicare del tempo.
4. È veloce e affidabile.
5. Ci offre un sacco di potenza nel processo in modo da poter gestire quasi tutti i tipi di codice che vorremmo.

### **Creazione del nostro primo codice in C ++**

Con questa piccola introduzione, è tempo per noi di dare un'occhiata a come scrivere un programma in questa lingua. Questo sarà divertente da provare perché ci darà la possibilità di vedere come lavora C ++ e persino di confrontarlo con il linguaggio C di cui abbiamo parlato prima. Per prima cosa, dobbiamo scaricare il linguaggio C ++. Puoi farlo dal sito ufficiale online e scegliere il sistema operativo e altro che corrisponderà alle tue esigenze. Puoi anche scegliere tra altri ambienti e altro in base al tipo di codifica su cui desideri dedicare del tempo.

Una volta che l'ambiente C ++ è stato scaricato e pronto per l'uso, inizieremo a scrivere il nostro primo codice. Faremo un codice semplice e produrrà qualcosa come "Prova questo!" per noi, ma ci insegna ancora alcune delle sintassi di cui avremo bisogno per ottenere i migliori risultati. Il codice che possiamo usare per scrivere questo programma è di seguito:



```
#include <iostream>  
using namespace std;  
int main ()  
{  
    court << "Usa questo!";  
    return 0;  
}
```

Hai alcune opzioni disponibili quando si tratta di scriverlo. Puoi scegliere di scriverlo nel tuo compilatore, che sarà disponibile nel tuo ambiente, oppure puoi scegliere di scriverlo e salvarlo sul tuo computer. La seconda opzione a volte è carina perché in tal caso dovresti salvare il codice e copiarlo e incollarlo a freddo ogni volta che ne avrai bisogno nel tuo codice.

In ogni caso, dovresti considerare attentamente il tipo di editor di testo con cui vuoi lavorare. La maggior parte di essi sarà specifica del dispositivo, quindi dovrai cercare quello che va con il tuo particolare computer. Alcune delle opzioni che puoi utilizzare includono Blocco note di Windows, vlm, vl, Brief ed EMACS. Se desideri avere un editor in grado di funzionare su più di una piattaforma, le opzioni vlm e vl sono le migliori.

## **Le variabili C ++**

Possiamo anche spendere alcune variabili quando si tratta di C ++. Funzioneranno in modo simile alle opzioni che lavoriamo all'interno del linguaggio C, ma verranno definite in modo diverso. Le definizioni del compilatore saranno lì per dire al compilatore quanto spazio di archiviazione è in grado di utilizzare e dove dovrebbe memorizzare e creare la variabile che hai.

Questa definizione è utile anche per aiutare a specificare il tipo di dati che si desidera utilizzare. Sarai in grado di digitare una definizione di variabile come "type variable\_list", quindi scegliere il tipo di variabile che desideri utilizzare e digitarla allo stesso modo. Questi diranno al compilatore che è ora di creare il tipo di variabile che vorresti elencare. Un esempio della codifica che useresti per questo include quanto segue:

```
#include <iostream>  
using namespace std;  
int. main ()  
int j = 10;  
int k = 5;  
int l = j + k  
{  
    cout <<l>> endl;  
    return 0  
}
```

*Dovresti ottenere la risposta 15*

È anche possibile esaminare e non solo definire, ma anche dichiarare le variabili che sono nel programma, ma è qualcosa su cui possiamo lavorare in seguito con un po' di pratica in C ++. È una buona idea fare pratica prima di passare al processo di dichiarazione.

Dedichiamo anche un po' di tempo ad assegnare un valore alla variabile con cui vogliamo lavorare. Se dichiari solo una variabile, tutto ciò che stai facendo è riservare un po' di spazio nella memoria del computer, ma non verrà salvato nulla finché non sarai in grado di assegnarle un valore. È meglio mantenerlo con uno o due valori per la variabile, ma puoi farlo in qualsiasi modo desideri che il codice funzioni.

### **Gli array C++**

Un nuovo argomento che esamineremo sono gli array in C ++. Questi sono un altro tipo di struttura dati che possiamo vedere con C ++ e sono responsabili della memorizzazione di elementi dello stesso tipo e hanno la stessa dimensione fissa. Per mantenerlo semplice, saranno una raccolta dello stesso tipo di variabili. Invece di esaminare e utilizzare le variabili su base individuale, dovresti impostarlo in un array. Per fare ciò, è possibile utilizzare i numeri da 0 a 99, quindi è possibile accedere a ciascuno tramite l'indice dell'array.

Saranno simili alle variabili che si trovano nella tua memoria. Ma scoprirai che funzionano in modo più continuo di quello che possiamo fare con una variabile. Il più basso sarà il primo elemento e quindi l'elemento più alto sarà l'ultimo. Hai anche alcune opzioni per aiutarci in questo. Ad esempio, puoi lavorare con gli array scegliendo di inizializzarli uno per uno, oppure possiamo lavorare con più di un'istruzione per farlo funzionare. Una buona opzione per aiutare con questo è la seguente:

```
double balance [5] = {1000.0, 2.0, 3.4, 17.0, 50.0};
```

Con alcune di queste informazioni in mente e aiutandoci a saperne di più sugli array, abbiamo bisogno di dedicare un po 'di tempo per imparare come possiamo prendere tutto questo e non solo scrivere i nostri array ma anche usare queste informazioni per lavorare con gli array nel programma. Questo programma sarà visto come più avanzato ma farà funzionare alcuni dei codici che abbiamo. Un buon esempio di come siamo in grado di usarlo all'interno del nostro codice include:

```
#include <iostream>  
using namespace std;
```

```
#include <iomanip>  
using std::setw;
```

```
int main ()  
{  
    int n [10]; // n è un array di dieci numeri interi  
// inizializza gli elementi dell'array n su 0  
    for (int i=0; i<10; i++)  
    {  
        n[i]=i+100; // imposta l'elemento nella posizione da i a i + 100  
    }  
    cout << element << setw (13) << valore << endl;  
// restituisce il valore di ogni elemento dell'array
```

```

    for (int j=0; j<10; j++)
    {
        cout << setw(7)<< j << setw(13) << n[j] << endl;
    }
    return 0
}

```

Prima di passare ad alcuni degli altri argomenti che possiamo trattare in questo linguaggio, dobbiamo aprire il compilatore scelto con cui vogliamo lavorare e scrivere questo codice. Questo sarà un ottimo modo per far funzionare l'array e ci darà anche molta pratica. questo programma funzionerà davvero con l'array per ottenere la funzione di setw() attiva e funzionante in modo da poter formattare l'output che siamo in grado di utilizzare.

## La dichiarazione Switch

Dedicheremo un po 'di tempo a esaminare le dichiarazioni condizionali. Entreremo in alcuni di questi in modo più dettagliato in seguito quando esamineremo alcuni degli altri linguaggi di codifica, ma passeremo un po 'di tempo a guardare l'istruzione condizionale che è piuttosto unica nel linguaggio C ++ noto come istruzione Switch.

Innanzitutto, le istruzioni condizionali saranno parti di codice che garantiranno che il codice sia in grado di prendere decisioni da solo, senza che il programmatore debba esaminare e pensare a tutte le possibilità, e aiuteranno il codice a sapere quali decisioni prendere .

Le istruzioni Switch ci aiuteranno a verificare se ci sarà qualche tipo di uguaglianza che si trova nella variabile confrontandola con i casi o i valori che hai delineato nel codice che stai facendo. La variabile che controllerai con questa istruzione verrà confrontata con molti altri casi, a seconda del codice che stai scrivendo. Un buon esempio della sintassi disponibile per queste istruzioni switch includerà:

```
Switch(expression){  
    case constant-expression:  
        statement(s);  
        break; //optional  
    case constant-expression:  
        statement(s);  
        break; //optional  
// puoi aggiungere tutte le dichiarazioni di casi che desideri  
Default: //Optional  
    statement(s);  
}
```

Quando passiamo del tempo a lavorare con queste istruzioni switch, ci saranno alcune regole che dobbiamo seguire per farle funzionare all'interno del nostro codice. La prima cosa qui è che l'espressione che stiamo usando con questa istruzione deve essere una classe integrale o un tipo di classe enumerato. Oltre a tutto questo, apparterrà anche a una classe che avrà una funzione di conversione.

La buona notizia con questo è che quando lavoriamo con C ++, non ci sarà un limite al numero di dichiarazioni di casi che puoi aggiungere, il che può permetterti di avere molta libertà sul lavoro che stai facendo. La cosa importante su cui concentrarsi qui è che ci deve essere un due punti in modo che l'affermazione funzioni nel modo che vogliamo.

### **In che modo C # è diverso**

C # può essere classificato come un linguaggio di programmazione generico che è funzionale, dichiarativo, generico, orientato ai componenti e orientato agli oggetti. Questo linguaggio è stato sviluppato da Microsoft nel 2000 come parte dell'iniziativa .NET e poi è stato approvato da Ecma e ISO per soddisfare lo standard internazionale. È un ottimo linguaggio facile da organizzare, si basa sulle idee di OOP per mantenerlo organizzato e facile da usare e altro ancora.

Come con gli altri due linguaggi, dobbiamo soffermarci un attimo qui e vedere come scrivere codice in C #. Questo ci aiuterà a vedere in che modo il linguaggio C # è simile agli altri linguaggi C e in che modo è diverso. Lavoreremo su un codice di base che utilizza il linguaggio C # e un dispositivo Windows. Il blocco note sarà il nostro editor di testo in questa lingua (purché utilizzi Windows), quindi lo apriremo e digiteremo il codice seguente:

```
classe FirstProgram
{
static void Main ()
{
        Console.WriteLine ("UsingC # is fun.");
    }
}
```

*Ora puoi accedere al prompt dei comandi e digitare:*

*csc FirstProgram.cs*

Dopo aver emesso questo comando, il compilatore per C # sarà in grado di elaborare questo file e creare il file .exe nella directory predefinita. Se hai salvato questo file sul desktop, ad esempio, puoi andare lì e dovresti vedere un nuovo programma chiamato FirstProgram.exe, per sapere che hai fatto bene. Puoi eseguirlo inserendo FirstProgram.exe nel prompt dei comandi e l'output dovrebbe essere "Usare C # è divertente".

## **I tipi di dati C #**



Poiché stiamo lavorando con un linguaggio OOP, dobbiamo concentrarci su alcuni dei diversi tipi di dati comuni in questo linguaggio. Questi sono facilmente suddivisi in due tipi, il riferimento e i tipi di valore. Con i tipi di valore, saremo in grado di trasferire i dati su qualsiasi metodo che stiamo utilizzando. Ma il tipo di riferimento verrà aggiunto in riferimento al metodo invece perché posizioneremo il valore in un'altra posizione.

Ci sono alcuni tipi di dati che siamo in grado di lavorare in questa lingua, inclusi i seguenti:

- Ulong
- Double
- Long
- Float
- UInt
- Int
- Ushort
- Sbyte
- Short
- Byte
- Bool: quando si utilizza questo tipo, i valori che è consentito memorizzare possono essere solo due. I valori saranno falsi e veri, quindi è utile per le istruzioni condizionali e le espressioni logiche.
- Decimal

- Char: questo è un tipo di dati che può contenere un solo carattere. Quando si scrive il valore char, è necessario includerlo tra virgolette singole come scrivere "t" "b" e così via.

## **Creazione di una classe**

In C # siamo in grado di lavorare con classi e oggetti. Questi aiutano a organizzare la lingua che abbiamo e faranno in modo che nulla vada perso nel processo. Daremo uno sguardo agli oggetti tra un momento, ma dobbiamo prima impostare una classe che manterrà questi oggetti. Questa è la bellezza dei linguaggi OOP. Creeranno le classi, che sono come piccoli contenitori, che contengono gli oggetti simili che abbiamo e li tiene tutti in un unico posto per consentirci di accedere facilmente all'interno dei nostri codici.

Per iniziare, quando vogliamo creare una classe, dobbiamo solo usare la parola chiave class. Una volta che siamo in grado di iniziare con il codice e lavorare con quella parola chiave, è tempo di indicare l'identificatore che scegli di utilizzare. Questo sarà fatto insieme ai metodi e alle variabili che vorresti usare per mettere tutto insieme e assicurarti che funzioni. Vogliamo assicurarci che questo sia semplice, assicurandoci anche che accada effettivamente, quindi le parti principali che vedremo con una classe C # includeranno quanto segue

1. Campi: queste saranno tutte le variabili che troveremo appartenenti a un particolare tipo di dati che vogliamo utilizzare.
2. Metodi: questi metodi verranno utilizzati per aiutarci a manipolare i dati in nostro possesso.

3. Proprietà: verranno utilizzate in modo da migliorare la capacità di lavoro dei campi. La proprietà ci fornirà alcune capacità extra per gestire le informazioni e inserirle nei campi come vogliamo.

Ora che sappiamo un po' di più su queste classi e su come funzionano, è tempo di vedere come possiamo effettivamente creare e lavorare con una di queste classi lungo il percorso. Daremo un'occhiata al codice seguente per aiutarci a iniziare con questo e per aiutarci a vedere come dovrebbe funzionare. Per questa classe, chiameremo book e poi aggiungeremo le proprietà size e type per farlo funzionare. Il codice che dobbiamo usare per creare questa particolare classe include:

```
Public class Book  
{  
    Private string bookType;  
    Private string size;  
    Public string BookType  
    {  
        Get  
        {  
            Return this. bookType;  
        }  
        Set  
        {  
            This.bookTyupe = value'  
        }  
    }
```

*Public string Size*

*{*

*Get*

*{*

*Return this size;*

*}*

*Set*

*{*

*this.size = value;*

*}*

*}*

*public Book()*

*{*

*this.bookType = "Dictionary";*

*this.size = "large";*

*}*

*public book(string bookType, string size)*

*{*

*this.bookType = bookType;*

*this.size = size;*

*}*

*public void Sample()*

*{*

*Console.WriteLine(" Is this a {0}, bookType):*

Ora, dobbiamo dare un'occhiata a questo un po '. Quando lo esaminiamo per la prima volta, sembra complicato, ma ci aiuta a vedere meglio la sintassi e la codifica con questo linguaggio e rende più facile vedere come verranno create le classi. Guardando questo, vediamo la classe book e avrà due proprietà definite. Il primo è che descriverà le dimensioni del libro e il secondo è il tipo di libro. Queste proprietà sono impostate per mantenere i valori all'interno dei campi che hanno lo stesso nome, quindi il codice ci aiuterà a dichiarare i due costruttori per la generazione su questa classe.

## **Creazione di oggetti**

Ricordi come abbiamo discusso del fatto che questi linguaggi C sono linguaggi di programmazione orientati agli oggetti? Ciò significa che dobbiamo dedicare un po 'di tempo alla ricerca di come prendere questi linguaggi e creare anche alcuni degli oggetti necessari. questo ci aiuterà ad assicurarci davvero che funzioni come vogliamo. Possiamo quindi esplorare anche come creare oggetti.

Dopo aver creato la classe del libro sopra, siamo in grado di creare effettivamente alcuni oggetti e aggiungerli a quella classe. Questo ci aiuterà a ottenere i migliori risultati possibili e, effettivamente, ad assicurarci che alcune delle classi contengano oggetti. La prima cosa che possiamo fare quando creiamo questa classe è creare l'oggetto. Dobbiamo configurarlo in modo da poter inventare le parole chiave di cui abbiamo bisogno.

Nella maggior parte dei casi, il programmatore sceglierà di assegnare un nuovo oggetto alla sua variabile in modo che rimanga dello stesso tipo di dati della classe con cui vogliamo lavorare. Potrebbe non essere sempre l'opzione migliore, ma ci aiuterà a ottenere una variabile da utilizzare come riferimento all'oggetto che stiamo assegnando. Il semplice codice che possiamo usare qui sarà di seguito:

```
Book someBook = new Book()
```

Questo è un buon esempio perché mette un oggetto semplice che abbiamo fatto prima, e ci aiuterà a prendere l'istanza del libro su cui stiamo lavorando e lo assegnerà alla variabile che è già stata nominata someBook. Questo inserirà quell'oggetto nella classe per noi. Possiamo ripeterlo tutte le volte che ne abbiamo bisogno.

## **I costruttori C #**

La prossima cosa sulla lista con cui possiamo lavorare sono i costruttori. Questi saranno un po 'diversi da quelli di cui abbiamo discusso prima, ma sono comunque importanti per la nostra codifica. Saranno alcuni metodi che vedremo quando verranno eseguiti mentre il programma si imposta per creare un nuovo oggetto. L'idea che vediamo con questi è che sono lì per inizializzare anche i dati di quell'oggetto nuovo di zecca.

Non sarai in grado di ottenere molto valore da mostrare quando usi il metodo del costruttore, e userà lo stesso nome che vediamo con la classe che lo ha chiamato, quindi dovrai dedicare un po 'di tempo ad abbinare un pochi nomi lungo la strada per farlo funzionare. Avrai anche la scelta di passare attraverso e assegnare i parametri giusti a questo quando i costruttori usciranno. I costruttori saranno quindi in grado di impostare i parametri in modo simile agli altri metodi. Puoi anche impostare due o più costruttori nella stessa classe, se lo desideri, ma se lo fai, assicurati che i costruttori provengano da diversi tipi di dati.

Poiché dedichiamo un po 'di tempo a concentrarci su questi costruttori, è sempre importante ricordare che ci saranno alcuni cambiamenti da eseguire quando crei un nuovo oggetto nella classe. Se stai lavorando con una classe che ha più di un costruttore che si trova al suo interno, potresti essere un po 'curioso di capire quale costruttore è in esecuzione mentre l'oggetto viene creato.

Questo linguaggio ci aiuterà a scegliere il costruttore di cui abbiamo bisogno in base a ciò con cui intendi lavorare, quindi non ti assumerai la responsabilità di fornire quelle informazioni. Il compilatore si occuperà di tutto questo perché ti aiuta a fare l'organizzazione senza che tu debba sentirti eccessivamente stressato.

Con alcune di queste informazioni sui costruttori pronte per l'uso e una buona comprensione, diamo un'occhiata a come possiamo gestirle e fare in modo che i costruttori lavorino nel C # sottostante;

*public class Device*

```

{
    private string type;
    private string size;
    // Un costruttore senza alcun parametro
    public Device()
    {
        this.type = "laptop";
        this.size = "large";
    }
    //Un costruttore con 2 parametri.
    {
        this.type = type;
        this.size = size;
    }
}

```

## I cicli C #

Possiamo anche dedicare un po 'di tempo a vedere quanto è facile scrivere loop in linguaggio C #. Proprio come abbiamo fatto con alcune delle altre lingue, ci sono anche alcuni loop disponibili. Ricorda che questi ci consentono di prendere ciò che di solito richiederebbe un bel po 'di righe di codice e ci aiuterebbero a farlo in poche. Mentre i loop che vediamo nel linguaggio C # richiederanno un po 'più di lavoro rispetto a quello che vediamo con Python o qualche altro linguaggio, rispetto alla quantità di lavoro che potrebbero richiedere, sono abbastanza facili da lavorare. Per iniziare, esamineremo il ciclo while di seguito:

*// Programma C # per illustrare il ciclo while*



```

using System;

class whileLoopDemo
{
    public static void Main()
    {
        int x = 1;

        // Esce quando x diventa maggiore di 4
        while (x <= 4)
        {
            Console.WriteLine("GeeksforGeeks");

            // Incrementa il valore di x per
            // la prossima interazione
            x++;
        }
    }
}

```

Questo è un buon ciclo da imparare perché ci ha aiutato a scrivere le parole GeekforGeeks tre volte. Possiamo sicuramente farlo e farlo scrivere più volte se lo desideriamo, ma questo è un buon inizio. C'è il ciclo while, insieme alla condizione che afferma come, quando l'elenco di quella frase è più di 3 volte, dovrebbe fermarsi. Ciò garantisce che non rimaniamo bloccati in un ciclo infinito con la frase GeekforGeeks che viene visualizzata un milione di volte e blocca i nostri computer.

Oltre al ciclo while che abbiamo fatto sopra, è possibile gestirne alcuni anche in questo linguaggio. Questo utilizza un'opzione leggermente diversa, ma noterai alcune somiglianze con il ciclo while. Un buon esempio di come funzionerà il ciclo for in C # include quanto segue:

```
// Programma C # per illustrare un ciclo.  
using System;  
  
class forLoopDemo  
{  
    public static void Main()  
    {  
        // il ciclo for inizia quando x = 1  
        // e viene eseguito fino a x <= 4  
        for (int x = 1; x <= 4; x++)  
            Console.WriteLine("GeeksforGeeks");  
    }  
}
```

Con questo, stiamo anche lavorando con quelli che sono noti come loop controllati dall'uscita. Questo ci aiuterà a esaminare e testare la condizione, ma è mostrata alla fine del corpo del ciclo. Ciò significa che il ciclo verrà eseguito da solo almeno una volta, quindi verificherà se la condizione è stata soddisfatta. Con l'altro ciclo, se la condizione è soddisfatta dall'inizio, non passerà mai attraverso il ciclo. Questo a volte può fare la differenza nel tipo di codici che stai scrivendo.

Un'altra cosa che noteremo con il linguaggio C #, che potrebbe non essere visualizzata in alcuni degli altri tipi di linguaggi, è il ciclo infinito. Questi vanno ai cicli che scopriranno che la condizione di test non valuterà nulla come falso in qualsiasi momento. Ciò significa che quando lavoriamo con loro, continueranno a eseguire l'affermazione che hai inserito fino a quando non utilizzerai una sorta di forza esterna su di loro.

Come possiamo immaginare, non ci sono molte situazioni in cui lavorerai con questo, ma è comunque una buona idea sapere che aspetto ha. Il codice che vedremo quando lavoriamo con un ciclo infinito in C # includerebbe:

```
// Programma C # per dimostrare il ciclo infinito  
using System;  
  
class infiniteLoop  
{  
    public static void Main()  
    {  
        // L'istruzione verrà stampata  
        // infinite volte  
        for(;;)  
            Console.WriteLine("Questo è stampato infinite volte");  
    }  
}
```

E infine, dobbiamo anche dare un'occhiata al ciclo do-while. Questo sarà simile al ciclo while, ma sarà una di quelle opzioni controllate dall'uscita, e il ciclo while non lo è. Alcune delle codifiche che siamo in grado di fare con il ciclo do-while che stiamo guardando qui sono di seguito:

```
// Programma C # per illustrare il ciclo do-while
```

```
using System;
```

```
class dowhileloopDemo
```

```
{
```

```
    public static void Main()
```

```
    {
```

```
        int x = 21;
```

```
        do
```

```
        {
```

```
            // La riga verrà stampata uniforme
```

```
            // Se la condizione è falsa
```

```
            Console.WriteLine("GeeksforGeeks");
```

```
            x++;
```

```
        }
```

```
        while (x < 20);
```

```
    }
```

```
}
```

Come possiamo vedere in questo capitolo, ci sono molte cose fantastiche che possiamo fare quando è il momento di utilizzare i linguaggi C, sia che si parli di C, C ++ o C #. Condivideranno molte somiglianze tra loro, ma ci sono momenti in cui avremo bisogno di passare attraverso e cambiare la sintassi e il metodo che usiamo su di loro. Ma quando hai bisogno di un programma ad alta potenza per portare a termine il lavoro, queste sono sicuramente le lingue che dovresti prendere in considerazione.

## **Capitolo 4: Come programmare in Python**

Il prossimo linguaggio a cui daremo un'occhiata è il linguaggio Python. Quando si tratta di un linguaggio multiuso in grado di gestire tutto il lavoro che vogliamo fare con la codifica, è stato progettato pensando ai principianti, prendendo alcuni dei codici su cui ci siamo concentrati nel linguaggio C e facendo più facile lavorare con loro.

Ci sono molti motivi diversi per cui vorremo lavorare con il linguaggio Python. È una delle migliori opzioni da scegliere e ci vorrà del tempo per esaminare alcuni dei diversi codici forniti con questo linguaggio in modo da poter vedere quanto sia semplice codificare in Python. Immergiamoci un po 'di più in questo linguaggio e vediamo cosa siamo in grado di fare per farlo funzionare per le nostre esigenze.

## **I vantaggi di Python**

La prima cosa a cui dobbiamo dare un'occhiata quando si tratta del linguaggio Python sono alcuni dei vantaggi di cui siamo in grado di godere quando si tratta di usarlo. Ci sono molti linguaggi di codifica là fuori, e gli abbiamo già dato una sbirciata. Concentriamoci proprio sui punti di forza.

Il primo vantaggio è sicuramente la semplicità, e la velocità di apprendimento. Questo linguaggio è stato progettato per aiutare i principianti a iniziare con la programmazione. Potrebbe non essere complicato lavorare con gli altri linguaggi macchina, ma questo lo rende perfetto anche quando è in assoluto la prima volta che ci si affaccia a questo fantastico mondo.

Un altro vantaggio è la sua estrema potenza e versatilità. Anche se è un linguaggio progettato per i principianti, scoprirai le grandi cose che si potranno fare. Questo può aiutarci a fare qualsiasi cosa, dalla codifica di base, fino alla codifica che è più difficile, come ciò che vediamo con il deep learning e l'apprendimento automatico e anche con un po' di intelligenza artificiale.

Anche la comunità che sta dietro a Python è forte. Scopriremo che questa comunità è in grado di aiutarci quando ci imbattiamo in alcuni dei problemi e dei bug che derivano dalla codifica. Come principiante, commetterai degli errori e incontrerai anche alcuni problemi lungo la strada. Essere in grado di fare affidamento su questa comunità renderà un po' più facile lavorare con alcuni dei codici in generale. Possono rispondere alle tue domande, mostrarti alcuni nuovi metodi per gestire le diverse parti del codice e possono persino rendere più facile per noi imparare qualcosa di nuovo.

Anche il linguaggio Python è considerato un linguaggio OOP. Ciò significa che è progettato per funzionare sulla base di classi e oggetti. Le classi saranno come i contenitori che usiamo con questo linguaggio, e quindi gli oggetti saranno le diverse parti che aggiungeremo all'interno dei contenitori. Questo è qualcosa che troveremo in molti dei migliori linguaggi di programmazione, è importante capire come funziona. Fondamentalmente mantiene il codice bello, organizzato e pulito garantendo che possiamo estrarre le parti giuste del codice quando sarà necessario.

E il vantaggio finale che impiegheremo del tempo per esplorare in questo capitolo su Python è come è in grado di funzionare bene anche con altri linguaggi di codifica. Mentre le librerie fornite con Python sono sorprendenti e avranno molte delle funzionalità che stai cercando, e ci sono molte librerie ed estensioni con cui siamo in grado di lavorare che possono aumentare la funzionalità di Python, ci saranno anche alcune volte quando questo non si allineerà e Python ha bisogno di una piccola spinta in più.

Ad esempio, Python può funzionare molto bene con C # e C ++ quando è il momento di lavorare con alcune delle diverse cose dell'apprendimento automatico. Questo è solo uno dei posti in cui Python ha bisogno di un piccolo aiuto extra. Con gli strumenti giusti, siamo in grado di scrivere il codice in Python, aggiungere quegli strumenti e quindi assicurarci di essere in grado di farlo funzionare con l'altro linguaggio lungo la strada.

## **Creare la tua classe**

Il fatto che Python sia considerato un linguaggio OOP sarà importante quando si tratterà di lavorarci e di farlo comportare nel modo che vorresti. Quando ci concentriamo su questi linguaggi OOP, otteniamo molta organizzazione e la codifica è più facile poiché stiamo lavorando con qualcosa che si basa su oggetti e pensieri concreti, piuttosto che astrazioni come alcuni dei vecchi linguaggi di codifica là fuori.

Uno dei componenti principali che aiuteranno a guidare questa idea sono le classi. Queste classi sono come piccoli contenitori che useremo per trattenere tutti gli oggetti che creiamo e cosa far funzionare all'interno dei nostri codici. Quando saremo in grado di creare alcune di queste classi, la scrittura del codice diventerà molto più semplice e otterremo risultati migliori che mai.



Per aiutarci a creare una di queste classi, tuttavia, è importante intervenire e scegliere le parole chiave giuste in modo da poter nominare correttamente la classe. La buona notizia qui è che siamo in grado di dare alla classe qualsiasi nome desideriamo. Dobbiamo solo assicurarci che il nome possa essere visualizzato subito dopo la parola chiave e che sia qualcosa di facile da ricordare in modo da poterlo richiamare in seguito.

Una volta che abbiamo scelto il nome che vorremmo dare alla classe, è il momento di esaminare e nominare le sottoclassi. Questa è la parte che sarà all'interno della parentesi della classe che mostreremo un po' più avanti. Assicurati che quando sei vicino alla fine di questa prima riga, dopo aver creato la classe, aggiungi un punto e virgola per aiutare a finire tutto. Anche se questo è qualcosa di cui non abbiamo bisogno tecnicamente e puoi scrivere il codice senza di esso, la maggior parte dei programmatori lo vedrà come il modo tradizionale e corretto per scrivere codice, quindi è spesso meglio aggiungerlo.

Scrivere alcune delle classi che desideri avere in Python sembrerà un po' complicato a questo punto, ma la buona notizia è che ci sono alcuni passaggi che siamo in grado di utilizzare. Siamo in grado di seguire il codice seguente per avere una buona idea di come impostare una di queste classi e come sarà il processo di creazione della nostra classe nel processo:

```
class Vehicle(object):  
#constructor  
def __init__(self, sterzo, ruote, frizione, freni, marce):  
self._steering = sterzo  
self._wheels = ruote
```

```

self._clutch = frizione
self._breaks = freni
self._gears = marce
#destructor
def _del_(self):
    print("Questo è il distruttore....")

#membri funzioni o metodi
def Display_Vehicle (self):
    print ('Sterzo:', self._steering)
    print ('Ruote:', self._wheels)
    print ('Frizione:', self._clutch)
    print ('Freni:', self._breaks)
    print ('Marce:', self._gears)
#istanza un'opzione veicolo
myGenericVehicle = Vehicle('Power Steering', 4, 'Super Clutch', 'Disk
Breaks', 5)

myGenericVehicle.Display_Vehicle()

```

Sembra piuttosto lungo, ma in realtà abbiamo creato la classe con cui vogliamo lavorare aggiungendo diversi oggetti al mix per farlo funzionare per le nostre esigenze. Con questo in mente, però, dobbiamo essere in grado di capire come accedere ad alcuni degli oggetti che sono nella nostra classe scelta. Abbiamo già svolto la lezione e aggiunto alcuni degli oggetti con cui vogliamo lavorare. Ora è il momento di concentrarci su alcuni dei passaggi che dobbiamo utilizzare per accedervi effettivamente per farli funzionare per le nostre esigenze.

Il metodo migliore da usare, sebbene ci siano alcune opzioni tra cui possiamo scegliere è con il metodo di accesso. Il codice che possiamo usare per trasformarlo in qualcosa con cui possiamo lavorare include:

```
class Gatto(object)  
    itsAge = None  
    itsWeight = None  
    itsName = None  
    #set funzione di accesso da utilizzare per assegnare valori ai campi o  
alle variabili membro  
    def setItsAge(self, itsAge):  
        self.itsAge = itsAge  
  
    def setItsWeight(self, itsWeight):  
        self.itsWeight = itsWeight  
  
    def setItsName(self, itsName):  
        self.itsName = itsName  
  
    #get funzione di accesso da utilizzare per restituire i valori da un  
campo  
    def getItsAge(self):  
        return self.itsAge  
    def getItsWeight(self):  
        return self.itsWeight  
  
    def getItsName(self):
```

```
return self.itsName
```

```
objFrisky = Gatto()  
objFrisky.setItsAge(5)  
objFrisky.setItsWeight(10)  
objFrisky.setItsName("Frisky")  
print("Il suo nome è:", objFrisky.getItsname())  
print("La sua età è:", objFrisky.getItsAge())  
print("Il suo peso è:", objFrisky.getItsName())
```

## **Scrittura delle variabili**

Le variabili sono una grande parte del codice su cui dobbiamo dedicare un po' di tempo quando siamo nel linguaggio Python. Più siamo in grado di guardarli e imparare da questo lavoro, meglio è per noi. Queste variabili, per cominciare, saranno solo piccole parti nella memoria che salveremo e utilizzeremo per mantenere alcuni dei diversi valori che vorremmo essere in grado di memorizzare.

Tuttavia, dobbiamo assicurarci di esaminare e aggiungere valore alla variabile. Non ci farà molto bene se passiamo e non aggiungiamo alcun valore qui. Ciò significa che la variabile manterrà semplicemente un punto vuoto nella memoria del nostro computer, e questo non sarà poi così efficiente. Ma quando assegniamo un valore, e talvolta assegneremo più di un valore alla stessa variabile, dice al computer di posizionare quel valore proprio in quella posizione e puoi facilmente recuperarlo in un secondo momento se lo desideri.

La buona notizia qui è che siamo in grado di assegnare qualsiasi valore che vorremmo a una variabile, purché utilizziamo un semplice simbolo. E questo è il segno di uguale. Finché abbiamo un segno di uguale tra il valore e la variabile, siamo tutti pronti e il computer posizionerà quel valore dove vorremmo. È così semplice lavorare con le variabili in Python

## **I loop di Python**

Abbiamo dato un'occhiata ad alcuni dei loop che sei in grado di fare in alcuni degli altri linguaggi esistenti, quindi ora è il momento di vedere come possiamo lavorare con i loop di Python. Questi hanno gli stessi scopi che vedremo con i loop in altri linguaggi di programmazione, quindi dovremmo avere familiarità con questo ormai.

Il primo ciclo che possiamo fare in Python è il ciclo while. Questo ciclo sarà buono da individuare quando sapremo dall'inizio quante volte abbiamo bisogno del ciclo per eseguire le sue iterazioni. Lo useresti quando vuoi creare un codice che può contare fino a dieci, per esempio. Assicurati con questo che ci sia una condizione alla fine che dirà al codice quando fermarsi; altrimenti, rimarrai bloccato in un ciclo e non sarai in grado di sbloccare il computer senza interrompere l'intero programma. Il codice che puoi usare per il ciclo while include:

*#calcolo di semplice interesse. Chiedere all'utente di inserire capitale, tasso di interesse, numero di anni.*

```
counter = 1  
while(counter <= 3):
```

```

principal = int(input("Immettere l'importo principale:"))
numberofyears = int(input("Immettere il numero di anni:"))
rateofinterest = float(input("Immettere il tasso di interesse:"))
simpleinterest = principal * numberofyears * rateofinterest/100
print("Simple interest =%.2f" % simpleinterest)
#incrementa il contatore di 1
counter = cunter + 1
print ("Hai calcolato l'interesse semplice per 3 volte!")

```

Quindi siamo in grado di passare al ciclo n Python. Questo è un po' diverso nel modo in cui funziona, ed è spesso visto come la forma più tradizionale di creare loop in Python. Se non sei sicuro del metodo con cui lavorare e sei preoccupato di apprendere tutti questi cicli, devi concentrarti sul ciclo for.

Con questi cicli for, non consentiremo all'utente di passare attraverso e fornire nessuna delle informazioni necessarie al codice per l'avvio del ciclo. Piuttosto, con questo ciclo, Python verrà impostato in modo che possa passare attraverso le iterazioni in qualsiasi ordine tu le scelga e le inserisca nel codice. L'utente non avrà bisogno di aggiungere l'input perché il ciclo passerà semplicemente attraverso l'iterazione impostata al momento. Un buon esempio di come siamo in grado di lavorare con il ciclo for include

```

# Misura alcune stringhe:
words = ["mela", "mango", "banana", "arancia"]
for w in words:
    print(w, len(w))

```

Scrivi questo codice nel tuo compilatore e poi eseguilò. Il ciclo for farà in modo che tutte le parole nella riga sopra di esso vengano visualizzate sullo schermo, esattamente come le hai scritte. Se vuoi che siano in un ordine diverso, devi farlo mentre lavori sul codice, non più tardi. Puoi aggiungere qualsiasi parola o altra informazione che desideri mostrare in questo tipo di ciclo, assicurati solo di averlo nell'ordine giusto dall'inizio.

L'ultimo tipo di ciclo che siamo in grado di esaminare è noto come ciclo annidato. È qui che siamo in grado di combinare insieme due dei loop che abbiamo prima. Entrambi inizieranno insieme e non si completeranno finché non saranno completati entrambi. Questo può essere utilizzato in situazioni come la creazione della nostra tabella di moltiplicazione. Alcune delle codifiche che possiamo usare per aiutarci a creare alcuni di questi cicli annidati includeranno:

*#scrivi una tabella di moltiplicazione da 1 a 10*

*For x in xrange(1, 11):*

*For y in xrange(1, 11):*

*Print '%d = %d' % (x, y, x\*y)*

Quando hai ottenuto l'output di questo programma, sarà simile a questo:

1\*1 = 1

1\*2 = 2

1\*3 = 3

1\*4 = 4

Fino a  $1*10 = 10$

Quindi passerebbe a fare il tavolo a due in questo modo:

$$2*1 = 2$$

$$2*2 = 4$$

E così via fino a quando non avrai  $10*10 = 100$  come punto finale della sequenza

Ogni volta che è necessario eseguire un ciclo all'interno di un altro ciclo, il ciclo annidato sarà in grado di aiutarti a farlo. Puoi combinare insieme il ciclo for, il ciclo while o ogni combinazione in base a ciò che vuoi ottenere all'interno del codice. Ma ti mostra sicuramente quanto tempo e spazio all'interno del codice possono risparmiare questi loop. La tabella di moltiplicazione sopra ha richiesto solo quattro righe per essere scritta e hai ottenuto una tabella enorme. Pensa a quanto tempo ci vorrebbe se dovessi scrivere ogni parte della tabella!

Il ciclo for, il ciclo annidato e il ciclo while saranno tutti tipi di ciclo comuni con cui siamo in grado di lavorare come principianti e possono aiutarci a ripulire alcune delle codifiche che stiamo facendo con Python nel primo posto. Sapere come funzionano e come siamo in grado di metterli insieme farà un mondo di differenza in alcuni dei codici che possiamo fare, può ripulire le cose ed è sicuramente un vantaggio quando si tratta di fare parte di questo lavoro in Python.



## L'importanza delle dichiarazioni condizionali

Ci sono tre tipi principali di istruzioni condizionali che siamo in grado di lavorare all'interno del linguaggio Python. Ognuno interverrà e ci aiuterà anche a svolgere più lavoro, in base a quali sono i nostri obiettivi e come dovrebbe funzionare insieme. Le tre affermazioni principali includeranno l'istruzione `if`, l'istruzione `if else` e l'istruzione `elif`.

La cosa bella di queste istruzioni condizionali è che sono impostate per assicurarsi che il codice sia in grado di prendere alcune decisioni, da solo, in base alle condizioni che hai impostato dall'inizio. Ciò rende il programma più interattivo e può aiutarlo a rispondere all'utente, anche quando non sei in grado di fare previsioni su tutto ciò che accadrà.

Per prima cosa, esamineremo l'istruzione `if`. Questo ci aiuterà a dare una risposta solo se l'input che otteniamo dall'utente è vero. Questo ci consentirà di mantenere il programma in esecuzione quando le nostre condizioni sono soddisfatte, ma può fornire alcune limitazioni quando lavoriamo con una situazione in cui l'utente non inserisce nella dichiarazione che volevamo basarci sulle nostre condizioni. Vedremo come risolvere il problema tra un attimo, ma il nostro primo obiettivo qui è guardare un esempio dell'istruzione `if`. Possiamo vedere come funziona di seguito

```
età = int(input ("Inserisci la tua età:"))
```

```
if(età <= 18):
```

```
    print("Non sei idoneo per il voto, prova le prossime elezioni!")
```

```
print("Il programma termina")
```

Ora, con questo, se l'utente mette la propria età come minore o uguale a 18, riceverà il messaggio che è tra parentesi sotto. Ma cosa succede quando l'utente entra e ha 42 anni o qualche altra età superiore a 18? L'istruzione `if` non sarà in grado di gestirla. Questo è il motivo per cui possiamo passare all'istruzione `if else`.

L'istruzione `if else` è molto più piacevole da usare in Python perché ci aiuta a gestire qualsiasi risposta che l'utente ci consegna. Ciò significa che se inserissero la loro età in 42 anni, l'altra affermazione sarebbe in grado di gestirla. Possiamo anche ampliarlo, consentendo diverse azioni o messaggi per una varietà di risposte, ma lo manterremo piuttosto semplice in questo momento.

Possiamo aggiungere l'istruzione `if-else` al codice che abbiamo fatto sopra. Questo ci aiuta a far accadere qualcosa al nostro programma, indipendentemente dal fatto che l'utente dica che la sua età è maggiore o minore di 18 anni. Il codice che siamo in grado di utilizzare per farlo accadere nel nostro codice Python include:

```
età = int(input ("Inserisci la tua età:"))  
if(età <=18):  
    print("Non sei idoneo per il voto, prova le prossime elezioni!")  
else  
    print("Congratulazioni! Puoi votare. Controlla il tuo seggio elettorale  
locale per saperne di più!)  
print("Il programma termina")
```

come possiamo già vedere qui, questa è un'opzione molto migliore con cui lavorare. Può gestire qualsiasi risposta che l'utente abbia e fornirà loro qualche tipo di informazione indipendentemente da ciò che usano per il programma.

Possiamo portare questo al livello successivo, se lo desideriamo, e questo includerebbe le dichiarazioni elif. Questi sono un po' simili a ciò che vedremo con l'istruzione if else, ma presenteremo le risposte all'utente. Riceveranno un tipo di menu, simile a quello che troveremo nei giochi, e quindi potranno scegliere tra una di quelle opzioni. Un buon esempio di come funzioneranno queste dichiarazioni elif sarà di seguito:

```
print("Godiamoci una pizza! Ok, entriamo in Pizzahut!")
print("Cameriere, seleziona la pizza che preferisci dal menu")
pizzachoice = int(input("Inserisci la tua scelta di Pizza:"))
if pizzachoice == 1:
    print('Voglio godermi una pizza napoletana')
elif pizzachoice == 2:
    print("Voglio godermi una pizza rustica")
elif pizzachoice == 3:
    print('Voglio gustare una pizza capricciosa')
else:
    print("Mi spiace, non voglio nessuna delle pizze elencate, per favore
portami una Coca Cola.")
```

Queste istruzioni condizionali saranno ottime opzioni che ci aiuteranno ad assicurarci che i nostri codici siano in grado di gestire le decisioni per noi senza doversi preoccupare della programmazione per tutti i diversi input che l'utente può provare a inserire. Ciascuno di questi può essere usati in diversi tipi di codici che vogliamo scrivere in Python e imparare come funzionano ognuno, e quando vogliamo aggiungerli ad alcuni dei nostri codici può essere importante.

## **Le eredità di Python**

L'ultima cosa con cui lavoreremo, ed è una caratteristica unica quando guardiamo un linguaggio OOP come Python, è l'idea di un'eredità. È qui che possiamo prendere parte del codice su cui abbiamo lavorato in precedenza nel programma e riutilizzarlo. Quindi la seconda parte, nota come classe derivata, può essere modificata, aggiunta, rimossa e altro ancora e non verrà emessa con la classe originale o la classe base.

Questo ci aiuta a mantenere un certo ordine nella codifica che facciamo e manterrà le cose organizzate, ci consente di riutilizzare alcune delle funzionalità che vogliamo e ci impedisce di dover riscrivere le stesse parti del codice più e più volte. Questo ci renderà più facile mantenere il codice organizzato e pronto per lavorare.

Durante il processo di creazione di una di queste eredità, dovremo prendere il nostro codice genitore originale e poi copiarlo, quindi è pronto per andare in un'altra parte del nostro programma. Questo nuovo codice sarà conosciuto come codice derivato o codice figlio, ed è uno di quelli a cui siamo in grado di apportare tutte le modifiche che vorremmo. Nessuna di queste modifiche avrà alcun effetto sul codice genitore, quindi non dovrai preoccuparti di questo. A volte basta eseguire questo processo una volta, ma altre volte potrebbe essere necessario farlo un paio di volte, e questo è possibile anche con le eredità.

Per aiutarti a dare più senso a queste eredità e come funzioneranno tutte e come sono così importanti per il codice mantenendolo pulito e organizzato, il tutto risparmiando tempo, dobbiamo guardare a un esempio di come funzionerà un'eredità in Python di seguito:

*#Esempio di eredità*

*#base class*

*class Student(object):*

*def \_\_init\_\_(self, name, rollno):*

*self.name = nome*

*self.rollno = rollno*

*#Graduate class eredita o deriva dalla classe Student*

*class GraduateStudent(Student):*

*def \_\_init\_\_(self, name, rollno, graduate):*

*Student.\_\_init\_\_(self, name, rollno)*

*self.graduate = laureato*

*def DisplayGraduateStudent(self):*

```
print("Nome dello studente:", self.name)
print("Numeri degli studenti:", self.rollno)
print("Gruppo di studio:", self.graduate)
```

*# La classe Post Graduate eredita dalla classe Student*  
*classe PostGraduate(Studente):*

```
def __init__(self, name, rollno, postgrad):
    Student.__init__(self, name, rollno)
    self.postgrad = postgrad
```

```
def DisplayPostGraduateStudent (self):
    print("Nome dello studente:", self.name)
    print("Numeri degli studenti:", self.rollno)
    print("Gruppo di studio:", self.postgrad)
```

*#istanzia dai corsi di laurea e post laurea*

```
objGradStudent = GraduateStudent("Mainu", 1, "MS-Mathematics")
objPostGradStudent = PostGraduate("Shainu", 2, "MS-CS")
objPostGradStudent.DisplayPostGraduateStudent()
```

Quando lo digiti nel tuo interprete, otterrai i risultati:

```
("Nome studente:", "Mainu")
("Numero studenti:", 1)
("Gruppo di studenti:", "MSC-Matematica")
("Nome studente:", "Shainu")
("Numero studenti:", 2)
("Gruppo di studenti:", "MSC-CS")
```

Potrebbe sembrare un sacco di codice con cui lavorare, ma in realtà si prende cura di molte parti diverse contemporaneamente. Stiamo scrivendo la parte originale del codice che vogliamo utilizzare, quindi spostandola verso il basso e apportando alcune modifiche, il tutto ottenendo ancora alcune nuove cose aggiunte o tolte in modo che il codice si comporti nel modo che vorresti .

Lavorare con il linguaggio Python è un'ottima scelta, indipendentemente dal tipo di codice su cui vorresti concentrare i tuoi sforzi. È semplice da usare pur continuando a fornirci parte della potenza di cui abbiamo bisogno per gestire alcune delle altre attività che desideri, come l'intelligenza artificiale e l'apprendimento automatico, solo per citarne alcuni. Quando sei pronto per saperne di più sul linguaggio Python e iniziare a usarlo per alcune delle tue esigenze, assicurati di controllare i codici sopra per renderlo più facile.

## **Capitolo 5: Programmazione SQL**

L'ultima lingua su cui dedicheremo del tempo è SQL. Questo sta per il linguaggio di query strutturato e ci aiuterà ad accedere e manipolare qualsiasi database con cui vogliamo lavorare. Questa è una delle opzioni più comuni con cui siamo in grado di lavorare quando si tratta di gestire uno qualsiasi dei database che vorremmo, sebbene ci siano anche altre opzioni nei linguaggi di codifica che possono aiutare. Fa anche parte degli standard stabiliti dalle organizzazioni ANSI e ISO lungo il percorso.

Ci sono molte cose con cui siamo in grado di utilizzare questo linguaggio SQL per aiutarci. Ad esempio, ha lo scopo di aiutarci a svolgere molto del lavoro di cui abbiamo bisogno con il database. Ci aiuterà ad eseguire query sul nostro database, ottenere dati, inserire nuovi record o aggiornare i record già presenti, eliminare alcuni record se non ne abbiamo più bisogno, creare un nuovo database o alcune delle tabelle che si trovano all'interno del database, memorizzare tutte le procedure che vogliamo in quel database, creare alcune nuove viste e persino impostare alcune delle autorizzazioni di cui abbiamo bisogno in relazione al database.

Come possiamo vedere, ci sono un sacco di cose che vedremo accadere quando si tratta di lavorare con la programmazione SQL. Siamo in grado di usarlo per aiutarci con tutto ciò che riguarda il database con cui stiamo lavorando, e questo fa parte di ciò che lo rende così potente.

Poiché così tante aziende stanno raccogliendo dati da così tante fonti e molte volte questi database sono necessari solo per gestire il sito Web e altre informazioni sui clienti per l'azienda, è fondamentale avere qualcosa che sia semplice ed efficace quando è il momento di guardare attraverso e utilizzare quel database. Il linguaggio di programmazione SQL sarà in grado di gestire parte del lavoro che stiamo facendo qui e può rendere un po' più semplice qualsiasi lavoro che vogliamo fare con questo database.



## I vantaggi di SQL

Da qui, dobbiamo dare un'occhiata ad alcuni dei vantaggi che vedremo quando sarà il momento di lavorare con l'opzione di programmazione SQL. Questo è un linguaggio unico in quanto è progettato per funzionare bene con alcuni dei database che vogliamo gestire lungo il percorso. Ma dobbiamo essere consapevoli di come funziona e perché è considerata una delle migliori opzioni, tra tante, quando è il momento di gestire anche i nostri database.

Ci sono numerosi vantaggi che vedremo quando si tratterà di lavorare con il linguaggio SQL, ed è importante per noi riconoscerne alcuni e imparare come possiamo farli funzionare per le nostre esigenze. Alcuni dei principali vantaggi che vedremo con questo linguaggio includono:

1. Non è necessario lavorare con la codifica: è possibile lavorare con l'SQL standard e passare attraverso il sistema di database, eseguendo tutte le attività che si desidera, senza dover utilizzare molta codifica. In effetti, la maggior parte delle cose che vuoi fare con SQL richiederà solo poche righe o meno di codice in questo linguaggio.
2. Gli standard sono ben definiti: gli standard seguiti in SQL esistono da molto tempo e questo li rende più affidabili e più facili da utilizzare. Nessuno di questi standard si trova nei database che non si basano su SQL.
3. È portatile: è possibile utilizzare SQL in qualsiasi programma che si trovi su telefoni cellulari, laptop, server e PC, se lo si desidera.

4. Linguaggio interattivo: questo linguaggio di dominio verrà utilizzato quando vorremo comunicare con il database e quindi ricevere risposte alle domande più complesse, facendo tutto questo in pochi secondi.
5. Viste multiple dei dati: con l'aiuto di questo linguaggio, gli utenti sono in grado di fare diverse viste delle strutture del database e dei database per i diversi utenti.

Questi sono solo alcuni dei vantaggi che puoi vedere quando è il momento di lavorare con questa lingua in generale. E non ci vorrà molto a lavorare con esso, insieme al tuo database, per capire quanto bene questo può aiutarci a portare a termine il lavoro. Sia che tu stia creando alcune delle tue tabelle, modificando il database o eseguendo alcune delle altre opzioni consentite qui, scoprirai che SQL sarà un ottimo linguaggio di programmazione su cui concentrarti.

## **Comandi comuni in SQL**

Lavorare con il linguaggio SQL sarà leggermente diverso rispetto ad alcuni degli altri linguaggi. Ci sono spesso comandi di una o due parole che sono in grado di gestire tutto ciò che vogliamo fare quando si tratta di lavorare sui nostri database con questo linguaggio. Alcuni dei comandi più comuni su cui possiamo concentrarci con SQL includeranno:

1. **ALTER TABLE**: questo renderà più semplice l'aggiunta di colonne alla tabella specifica su cui ti stai concentrando nel database.

2. **AND:** Questo sarà uno degli operatori che possiamo usare che è impostato per combinare insieme due condizioni. Entrambe le condizioni devono essere vere per la riga prima che venga aggiunta con il set di risultati.
3. **AS:** Questa sarà una parola chiave in questa lingua che ci aiuterà a rinominare una colonna, o anche una tabella, con un alias per tenerla al sicuro.
4. **AVG():** questa sarà una funzione aggregata che restituirà per noi il valore medio che vediamo in una colonna numerica.
5. **BETWEEN:** questo sarà un operatore utilizzato per filtrare il set di risultati in modo che rientri in un determinato intervallo. I valori che otterremo con questo sono date, testo o numeri.
6. **CREATE TABLE:** vedremo questa opzione apparire nella sezione successiva quando esploreremo cosa siamo in grado di fare con alcuni dei processi di creazione di una tabella. Ma sarà quello che useremo per creare una nuova tabella nel nostro database. Ci aiuta anche a specificare quale nome vorremmo dare alla tabella e possiamo persino nominare le colonne che si trovano nella tabella.
7. **DELETE:** questo sarà quello che useremo quando vogliamo rimuovere alcune delle righe nella nostra tabella.

8. GROUP BY: Questa sarà una clausola SQL che verrà utilizzata quando avremo le funzioni aggregate. Verrà spesso utilizzato insieme all'istruzione SELECT per organizzare alcuni dei dati che abbiamo identici in gruppi per funzionare un po 'meglio.
9. INSERT: questa sarà l'affermazione che useremo quando sarà il momento di aggiungere una nuova riga a qualunque tabella stiamo lavorando.
10. IS NULL/IS NOT NULL: questi saranno gli operatori che possiamo usare per verificare se ci sono alcuni valori che sono visti come vuoti. Questo si baserà sulla clausola WHERE.
11. LIKE: Questo è in realtà un tipo speciale di operatore che useremo insieme alla clausola WHERE per aiutarci a cercare un modello specifico nella nostra colonna.
12. LIMIT: questa è un'altra clausola importante perché ci consentirà di specificare il numero massimo di righe che l'insieme dei nostri risultati mostrerà per noi.
13. MAX(): questa sarà una funzione che prenderà il nome di o colonna e lo trasformerà nell'argomento. Quindi restituirà anche il valore più grande su quella colonna.
14. MIN(): questa sarà una funzione simile alla precedente, ma restituirà il valore più piccolo in quella colonna.

15. OR: Questo è un operatore in SQL che filtrerà i risultati che otteniamo e quindi includeremo solo le righe in cui entrambe le condizioni sono vere.
16. ORDER BY: questa sarà una clausola che indicherà cosa si desidera ordinare il set di risultati da una particolare colonna. Puoi scegliere di farlo in ordine numerico o alfabetico in base alle informazioni con cui stiamo lavorando.
17. ROUND(): questa sarà una funzione che prenderà il nome di una colonna e lo avrà come numero intero e come argomento. È quindi in grado di arrotondare i valori nella colonna al numero di cifre decimali re-specificato dal nostro numero intero scelto.
18. SELECT: Questa è un'istruzione che useremo per recuperare alcuni dei dati che si trovano all'interno del database, purché siano presenti. Ciascuna delle domande che faremo tra un minuto è, per cominciare, questa affermazione.
19. SELECT DISTINCT: questo specificherà che l'istruzione sarà una query in grado di restituire valori che sono univoci nelle colonne specificate che scegliamo.
20. SUM: Questa è un'altra funzione con cui siamo in grado di lavorare che prenderà come argomento il nome delle nostre colonne e quindi

restituirà la somma di ogni valore che si trova in quella particolare colonna lungo il percorso.

21. UPDATE: La prossima opzione sulla nostra lista sarà l'istruzione UPDATE. Questo ci consentirà di apportare alcune modifiche alle righe che si trovano sulla nostra tabella, se lo desideriamo.
22. WHERE: questa è una clausola che possiamo usare per indicare che desideri filtrare il set di risultati, includendo solo le righe in cui le condizioni che hai impostato saranno vere.
23. WITH: questa è una clausola che ci aiuterà a memorizzare i risultati della nostra query in una tabella temporanea che utilizza un alias. Puoi anche definire più di una tabella utilizzando una virgola e un'istanza con questa parola chiave. La clausola WITH sarà spesso nota come CTE, o espressione di tabella comune e factoring di subquery.

## **Come creare una tabella**

La prossima cosa a cui daremo un'occhiata è come prendere effettivamente alcuni dei comandi che abbiamo fatto sopra e metterli a frutto. E la prima opzione è dare un'occhiata all'istruzione CREATE TABLE. Ad un certo punto, avrai un database e vorrai essere in grado di creare almeno una tabella, anche se a volte molte tabelle, all'interno di quel database. Queste tabelle sono in grado di conservare alcune delle informazioni importanti di cui si desidera occuparsi quando si tratta di lavorare in SQL, ed è per questo che è importante lavorarci.

Per creare una delle tue tabelle, utilizzeremo semplicemente l'istruzione CREATE TABLE per realizzarla. La sintassi di base che useremo per questo sarà di seguito:

```
CREATE TABLE TABLE_NAME (  
    COLUMN1 DATATYPE,  
    COLUMN2 DATATYPE,  
    COLUMN3 DATATYPE,  
    ....  
);
```

I parametri della colonna che siamo stati in grado di passare e impostare qui sono importanti perché sono dove troveremo tutti i nomi che vogliamo dare alle colonne di quella tabella. Quindi siamo in grado di vedere il parametro del tipo di dati e vedere come questo specificherà il tipo di dati su cui la colonna è in grado di trattenere. Ciò potrebbe includere la data, il numero intero, il varchar o qualcosa di simile.

Tieni presente che in realtà ci sono diversi tipi di dati che siamo in grado di utilizzare. Vedremo molti degli stessi tipi di dati mostrati negli altri linguaggi di codifica. Dobbiamo scegliere anche qual è il più importante per le nostre esigenze.

Da lì, siamo in grado di andare un po' oltre e creare effettivamente una delle nostre tabelle. Lavoreremo sulla creazione di una tabella che etichetteremo come "Persone" e poi aggiungeremo anche alcune delle informazioni di identificazione su queste persone. il codice che saremmo in grado di utilizzare per realizzare questo include:

```
CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);
```

La colonna su cui stiamo lavorando per il PersonID sarà di tipo int e quindi manterrà il numero intero. Ad esempio, d'altra parte, non vogliamo numeri in alcune delle altre colonne che abbiamo creato sopra. Quindi questo significa che le altre colonne sono in grado di trattenere i caratteri o sono di tipo varchar, quindi impostiamo la lunghezza massima per queste tutte a 255 caratteri, che dovrebbe essere sufficiente per contenere alcune delle informazioni che vogliamo sui nomi e indirizzi.



Con le informazioni che abbiamo fatto sopra, abbiamo praticamente appena creato una tabella di una riga vuota, ma dobbiamo iniziare da qualche parte, e questo ci aiuterà a vedere alcune delle codifiche che dobbiamo fare con questo linguaggio .

C'è un'ultima cosa che daremo un'occhiata a fare mentre siamo in tema di creazione di tabelle. Qui è dove creeremo una tabella con l'aiuto di un'altra tabella già esistente. Ciò significa che una copia di una delle nostre altre tabelle può essere utilizzata per creare una nuova tabella, utilizzando lo stesso comando di prima.

La nuova tabella che stiamo cercando di creare finirà con le stesse definizioni di colonne. E possiamo scegliere di lavorare con tutte le colonne in questo modo o anche con alcune colonne specifiche. se vuoi creare questa nuova tabella con l'aiuto della tabella che abbiamo già, allora troveremo che la nuova tabella verrà riempita con tutti gli stessi valori esistenti che troveremo da quella tabella originale come bene. la sintassi che siamo in grado di trovare con questo includerà:

```
CREATE TABLE NEW_TABLE_NAME AS
SELECT COLUMN1, COLUMN2,...
FROM EXISTING_TABLE_NAME
WHERE ....;
```

Possiamo andare un po' oltre e guardare un esempio di come tutto questo può essere impostato per le nostre esigenze. La seguente opzione ci aiuterà a creare una nuova tabella, che chiameremo "TestTables" e sarà una copia della tabella nota come Clienti. Il codice che possiamo usare per far sì che ciò accada includerà:

```
CREATE TABLE TestTable AS  
SELECT customer name, contact name  
FROM customers;
```

## **Modifica delle tabelle**

Ad un certo punto, ci sarà qualcosa nel codice che stai facendo con SQL che deve essere cambiato. Hai già esaminato e creato alcune delle tabelle con cui vorresti lavorare, ma è arrivato un nuovo cliente, o le loro informazioni sono cambiate, oppure è successo qualcos'altro e devi apportare alcune modifiche al lavoro che hai stavano facendo altrettanto. È qui che si impara a modificare le tabelle.

Per aiutarci ad apportare alcune delle modifiche che vorremmo nel nostro database, dobbiamo semplicemente usare l'istruzione di ALTER TABLE. Questa dichiarazione andrà bene per aiutarci a modificare, eliminare o aggiungere colonne che si trovano in una delle tabelle che abbiamo già. Questa istruzione verrà utilizzata anche per aiutarci ad aggiungere e eliminare alcuni dei diversi vincoli che si troveranno in una delle tabelle esistenti.

La prima cosa che possiamo fare è assicurarci di aggiungere una nuova colonna a una delle tabelle che abbiamo. Vorremo utilizzare due comandi qui. Inizieremo con ALTER TABLE e poi diremo al compilatore che vogliamo AGGIUNGERE ad esso. La sintassi con cui siamo in grado di lavorare qui sarà di seguito:

```
ALTER TABLE TABLE_NAME  
ADD COLUMN_NAME DATATYPE;
```

Ora, possiamo restare con questo stesso tipo di idea quando lavoriamo con altre cose nel nostro codice. Ma invece di lavorare con l'istruzione ADD, possiamo inserire la colonna DROP COLUMN, la colonna ALTER o MODIFY e altro ancora. L'ALTER TABLE ha lo scopo di aiutarci a completare e assicurarci che questo codice funzioni nel modo che vorremmo e può apportare qualsiasi modifica che vogliamo a una tabella che abbiamo creato in passato.

## **Fare una query**

L'ultima cosa a cui dobbiamo dare un'occhiata quando è il momento di lavorare con SQL è che vogliamo essere in grado di eseguire una query. Se disponiamo di un database piuttosto ampio, è importante poter esaminare e trovare le informazioni che vorremmo. Non vuoi passare attraverso questo e cercare di capire manualmente cosa sta succedendo o come puoi usarlo. Vuoi essere in grado di fare una ricerca veloce e le query in SQL, o l'istruzione SELECT, si assicureranno che ciò possa accadere.

Con questo in mente, dobbiamo prendere un momento per esaminare alcune delle codifiche che siamo in grado di fare con queste query e come possiamo farle funzionare per le nostre esigenze. All'interno dell'editor SQL che è in linea, è possibile esaminare e lavorare anche con questo tipo di codice.

Per renderlo semplice, dobbiamo prima dare un'occhiata a cos'è una query. Una query di un database sarà lì per aiutarci a estrarre le informazioni che si trovano in qualsiasi database che utilizziamo, quindi possiamo formattarlo in una forma che è molto più facile da leggere e capire veramente. Dobbiamo essere in grado di scrivere questo nella sintassi giusta in modo che il database sappia cosa vorremmo. Ogni database su cui ci concentriamo sarà leggermente diverso, ma molte volte funzionerà con SQL e si baserà su questo tipo di query, quindi è una buona idea per noi sapere come farli funzionare.

Queste query SQL, con l'aiuto di un linguaggio di manipolazione dei dati, arriveranno con quattro blocchi diversi. I primi due non saranno opzionali, quindi dobbiamo assicurarci che siano presenti. Questi diranno al compilatore cosa speriamo di far emergere e dove verrà trovato. E spesso, questo includerà il comando SELECT per aiutarci a fare tutto. Nella forma più semplice, troveremo che la query SQL sarà simile alla seguente:

```
SELECT X from Y;
```

In questo caso, la parola chiave SELECT ci aiuterà a identificare quali informazioni vorremmo visualizzare fuori dal database, quindi la parola chiave FROM ci aiuterà a identificare da dove provengono i dati e come queste fonti di dati stanno per associarsi tra loro.

C'è così tanto che possiamo fare quando si tratta di lavorare con questo particolare linguaggio, specialmente quando stiamo cercando di gestire alcuni dei database che sono così importanti per il lavoro che molte aziende cercano di gestire. Quando sei pronto per iniziare con questa lingua, dai un'occhiata a questa guida e scopri quanto può essere facile.

## **Conclusione**

Grazie per aver completato la programmazione per principianti: C, Python, SQL, speriamo che sia stato informativo e in grado di fornirti tutti gli strumenti necessari per raggiungere i tuoi obiettivi qualunque essi siano.

Il prossimo passo è iniziare con alcune delle programmazioni con cui siamo in grado di lavorare. Ci sono molti ottimi linguaggi di codifica disponibili con cui lavorare. Ma alcune delle migliori opzioni con cui lavorare, quelle che ci aiuteranno davvero a gestire la maggior parte del codice che vogliamo realizzare, inclusi C, Python e SQL. Questa guida impiegherà del tempo per esaminare questi linguaggi, insieme alla programmazione per computer in generale, per aiutarci a imparare davvero di più sulla programmazione e su come iniziare come principianti assoluti.

Le prossime sezioni che abbiamo esaminato hanno impiegato del tempo per esplorare cosa sta succedendo con alcune delle lingue più comuni là fuori. Abbiamo esplorato alcuni dei linguaggi C, che includono il linguaggio C, C # e C ++, come sono simili e come ognuno è diverso l'uno dall'altro. Siamo quindi passati a esaminare il linguaggio Python e come questo sia spesso il linguaggio preferito da molti nuovi programmatori che non hanno mai avuto la possibilità di lavorare con il codice in passato. Infine, abbiamo esaminato parte del lavoro che siamo in grado di fare con il linguaggio SQL e come questo può entrare in gioco quando vogliamo gestire i nostri database e renderli il più forti possibile.

Ci sono così tante cose interessanti che puoi fare quando è il momento di concentrarti sulla programmazione del computer, e con alcuni dei suggerimenti e dei trucchi di cui abbiamo discusso in questa guida, sarai in grado di imparare questi linguaggi di programmazione in pochi giorni, ea volte anche in un solo giorno, invece di perdere tempo e spendere anni sulle basi. Quando sei pronto per saperne di più sulla programmazione per computer e su alcuni dei linguaggi di codifica più comuni, assicurati di dare un'occhiata a questa guida per aiutarti a iniziare.

Infine, se hai trovato questo libro utile in qualche modo, una recensione su Amazon è sempre gradita!

**Programmazione computer:**

**PYTHON**

*Una guida passo passo per apprendere i concetti di  
base del linguaggio di programmazione Python con  
esercizi pratici*

**Tony Chan**

In questo libro imparerai i concetti di base della programmazione Python. È una guida passo passo, che assicura la comprensione della programmazione Python. Prima di tutto, è necessario conoscere la struttura della frase e la sintassi di Python, insieme alle espressioni di base. Quindi, guida le principali "Funzioni" come la manipolazione matematica e la Standard Library. Inoltre, ci sono una guida completa sulla selezione di operatori e dati logici, concetti di array, puntatori, classi e stringhe in Python.



Questa guida è progettata per i principianti che desiderano apprendere i concetti di base di Python in 7 o pochi giorni. Troverai questo linguaggio facile se sviluppi i tuoi programmi durante la lettura di questo libro.

## **Capitolo 1: Introduzione a Python**

Se parliamo delle radici di Python, dobbiamo parlare del linguaggio di programmazione ABC utilizzato nei primi anni '80. Principalmente, è stata l'influenza del linguaggio di programmazione ABC che ha portato allo sviluppo e all'ascesa di Python, che oggi è uno dei linguaggi di programmazione più influenti.

Il signor Van Rossum, che era un informatico e matematico, serviva come implementatore del linguaggio di programmazione chiamato ABC al Centrum Voor Wiskunde en Informatica all'inizio degli anni '80. Alla fine degli anni '80, mentre lavorava su un nuovo sistema operativo distribuito chiamato AMOEBA, Van Rossum iniziò a cercare un linguaggio di scripting con una sintassi come ABC ma poteva accedere alle chiamate di sistema da Amoeba. Così, lo stesso Van Rossum iniziò a progettare un nuovo semplice linguaggio di scripting in grado di superare i difetti del linguaggio di programmazione ABC.

Alla fine, nei primi anni '90, Van Rossum alla fine ha rilasciato la prima versione del suo linguaggio di programmazione. Inizialmente, il suo linguaggio di programmazione portava il sistema del modulo Modula-3. Oggi chiamiamo questo linguaggio di programmazione "Python".

Le persone di solito presumono che Python abbia preso il nome da un serpente o pitone, anche le insegne di Python raffigurano due serpenti, il primo blu e il secondo giallo, nel loro logo. Ma pochissime persone tra loro conoscono la vera storia dietro il nome di Python. Negli anni '70, c'era una commedia televisiva molto popolare sulla BBC chiamata Monty Python's Fly Circus. Van Rossum era un grande fan di quello spettacolo. Così, Rossum ha chiamato il suo progetto "Python" dopo il Fly Circus di Monty Python.

## **1.1 Passaggi per imparare Python**

Python è uno dei linguaggi di programmazione più significativi che ogni sviluppatore dovrebbe conoscere. Questo linguaggio viene utilizzato dalla maggior parte dei programmatori per creare siti Web, creare algoritmi di apprendimento ed eseguire altre attività di programmazione. In primo luogo, può essere travolgente, frustrante e un compito impegnativo cercare di imparare Python, soprattutto se non sei sicuro di come affrontare gli strumenti e le risorse.

Una delle cose che potresti trovare più frustrante è quanto siano familiari tutte le risorse di apprendimento. Se sei interessato a comprendere la lettura e l'apprendimento di Python, vedrai che ogni risorsa di apprendimento vuole che tu dedichi molto tempo alla sintassi di Python prima ancora che tu possa pensare a fare ciò che ti interessa.

Questa incoerenza ha reso in qualche modo difficile per molti programmatori imparare Python. I programmatori esperti spesso ti lanciano allusioni come questa: “Python è per bambini!” "Oh, amico! È troppo facile imparare Python!" A volte, ti verrà detto anche questo. Ma, in realtà, anche un paio di righe di codice apparentemente semplici possono essere incredibilmente confuse. Ad esempio, perché anche una riga di codice ti confonde? Cosa significa "django.http" in Python? Perché ci sono parentesi in alcune righe di codice? Può essere molto difficile capire come tutto funzioni insieme quando non sai molto di Python.

Il problema è che per creare qualcosa di interessante, e devi capire i mattoni di Python. Il suddetto frammento di codice "django.http" utilizza la popolare architettura MVC per generare una vista di un sito Web, che è uno dei principali elementi costitutivi di un sito Web. Se non sei a conoscenza di scrivere il codice per costruire una vista, la creazione di un sito web dinamico non è possibile per te, dopotutto.

La maggior parte dei tutorial e dei corsi sulla programmazione Python presuppongono che dovrai imparare tutto sulla sintassi di Python prima di poter iniziare a fare qualsiasi cosa. Questo è ciò che si traduce nel trascorrere i tuoi mesi di tempo solo sulla sintassi, quando ciò che vuoi veramente fare è analizzare i dati, costruire un sito web o creare un drone autonomo.

In genere, dedicare tutto il tempo all'apprendimento della sintassi porta a svanire la motivazione. Qui, vorremmo pensarlo come un "muro noioso". Molti tutorial e corsi dicono che devi essere abbastanza disposto a scalare il "muro della noiosità" per arrivare alla "terra delle cose interessanti su cui lavorerai" ma, credici, non è l'unico modo per imparare Python!

Dopo aver affrontato quel "muro di noiosità" un paio di volte e dopo essersi allontanati, si può trovare un processo che funziona meglio per lui o per lei. In particolare, potrebbe trovare un modo per combinare l'apprendimento delle basi con la creazione di idee e programmi interessanti. In un certo senso aggira il "muro della noia" e si dirige verso l'alto. Potrebbe dedicare il minor tempo possibile all'apprendimento delle basi e quindi saltare immediatamente a fare cose che lo interessano.

A questo punto, ti mostreremo come replicare questo processo passo dopo passo, indipendentemente dal motivo per cui vuoi imparare Python. Tutto questo inizia con la ricerca della tua motivazione!

---

- **Motivazione per imparare Python**

Vale la pena chiedersi perché vuoi imparare Python prima di iniziare a immergerti, dato che sarà un viaggio lungo e talvolta doloroso per essere un Python Master. Non ce la farai senza una motivazione sufficiente!

La motivazione conta molto!

La maggior parte delle persone dormono durante le lezioni di programmazione al liceo e all'università, dove devono memorizzare la sintassi e non sono motivate. D'altra parte, quando hanno bisogno di usare Python per creare un sito web per generare saggi automaticamente, rimangono a lavorare molte notti studiando Python per finirlo.

Quindi, se la mettiamo in un altro modo, possiamo dire che imparare qualsiasi cosa è molto più facile quando abbiamo un motivo per imparare.

Se sei in grado di capire cosa ti motiva, ti aiuterà a scoprire l'obiettivo finale, così come un percorso che ti porterà lì senza essere frustrato. Non devi tracciare un progetto specifico. Puoi scegliere un'area generale che ti interessa, mentre prevedi di imparare Python.

Python offre molti campi che potrebbero interessarti. Puoi scegliere un campo che ti interessa, come ad esempio:

- Scienza dei dati
- Apprendimento automatico
- App mobili
- Siti web
- Giochi
- Hardware / sensori / robot
- Script di automazione

Puoi scegliere una o più aree che ti interessano e che desideri seguire. Concentrerai il tuo apprendimento nel tuo campo di interesse e alla fine sarai in grado di costruire progetti in quelle aree.

---

- **Imparare la sintassi di base**

Purtroppo, non puoi saltare questo passaggio specifico mentre impari Python. Devi imparare le basi della sintassi di Python prima di immergerti più a fondo nella tua area di interesse. Sappiamo che, su questo, vuoi dedicare il minor tempo possibile, poiché non è stimolante o motivante. In questo libro, ti aiuteremo ad apprendere alcune basi della sintassi di Python.

---

- **Fare progetti**

Dopo aver appreso la sintassi di base, puoi iniziare a creare i tuoi progetti. Sviluppare i tuoi progetti è un ottimo modo per imparare perché ti consente di trasferire le tue conoscenze in qualcosa di produttivo. Dovresti ricordare che sarà difficile acquisire più conoscenza a meno che non applichi la tua. I progetti miglioreranno le tue capacità, ti aiuteranno a capire e imparare cose nuove e ti aiuteranno a costruire un portfolio che i potenziali datori di lavoro possano presentare.

Ma, in questa fase, i progetti molto liberi saranno difficili. Ti bloccherai molto e dovrai tornare alle scartoffie e alla lettura. Per questo motivo, sviluppare progetti più formali di solito è meglio prima di sentirti abbastanza a tuo agio da costruire progetti interamente sotto il tuo. La maggior parte dei siti web o dei libri tutorial forniscono progetti organizzati in modo che tu possa imparare da essi. Dopo aver modificato un po' questi progetti, ti consente di creare cose interessanti nell'area a cui tieni impedendoti di perderti. Potresti essere coinvolto in Python perché ti piacerebbe creare un gioco, lavorare nella tecnologia o fare qualcos'altro. Quindi, guarderemo al futuro di Python alla fine di questo libro.

---

- **Lavora sui progetti da solo**

È il momento di lavorare sui tuoi progetti dopo aver completato alcuni progetti organizzati offerti da diversi siti Web di tutorial, come menzionato sopra. È difficile capire quanto hai veramente imparato nel tuo viaggio per imparare e capire Python prima di uscire e provare a sviluppare un progetto da solo. Nessuno è perfetto, quindi sappiamo che continuerai a consultare risorse e apprendere nuovi concetti di programmazione Python mentre il tuo lavoro continua ma, in questo momento, lavorerai su qualsiasi cosa su cui vuoi lavorare.

Come accennato nel punto precedente, dovresti essere a tuo agio con il debug di errori e problemi all'interno del tuo programma prima di immergerti nello sviluppo dei tuoi progetti Python. Tuttavia, ecco alcune risorse utili che potresti consultare in caso di problemi con il tuo progetto.

- **Stack Overflow**

Stack Overflow è un forum in cui puoi porre domande sul tuo problema e ottenere una risposta utile da qualcun altro. La maggior parte delle persone discute di problemi relativi alla programmazione su Stack Overflow.

Andando avanti, qui è possibile rispondere alle domande specifiche di Python poiché ci sono migliaia di sviluppatori Python che lo visitano ogni giorno. Se necessario, puoi sempre porre la tua domanda, ma di solito una ricerca su Stack Overflow mostrerà che qualcuno ha già chiesto del problema che stai affrontando e ha ottenuto una bella risposta.

- **Google**

In secondo luogo, nell'elenco c'è Google. Dobbiamo credere che Google sia lo strumento e il motore di ricerca più utilizzati dai programmatori. Potresti trovare Google molto utile quando tenti di eseguire il debug o correggere un errore di programmazione.

- **Documentazione ufficiale di Python**

La documentazione di Python è un ottimo modo per cercare il materiale specifico di Python. Se hai una presa potente sul debug e sulla risoluzione degli errori, è tempo di immergerti nei tuoi progetti e, se sei bloccato in qualcosa, è tempo di consultare la documentazione ufficiale di Python. Puoi consultare le cose che ti interessano, lì. Ad esempio, il concetto di trading azionario automatizzato potrebbe interessarti. Questo è qualcosa, quello che ti ha motivato! Quindi, molto presto, dopo aver compreso le basi della programmazione Python, inizierai a lavorare su alcune risorse che potrebbero aiutarti a sviluppare un progetto che potrebbe aiutarti a fare automaticamente scambi nel mercato azionario.

---

- **Continuare a lavorare**



Una volta completate tutte le fasi sopra menzionate, dovresti continuare ad aumentare la complessità e la varietà dei tuoi progetti nella programmazione Python. Se sei completamente a tuo agio con ciò che stai progettando, significa che è tempo per te di provare qualcosa di più complesso e molto più complicato. Potresti considerare di iniziare un progetto nuovo e più difficile o aggiungere un po' di complessità al tuo progetto recente, o forse potresti portare il tuo progetto a un tipo di sfida completamente diverso. Per questo! Devi ricordare che la pratica rende un uomo perfetto!

Di seguito sono riportati alcuni suggerimenti per aumentare la complessità dei tuoi progetti per assicurarti di fare davvero progressi nell'apprendimento di Python.

- Puoi provare a insegnare a un noob come sei riuscito a costruire il tuo progetto. Niente di meglio può aiutarti ad imparare una materia come insegnarla!
- Dovresti pensare ai processi che possono aumentare la capacità del tuo progetto.
- Potresti considerare di lavorare sull'efficienza del tuo progetto.
- Potresti pensare a modi per rendere il tuo progetto più user-friendly.
- Potresti considerare dei modi per pubblicizzare il tuo progetto.

## **1.2 Fondamenti di Python**

Devi imparare le basi della sintassi di Python prima di iniziare a scrivere il tuo primo programma Python, che è tipicamente noto come "Hello World". A questo punto, parleremo delle basi della programmazione Python e della sua sintassi, che sicuramente ti aiuteranno a iniziare a costruire la tua carriera come sviluppatore Python. Ora componiamo prima il programma Python di base.

---

- **Scrivere il tuo primo programma Python**

Puoi stampare e compilare i programmi Python, di solito, in due modi

- **Modalità interattiva**

In modalità interattiva, un programmatore può scrivere e compilare un programma Python.

- **Modalità script**

In modalità script, puoi eseguire il tuo programma Python, che di solito è un "file .py", già salvato sul tuo dispositivo.

Vediamoli entrambi in pratica.

---

- **Ciao mondo! In modalità interattiva**

Per scrivere il tuo primo programma, devi inserire il comando indicato di seguito nel tuo Terminale nella modalità interattiva della programmazione Python.

**\$ python**

Se hai inserito questo comando nel tuo terminale, ora sei nella modalità interattiva di Python.

D'altra parte, se stai usando un IDE, non devi digitare il comando sopra menzionato per entrare nella modalità interattiva di Python. Di seguito è riportata la sintassi della programmazione Python di base per compilare il tuo primo programma. In genere, il primo programma è noto come "Ciao mondo!", Ma cambieremo le probabilità. Scriveremo il nostro primo programma come "Ciao al mondo della programmazione Python!".

Il tuo IDE dovrebbe visualizzare "Ciao al mondo della programmazione Python!" quando scrivi il tuo programma e premi invio. Ricorda che tutto ciò che scrivi tra parentesi "" verrà stampato sul tuo IDE. Nel nostro caso, è: Ciao al mondo della programmazione Python.

```
print("Ciao al mondo della programmazione Python")
```

## **Produzione:**

Dopo aver compilato questo comando, vedrai un messaggio "Ciao al mondo della programmazione Python!" sullo schermo del computer.

---

- **Ciao mondo! In modalità Scripting**

Supponiamo che tu abbia salvato il tuo programma scritto Ciao al mondo della programmazione Python come file Python, cioè "file .py". Ora, sul tuo dispositivo, dovresti cercare un file con estensione ".py". Dì solo che hai salvato il tuo progetto come "Hello to the Python.py", ecco come funzionerà il tuo codice nella modalità di scripting di Python.

All'inizio, il file di progetto che hai scritto e salvato deve essere funzionale ed eseguibile. In genere, i programmatori lo fanno preferibilmente utilizzando il comando:

**\$ chmod +x test.py**

Ora, se il tuo file funziona ed è eseguibile, puoi eseguire il programma in modalità di scripting di Python.

**\$ Python Ciao a Python.py**

Una volta eseguito questo comando, vedrai "Ciao al mondo della programmazione Python!" scritto sul terminale del computer o sulla console. Viene eseguito un "Ciao al mondo della programmazione Python", hai imparato la tua prima sintassi Python.

---

- **Parole chiave e identificatori in Python**

Nella programmazione Python, c'è un numero totale di trentuno parole chiave e cinque identificatori significativi.

Identificatori e parole chiave sono qualcosa a cui ti abituerai senza sforzo mentre ti concentri sulle tue capacità di programmazione Python.

---

### ● **Identificatori Python**

Un identificatore Python è tipicamente una funzione, un modulo, una variabile, una classe o qualcos'altro. Nella programmazione Python, si assegna un nome a un oggetto e questo è noto come identificatore. In genere, un identificatore valido inizia con una lettera maiuscola (AZ), una lettera minuscola (az) o un trattino basso (\_). Di solito, questi sono seguiti da trattini bassi, zero, lettere o numeri (compresi tra 0 e 9).

Nella programmazione Python, ci sono cinque tipi fondamentali di identificatori:

- Funzioni
- Moduli
- Classe
- Variabili
- Altri oggetti

Andando avanti, dovremmo dare un'occhiata alle parole chiave disponibili nella sintassi di Python.

---

### ● **Nozioni di base sulle parole chiave della sintassi di Python**

Per vedere tutte le trentuno parole chiave in Python, devi aprire il tuo IDE e digitare il seguente comando:

```
>>> import keyword
>>> keyword.kwlist
```

### **Produzione:**

Quando inserirai questi comandi, il tuo IDE stamperà questo:

```
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue',
'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if',
'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise',
'return', 'try', 'while', 'with', 'yield']
```

Queste parole chiave menzionate possono cambiare con ogni nuova versione di Python. Ora, ci sono alcune altre cose che dovresti tenere a mente mentre lavori con Python come linguaggio di programmazione

- Le parole chiave sono fisse e non possiamo usarle come identificatori.
- Le parole chiave in Python fanno distinzione tra maiuscole e minuscole. In genere, chiamiamo queste parole chiave come parole riservate.

---

### ● **Dichiarazioni Python**

Le prossime cose che dovresti tenere a mente sono Linee e Rientro prima di procedere ulteriormente con i concetti di base della sintassi Python.

In Python, l'indentazione è il supporto principale nell'organizzazione di blocchi o moduli di codice. Inoltre, è rigidamente applicato e rende il tuo codice Python leggibile, scansionabile e revisionabile.

Dobbiamo ricordare che in ogni situazione, lo spazio che va con il rientro varia. Per capirlo, dobbiamo osservare il seguente codice:

```
>>> person=["Cindy", "Nida", "Malia"]
>>> for n in person:
>>>     if n== "Nida": print(n)
>>>     else: print("Not Nida")
```

Nel codice seguente, è necessario notare che come "l'istruzione for" ha uno spazio, ovvero il rientro e "l'istruzione if" ha due spazi.

In Python, quando finisci una riga usando un punto e virgola, termini una frase.

;

Andando avanti, in Python, puoi anche dichiarare più istruzioni in modo efficace in una singola riga. Per esempio:

```
person="Faheel Nasir";age=25;location="Lahore"
```

---

- **Argomenti della riga di comando**

Fondamentalmente abbiamo utilizzato un argomento della riga di comando in "Hello to the Python.py" e nello script interattivo. Quindi, è il modo più semplice per introdurti agli argomenti della riga di comando. Questi sono una parte essenziale dei fondamenti della sintassi di programmazione Python, che dovresti conoscere.

---

- **Come creare un ambiente virtuale**

Gli ambienti virtuali saranno una di quelle cose che creerai per ogni nuovo progetto. In Python, puoi anche lavorare su nuovi progetti senza dover creare nuovi ambienti virtuali. Tuttavia, ciò significherebbe che ti ritroverai in una situazione in cui

- Non è possibile migliorare / ottimizzare le dipendenze. Ad esempio, potresti non aggiornare la versione predefinita di Python (dalla 2.7 alla 3.8).

Per sviluppare un nuovo ambiente virtuale, dovresti andare sul tuo Terminale e digitare il seguente comando

**\$virtualenv newenvironment**

In questo comando, il nome dell'ambiente virtuale che abbiamo appena sviluppato è "newenvironment". Ora digita il seguente comando per avviare il tuo nuovo ambiente

**source newenvironment**

---

- **Prendere l'input dell'utente**

È possibile trovare due tipi comuni di sintassi Python che potrebbero richiedere l'input dell'utente

- **raw\_input**
- **input ()**

Ognuno di loro richiede a un utente di inserire i propri input.

- **raw\_input ()**

In Python 3, questo non è più vero o valido. Invece, input () è il nuovo modulo di input.

- **Input()**

Per comprendere questo concetto, dobbiamo guardare al seguente esempio:

```
user_input_request = input("Inserisci il nome del tuo amico  
immaginario:")
```



Dopo aver inserito questo, il tuo IDE ti permetterà di inserire il nome del tuo amico immaginario.

---

- **Capitolo 2: Gli attributi irresistibili di Python**

Ormai abbiamo quasi tutti i linguaggi di programmazione che possiamo contare sulle dita. Ci sono così tanti linguaggi di programmazione e ognuno di essi ha le sue caratteristiche uniche, distinguendoli dagli altri. In definitiva, a causa delle caratteristiche di un linguaggio, questo potrà essere approvato o scelto per i progetti più grandi. Quindi, prima di iniziare i concetti di Python, in dettaglio, sarà utile dare un'occhiata alle caratteristiche, unicità e applicazioni di Python, ecc. Nel capitolo seguente, sarai in grado di apprendere i concetti di base, le caratteristiche, le applicazioni e il design di Python, che lo rende unico e potente rispetto a molti altri linguaggi.

---

- **2.1 Caratteristiche uniche della programmazione Python**

Per iniziare, inizieremo con le caratteristiche uniche di Python.

---

- **Python è facile**

Quando usiamo la parola "facile" per la programmazione Python, dobbiamo prenderla in un contesto molto diverso.

- **Facile da codificare**

Ne abbiamo discusso nel nostro primo capitolo, dopo aver sviluppato il nostro primo programma, possiamo dire che è davvero facile programmare in Python. Nelle sezioni successive, confronteremo Python con altri linguaggi di programmazione come C, C ++, Java e HTML. Vedremo che rispetto agli altri linguaggi di programmazione, Python è più comodo da programmare. Chiunque abbia un background di programmazione può imparare la sintassi di Python in pochissime ore. Tuttavia, per diventare un maestro della programmazione Python con tutti i suoi concetti, moduli e pacchetti richiederà tempo. Quindi, in altre parole, possiamo dire che Python è un linguaggio adatto agli sviluppatori.

- **Facile da leggere e capire**

Abbiamo letto in precedenza che Python è un linguaggio di alto livello. Per lo più, potresti trovare il codice di Python uguale all'inglese, con pochissime parole chiave e identificatori. Guardando il codice, qualsiasi programmatore o anche un non programmatore può dire che cosa farà questo codice.

Inoltre, Python viene digitato dinamicamente e questo richiede l'indentazione. Questa caratteristica rende il codice di Python più affidabile e facile da capire.

---

- **Linguaggio espressivo**

Inizialmente, concentriamoci su una domanda; cos'è l'espressività in un linguaggio di programmazione per computer!

Supponiamo che tu abbia due linguaggi di programmazione per computer, considera il primo "N" e il secondo "M" e utilizzando le trasformazioni locali puoi usare e convertire tutti i programmi scritti da M a N., Potrebbero esserci, tuttavia , molti programmi che possono essere scritti usando il linguaggio M ma non sarai in grado di trasformarli in linguaggio N, usando le trasformazioni locali.

Adesso! Si può dire che il linguaggio M è più espressivo rispetto al linguaggio N. Quindi, per quanto riguarda Python, ci offre una moltitudine di costrutti che possono aiutarci a concentrarci più sulla soluzione di un problema che sulla sintassi. Questa espressività è una delle tante caratteristiche uniche di Python che potrebbe dirti il motivo e ti dà la motivazione per imparare la programmazione Python.

---

- **Open-source e linguaggio libero**

Prima di tutto, devi sapere che Python è gratuito ed è disponibile su molte piattaforme. La fonte più affidabile per scaricare e installare Python è il proprio sito Web ufficiale, ad es

**[https: /www.python.org/](https://www.python.org/)**

La seconda cosa in questa preoccupazione è che Python è un linguaggio di programmazione Open-Source. Open-Source significa che il pubblico può accedere al codice sorgente di Python in qualsiasi momento. Python può essere scaricato, modificato, utilizzato e distribuito anche da un laico. Questo tipo di software o linguaggio è comunemente chiamato FLOSS "Software libero e open source".

Nel mondo di Python, tutti gli sviluppatori Python si stanno muovendo verso un unico obiettivo, ovvero il miglioramento di Python e delle sue caratteristiche. Per questo, Python ha reso il suo codice sorgente un Open-Source.

---

- **Linguaggio di alto livello**

Fin dall'inizio del nostro libro, abbiamo affrontato questo punto che Python come linguaggio è un linguaggio di programmazione per computer di alto livello.

In qualità di sviluppatore, non devi preoccuparti o ricordare l'architettura della tua macchina o sistema. Il tuo codice può essere utilizzato globalmente, per qualsiasi computer, per qualsiasi macchina o per qualsiasi dispositivo, scritto in Python. Non ti preoccuperai nemmeno della memoria del sistema. Questa caratteristica rende Python più facile scrivere e scrivere codice. Quindi, come accennato in precedenza, potremmo chiamare Python un "linguaggio adatto agli sviluppatori".

---

- **Linguaggio di programmazione portatile**

Supponiamo che tu abbia sviluppato un programma o un'app Python per la tua macchina orientata al sistema operativo Windows. E, diciamo che vuoi solo eseguire quel programma su un dispositivo orientato al sistema operativo MAC. Ora, non avrai bisogno di fare troppe improvvisazioni al tuo codice Python.

In poche parole, prenderai il tuo codice Python dal tuo dispositivo orientato al sistema operativo Windows e lo eseguirai su qualsiasi macchina con sistema operativo MAC. Quindi, se utilizzi Python, non è necessario codificare in modo diverso per strumenti o sistemi operativi diversi. Questa caratteristica di Python rende il nostro linguaggio in questione, un linguaggio di programmazione molto portatile. In questa preoccupazione, devi tenere a mente una cosa e devi evitare ogni caratteristica dipendente dal sistema.

---

- **Linguaggio interpretato**

Se sei già uno sviluppatore di C ++ o Java o qualsiasi altro linguaggio di programmazione, o hai familiarità con i linguaggi di programmazione, devi sapere che devi prima compilare il tuo codice. Quindi sarà in grado di eseguire o eseguire attività. Ma il tuo codice non richiede di essere compilato all'inizio, quando selezioni Python come linguaggio di programmazione. All'interno del sistema, Python trasforma il codice sorgente in un formato immediato, noto come bytecode. In altre parole, devi eseguire il tuo codice sorgente, scritto in Python, senza pensare a connetterti alle librerie interessate o a qualsiasi altra cosa importante.

Quando la macchina o il dispositivo sta leggendo il tuo codice Python, potresti dire che il codice sorgente viene eseguito da ogni riga a un'altra riga e non tutto il codice in una volta. Questa caratteristica di Python semplifica il debug del codice Python. Su un altro punto, è vero che l'interpretazione rende Python un po 'più lento di altri linguaggi, come Java, ma non importa se confrontato con i vantaggi di Python.

---

- **Linguaggio di programmazione orientato agli oggetti**

Stiamo anche discutendo questo punto dall'inizio che Python è in grado di rimodellare il mondo reale e, per questo motivo, è un cosiddetto “linguaggio di programmazione orientato agli oggetti” o “OOP”. Un linguaggio OOP si concentra sugli oggetti e riunisce i dati e le sue funzioni.

Contrariamente a ciò, ogni linguaggio orientato alle procedure ruota attorno alle funzioni; Le funzioni sono i codici che possono essere utilizzati ancora e ancora. Python è un linguaggio che supporta sia la programmazione orientata alla procedura che la programmazione paradigmatica orientata agli oggetti e questa caratteristica rende Python il linguaggio di programmazione più unico. Rispetto a Java, Python è un linguaggio che supporta l'ereditarietà multipla in un unico codice. Inoltre, una classe in Python è una mappa per un oggetto come questo. È un tipo di dati astratto, che non ha molta importanza.

---

- **Estensibile**

Diciamo che sei uno sviluppatore di qualsiasi altro linguaggio di programmazione e devi scrivere un codice Python; per questo, puoi anche usare diversi comandi di altri linguaggi di programmazione, come C ++, nel tuo codice Python.

Questa caratteristica di Python lo rende un linguaggio di programmazione molto estensibile, il che significa che può essere espanso ed esteso per includere altri linguaggi di programmazione in esso.

---

- **Linguaggio incorporato**

Nel punto precedente, abbiamo discusso che nel nostro codice sorgente Python è in grado di accettare codice e comandi anche da altri linguaggi. Tuttavia, puoi anche inserire il tuo codice Python in un codice sorgente di qualsiasi altro linguaggio di programmazione come C ++. Questo ci consente di integrare le capacità di scripting di altri linguaggi di programmazione nel nostro codice.

---

- **Ampia libreria standard**

Puoi scaricare Python con pacchetti e librerie più estesi in modo da non dover scrivere l'intero codice per ogni singola cosa del tuo programma. Esistono librerie più estese per espressioni utilizzate regolarmente, per identificare diverse generazioni di documenti, per unità di test, per lo sviluppo di siti Web e browser Web, per il threading, per lo sviluppo e la manutenzione di database, per Common Gateway Interface, per lo sviluppo e la manutenzione della posta elettronica, per molti tipi di manipolazione delle immagini e per molte altre funzionalità.

---

- **Programmazione dell'interfaccia utente grafica**

Qualsiasi programma o software non può essere considerato di facile utilizzo finché non si sviluppa la sua interfaccia grafica utente (GUI). Qualsiasi utente che utilizza il software sviluppato può comunicare facilmente con il software tramite l'interfaccia utente grafica del software. Python offre una varietà di librerie per migliorare l'interfaccia utente grafica del tuo programma, software o applicazione. Le più comuni tra queste funzionalità sono wxPython, JPython e Tkinter. Python offre questi toolkit e ti consentono di creare o sviluppare la GUI del tuo programma in modo molto efficiente e rapido.

---

- **Linguaggio digitato dinamicamente**

Python è un linguaggio di programmazione digitato dinamicamente. Ciò significa che la tua macchina o dispositivo deciderà il tipo per qualsiasi valore solo in fase di esecuzione, non prima. Pertanto, in Python, non è necessario definire il tipo di dati quando lo si definisce o lo si dichiara.

---

- **2.2 Versioni di Python**

- **Fase 1**

Nella prima fase di Python, Python ha sviluppato e annunciato la sua prima versione, ovvero Python 1.0, nel gennaio 1994. In questa versione, le caratteristiche principali introdotte erano gli strumenti di programmazione funzionale come [ridurre](#), [filtro](#), [lambda](#), e [carta geografica](#). L'ultima versione rilasciata mentre il signor Van Rossum era ancora alla CWI era Python 1.2. Questa versione è stata rilasciata nell'anno 1995.

Nel 1995, il signor Van Rossum stava lavorando al suo linguaggio creato, Python, presso la Corporation [per iniziative di ricerca nazionali](#). Da lì, ha lanciato molte altre versioni di Python. Con il rilascio della versione 1.4, Python ha acquisito molte funzionalità nuove. La più popolare tra queste funzionalità era Modula-3. Modula-3 doveva ispirare argomenti di parole chiave. In Python 1.4, hanno anche introdotto una forma base di [dati nascosti](#) mediante l'uso di [name mangling](#).

Van Rossum lanciò anche la [programmazione per computer](#) “all in one”, noto come CP4E. La sua missione era rendere la programmazione accessibile ai profani, con il know-how necessario di molti linguaggi di programmazione. A questo proposito, Python ha svolto un ruolo importante, soprattutto per la sua sintassi chiara e semplice.

La versione successiva fu la 1.6. In Python 1.6, venne introdotto una nuova licenza CNRI e un'altra versione con correzioni di vari bug.

---

- **Fase 2**



Nella seconda fase, Python ha rilasciato la sua versione 2.0. È stato rilasciato nell'ottobre 2000. In questa versione, Python ha introdotto [elenchi di comprensioni](#). Questa era una caratteristica che Python prese in prestito dagli altri linguaggi di programmazione. Successivamente, venne introdotto Python 2.1, che era un parente stretto di Python 1.6.1 ed era strettamente collegato a Python 2.0.0. Il rilascio di Python 2.1 ha introdotto una modifica fondamentale per supportare gli ambiti annidati in modo che possano competere con altri [con ambito statico](#) linguaggi di programmazione.

Andando avanti, la prossima versione di Python è stata Python 2.2. Python 2.2 è stato rilasciato nel 2001. In questa versione, Python è stato in grado di introdurre un cambiamento importante; unificazione dei tipi di Python, cioè classi differenti in un'unica gerarchia. Questo cambiamento è stato fondamentale, grazie al quale Python è diventato un linguaggio di programmazione paradigma orientato agli oggetti. La versione successiva di Python fu la 2.5. Python ha introdotto questa versione nel settembre 2006. La modifica necessaria in questa versione erano le dichiarazioni, che erano in grado di racchiudere qualsiasi blocco di codice con un gestore di contesto.

Le versioni successive di Python, rilasciate nella sua Fase 2, furono Python 2.6 e Python 2.7. Python 2.6 è stato creato per supportare Python 3.0 e c'erano alcuni bug, avvisi ed errori di sintassi che sono stati sottoposti a debug con il rilascio di Python 3.0. D'altra parte, Python 2.7 è stato rilasciato per supportare Python 3.1.

---

- **Fase 3**

Nella fase 3 dello sviluppo di Python, Python ha rilasciato Python 3.0. Python 3.0 è anche conosciuto come Py3K o Python 3000. Python 3000 è stato rilasciato nel 2008. Inoltre, Py3K è stato sviluppato per superare i difetti di Python.

Python 3.0 è considerato un file [linguaggio multi-paradigma](#). Tuttavia, gli sviluppatori possono ancora usarlo come [un linguaggio strutturato, funzionale, e orientato agli oggetti](#).

Quindi, dopo aver letto questo, possiamo dire che Python viene continuamente aggiornato con versioni, supporti e componenti più recenti. Potresti trovare utile questa tabella se desideri conoscere le versioni di Python e le loro date di rilascio.

| <b>Versione di Python</b> | <b>Data di rilascio</b> |
|---------------------------|-------------------------|
| Python 1.1.0              | Gennaio 1994            |
| Python 1.5.1              | Dicembre 1997           |
| Python 1.6.1              | Settembre 2000          |
| Python 2.0.               | Ottobre 2000            |
| Python 2.0.1              | Aprile 2001             |
| Python 2.2.3              | Dicembre 2003           |
| Python 2.3.7              | Luglio 2003             |
| Python 2.4.6              | Novembre 2004           |
| Python 2.5.6              | Settembre 2006          |
| Python 2.6.9              | Ottobre 2008            |
| Python 2.7.17             | Luglio 2010             |
| Python 3.0.1              | Dicembre 2008           |
| Python 3.1.5              | Giugno 2009             |
| Python 3.2.6              | Febbraio 2011           |
| Python 3.3.7              | Settembre 2012          |
| Python 3.4.10             | Marzo 2014              |
| Python 3.5.9              | Settembre 2015          |
|                           |                         |

|               |               |
|---------------|---------------|
| Python 3.6.10 | Dicembre 2016 |
| Python 3.7.6  | Giugno 2018   |
| Python 3.8.1  | Ottobre 2019  |
| Python 3.9.0  | Ottobre 2020  |

---

- **2.3 Applicazioni di Python**

Python è un linguaggio di programmazione per computer open source noto per la sua natura ampiamente utile che lo separa da altri linguaggi di programmazione. Python come linguaggio può essere utilizzato in ogni disciplina della programmazione di computer in molti modi.

---

- **Sviluppo di applicazioni Web**

Puoi usare Python per creare molti tipi di applicazioni web. Python fornisce molte librerie per sviluppare e gestire siti Web e applicazioni Web. Ad esempio, Email Handling, beautiful soup, demand, JSON, XML, HTML e molti altri. Inoltre, Python fornisce molti framework incorporati, ad esempio Pyramid, Flask, Django (ne abbiamo parlato nel capitolo 1) e molti altri per gestire e sviluppare software e applicazioni web.

---

- **Applicazioni dell'interfaccia utente grafica desktop**

Python fornisce molte librerie dell'interfaccia utente grafica per creare l'interfaccia utente nelle applicazioni basate su Python. Alcuni degli strumenti e delle applicazioni a questo proposito sono Kivy e wxWidgets. Puoi utilizzare questi strumenti in molte fasi di sviluppo. Ricorda che Kivy è comunemente usato per sviluppare applicazioni multi-touch.

---

- **Sviluppo software**

È possibile utilizzare Python per il progresso delle procedure di programmazione del computer. Python funziona come linguaggio di supporto e può essere utilizzato per il test del software, il controllo di fabbricazione e la scheda.

---

- **Uso scientifico e numerico**

Python è un linguaggio mainstream ed è comunemente utilizzato per capire i problemi logici e numerici. A questo proposito, potresti trovare utili molti bundle Python e librerie Python, come IPython, Pandas, SciPy, ecc.

---

- **Utilizzo delle applicazioni aziendali**

Molti sviluppatori utilizzano Python per la produzione e la manutenzione dei propri software o applicazioni aziendali. Molti sviluppatori usano Python per mantenere i loro siti di e-commerce.

---

- **Applicazione basata su console**

Python può essere utilizzato per sviluppare applicazioni basate sul supporto. Ad esempio, è possibile utilizzare IPython per creare diverse applicazioni basate sul supporto.

---

- **Programmazione di applicazioni basate su audio o video**

Python è un ottimo linguaggio per eseguire molti compiti basati su video o audio. Molti sviluppatori lo usano per sviluppare diversi tipi di applicazioni multimediali. Per questo, puoi usare una funzionalità di Python, nota come cplay.

---

- **Applicazioni di disegno assistito da computer basate su 3D**

Molti sviluppatori Python lo utilizzano per creare e sviluppare applicazioni di disegno assistito da computer basate su 3D. Per lo sviluppo di questo tipo di applicazioni, Fandango è un'applicazione basata su Python molto utile che consente di visualizzare tutti i punti salienti del CAD.

---

- **Applicazioni aziendali**

Molti sviluppatori Python lo usano per sviluppare molte applicazioni che si possono utilizzare all'interno di un'organizzazione o in un'azienda. A questo proposito, Tryton e Picalo sono le applicazioni più conosciute.

---

- **2.4 Installazione di Python**

Come abbiamo detto prima, Python e il suo codice sorgente sono disponibili su Internet e possono essere contattati da molte fonti. Il modo migliore per accedere e scaricare Python è dal proprio sito Web, ad es

**<https://www.python.org>**

---

- **Installazione su Windows**

Per installare Python su Windows, dovresti andare al collegamento:

**<https://www.python.org/downloads/>**

Questo ti aiuterà a scaricare l'ultima versione, rilasciata da Python. Se stai usando questo metodo, il collegamento, come accennato in precedenza, ti reindirizzerà a una nuova finestra dove potresti trovare un elenco di molte versioni di Python. Da lì, puoi scaricare l'ultima versione di Python, ovvero Python 3.9.0 Tutto quello che devi fare è scaricare e salvare la recente versione di Python. Dopo averlo scaricato e salvato, fare doppio clic sul file eseguibile. Dopodiché, si aprirà una nuova finestra. Basta fare clic su "Personalizza installazione" e lasciarlo installare sul tuo sistema.

Dopodiché, apparirà una nuova finestra, raffigurante tutte le funzionalità opzionali di Python. È necessario installare tutte le funzionalità opzionali e selezionarle o verificarle tutte. Successivamente, fare clic sul pulsante Avanti per continuare la procedura.

Apparirà una nuova finestra con le opzioni avanzate per l'installazione di Python. Devi solo controllare il pulsante "seleziona tutto" per una migliore installazione, quindi devi fare clic sul pulsante "Avanti".

Dopo aver fatto clic sul pulsante "Avanti", Python 3.9.0 inizierà a installarsi sul tuo computer.

Dopo l'installazione, devi eseguire Python sul CMD del tuo sistema, aka. Prompt dei comandi. Per verificare se Python è installato correttamente, scrivi il seguente comando nel tuo prompt dei comandi.

**python;**

Quando si preme Invio, il prompt dei comandi potrebbe mostrare un errore. Di solito, questo accade perché il percorso di Python non è stato impostato correttamente.

Ora, per impostare correttamente il percorso di Python, devi solo andare sul desktop, fare clic destro su "Il mio PC" e andare nelle sue "Proprietà". Dopo aver selezionato le proprietà, vai su "Avanzate" e poi procedi verso "Variabili d'ambiente". Vedrai che verrà aggiunto un nuovo percorso nella sezione della variabile utente. Nelle tue "Variabili d'ambiente", puoi impostare il percorso di Python come nome della variabile e nella directory di installazione di Python. Quando il percorso per Python è impostato, ora puoi eseguire Python sulla tua stazione locale.

Ora, reimposta il prompt dei comandi e digita nuovamente il seguente comando.

**python;**

Infine! L'interprete Python si aprirà dove puoi iniziare a scrivere ed eseguire i comandi e le istruzioni Python.

---

- **Installazione su MAC**

Python 2.7 è integrato in Mac OS X 10.8 dalla stessa Apple. Se vuoi installare l'ultima versione di Python, puoi farlo andando sul sito web:

<https://www.python.org>

Da questo sito Web è possibile ottenere la versione recente di Python e molti dei suoi componenti. Per questo, devi scaricare il pacchetto per Python dal sito Web interessato.

Dopo aver scaricato il pacchetto, verrà scaricata una cartella nella "Cartella dell'applicazione" con il nome "Python 3.9.0". Qui puoi trovare l'ambiente di sviluppo per Python, cioè IDLE.

Il tuo sistema scaricherà anche un framework, noto come "Python Framework". Questo framework di solito ha diverse librerie per la programmazione Python e molti altri file eseguibili.

---

- **Capitolo 3: Compilatori, IDE e EDITOR di testo**

Nella programmazione Python, i compilatori Python, gli IDE e gli editor di testo svolgono un ruolo fondamentale. Come altri linguaggi di programmazione, per compilare la sintassi di Python in modo molto semplificato, ci sono molte applicazioni, compilatori ed editor di testo che possiamo usare. In questo capitolo studieremo su compilatori, editor di testo e IDE.

---

- **3.1 Compilatori per la programmazione Python**

Se hai un background di programmazione, devi sapere che un compilatore è un'applicazione di traduzione di codici o un semplice software che converte il tuo codice scritto da linguaggio di programmazione per computer di alto livello a linguaggio macchina, in modo che il tuo computer possa capirlo. Per Python, sono disponibili molti compilatori, con i propri sistemi di conversione del linguaggio di programmazione per computer specifici. Molti programmatori utilizzano questi compilatori per eseguire il debug dei propri programmi o per rendere eseguibile il proprio codice o per sviluppare algoritmi per i propri programmi.

Per la programmazione Python, PyCharm è uno dei migliori IDE che la maggior parte dei programmatori Python preferisce.

Discuteremo alcuni dei migliori compilatori qui.

---

- **Il compilatore Nuitka**



Nuitka è un compilatore molto noto per la compilazione e l'esecuzione del codice in Python. Nautika prende il tuo codice scritto in Python come input e trasforma il tuo codice in C per compilarlo ed eseguirlo. Puoi accedere a Nuitka su quasi tutti i sistemi operativi e tutte le piattaforme. Nautika si aggiorna regolarmente e sta diventando molto user-friendly per Windows, MAC, UNIX e altri sistemi operativi.

Un'altra caratteristica di Nautika è che puoi codificarci sopra, senza aver installato Python sulla tua workstation.

Nautika è un compilatore basato su Python, che non supporta nessun altro linguaggio, in generale. Quindi, ha varie funzionalità e librerie disponibili per lo sviluppo di Python. Gli sviluppatori utilizzano Nautika anche per sviluppare programmi nel campo dell'intelligenza artificiale e delle scienze dell'informazione.

---

- **Il compilatore PyJS**

Il compilatore PyJS per Python è un compilatore utilizzato da molti programmatori professionisti e società di sviluppo software. È un diverso tipo di compilatore. La sua caratteristica più nota è che traduce il codice Python in JavaScript. La maggior parte degli sviluppatori di siti web preferisce questo compilatore, a causa della sua natura unica per i programmi web.

Se parliamo della caratteristica unica fornita da PyJS, PyJS fornisce il supporto di runtime al codice. Ecco perché gli sviluppatori web lo consigliano principalmente per applicazioni e programmi basati sul web. Fornisce supporto a coloro che sperano di scrivere il proprio codice in Python e vogliono eseguirlo direttamente nel proprio browser Internet. Il PyJS interpreta anche il codice Python in un codice JavaScript debug che può essere eseguito ed eseguito nel browser Internet desiderato.

Andando avanti, PyJS è un compilatore molto leggero. Inoltre, PyJS fornisce anche il supporto Runtime per il debug degli errori di runtime.

---

- **Il compilatore SKULPT**

Il compilatore SKULPT esegue anche il codice Python scritto nel tuo browser web, direttamente. Rende il tuo codice Python più eseguibile. SKULPT è stato recentemente introdotto in un blog su una pagina del sito web. SKULPT può essere utilizzato anche per convertire il codice Python in HTML. È anche idoneo a compilare il codice, avendo una struttura JavaScript per il tuo browser web.

Poiché abbiamo discusso che SKULPT è un programma utilizzato da Python, non sono necessari moduli, file o impostazioni del server aggiuntivi per eseguire Python nel browser Internet. Ogni codice Python, scritto in SKULPT, verrà eseguito direttamente nel tuo browser Internet.

SKULPT è utilizzato anche da molti ingegneri web e sviluppatori web che desiderano sviluppare alcune applicazioni basate sul web che possono consentire agli utenti di eseguire programmi Python direttamente su un browser Internet.

---

- **Il compilatore BRYTHON**

BRYTHON è considerato uno dei compilatori per Python. Converte anche il tuo codice Python nel codice di JavaScript. Questo compilatore ha alcune caratteristiche uniche che possono trasformare il tuo codice e programma come un editor di terze parti per ottenere risultati straordinari nel minor tempo possibile. Inoltre, questo compilatore offre assistenza in tempo reale per molti dei moduli che supportano CPython. Il punto da ricordare è che questo compilatore supporta molti altri linguaggi e anche le loro versioni più recenti. La maggior parte delle volte, BRYTHON viene utilizzato per la programmazione basata sul Web lato utente. Questo compilatore è una compressione per il browser Python.

Andando avanti, puoi anche usare BRYTHON per lo sviluppo di applicazioni basate su 3D in Python.

---

- **Il compilatore Shed Skin**

Il prossimo nell'elenco è il compilatore Shed Skin per la programmazione Python. La caratteristica principale di questo compilatore è che cambia il codice orientato a Python in un programma C ++. L'unico inconveniente del compilatore Shed Skin è che non offre assistenza in tempo reale per molti bug e problemi normali. Inoltre, supporta pochissime librerie e moduli Python.

La maggior parte degli sviluppatori Python utilizza il compilatore Shed Skin per estrarre il codice di python scritto staticamente per ottenere il codice rivisto di C e C ++. Potresti trovare utile il compilatore Shed Skin perché questo compilatore presenta il supporto di visualizzazione di base.

---

- **3.2 IDE di Python**

Per lo sviluppo Python, puoi utilizzare un IDE, noto come ambiente di sviluppo integrato. Per questi scopi viene utilizzato principalmente un IDE.

- Un IDE può essere utilizzato per gestire il codice.

- È possibile utilizzare IDE per eseguire, sviluppare, eseguire il debug o produrre il codice, il compilatore o l'apparato.

- IDE può controllare script e sorgenti sul codice e sui compilatori.

La maggior parte degli IDE fornisce un numero più significativo di formati di programmazione e contengono molti punti salienti specifici. Tuttavia, avrai bisogno di informazioni per utilizzare correttamente gli IDE.

IDE è l'acronimo di Incorporated Development Environment. Un IDE è considerato uno strumento o un apparato di codifica che aiuta a meccanizzare la meraviglia dell'assemblaggio, del test, dell'alterazione e in molti altri modi, in un SDLC. Inoltre, fornisce un percorso a uno sviluppatore per eseguire, sviluppare e compilare il suo codice.

Ecco un elenco di alcuni dei più noti IDE di Python.

- PyCharm
- Spyder
- PyDev
- Atom
- Wing
- Jupyter Notebook
- Thonny
- Microsoft Visual Studio
- Eric Python

---

- **II PYCHARM IDE**

PYCHARM è un IDE cross-stage, ovvero un ambiente di sviluppo integrato cross-stage. Questo strumento è idealmente pensato per Python. PYCHARM è lo strumento più utilizzato al mondo ed è accessibile in entrambe le forme; in forma a pagamento e anche in forma gratuita e open source. PYCHARM ti fa risparmiare troppo tempo mentre ti occupi del tuo lavoro quotidiano.

Andando avanti, PYCHARM è un IDE molto utile per Python. Fornisce molte funzionalità speciali come il percorso di avventura veloce, la fruizione del codice automatico, il supporto per l'avanzamento remoto, il controllo rapido degli errori e il rimedio, il database e molti altri.

### **Caratteristiche speciali:**

- Debug efficace.
- Fornisce un efficiente codice route.
- Evidenzia in modo significativo gli errori.

---

### **● L'IDE SPYDER**

SPYDER IDE è il miglior IDE per gli sviluppatori di data science. È un IDE open source. La forma completa di SPYDER è "Scientific Python Development Environment". SPYDER è compatibile con i sistemi operativi Windows, MAC OS X e LINUX OS. SPYDER di solito include la dispersione del regista del bundle Anaconda, in modo che tu possa fare affidamento sul tuo.

Il fatto affascinante di SPYDER è che viene utilizzato per intendere i gruppi di interesse dei ricercatori dell'informazione che utilizzano Python. SPYDER si coordina bene con le librerie standard di informatica Python come Matplotlib, SciPy e NumPy.

SPYDER ha un vasto numero di funzioni IDE di base che potresti immaginare. Ad esempio, SPYDER è un gestore di codice Python con una funzionalità durevole per la struttura del linguaggio. Inoltre, INCLUDE l'adempimento per il codice Python, ed è persino un IDP, ovvero un programma di documentazione incorporato.

Andando avanti, un elemento diverso che potresti non vedere in altri IDE Python è che SPYDER contiene una "variabile voyager". Ti consente di visualizzare le informazioni, mentre utilizzi un design basato su tabelle, all'interno del tuo IDE.

### **Caratteristiche speciali:**

- IPython integrato
- Sintassi di qualità corretta

---

### **● L'IDE PYDEV**

L'IDE PYDEV è uno degli IDE Python molto richiesti al mondo. Per quanto riguarda gli sviluppatori Python, preferiscono PYDEV per tanti motivi. PYDEV ha una caratteristica che incorpora la combinazione "django", rientri accorti, fruizione del codice programmato e rientri quadrati e molti altri.

PYDEV è accessibile su sistemi operativi Windows, MAC OS X e sistemi operativi Linux. Potremmo considerare PYDEV come l'IDE open source accettato anche per il progresso basato su Java. Include un centro commerciale per articoli aggiuntivi e di espansione. Questa caratteristica rende PYDEV utile per un'ampia gamma di miglioramenti ed esercizi.

### **Caratteristiche speciali:**

- Contiene combinazione PyLint, debugger remoto, partecipazione a unit test.

- Ispezione e verifica del codice.

---

- **Atom IDE**

Atom IDE è l'IDE più popolare per Python, sviluppato da GitHub. È un IDE multiplatforma e open source.

In Atom IDE, gli esercizi iniziali e didattici ti illustrano passo dopo passo le funzioni dell'IDE interessato. Una tecnologia di avanzamento coordinata consente di avviare un lavoro altamente integrato tra due diverse piattaforme. Questo è il motivo per cui Atom IDE è altamente raccomandato per la gestione di programmi multiplatforma e per l'editing efficace ed efficiente del codice Python durante l'esecuzione dei programmi live.

Atom IE ha anche uno svantaggio. Gli sviluppatori, utilizzando Atom IDE, lo considerano lo strumento meno sicuro. Ma è considerato lo strumento migliore per amatori e principianti. Tuttavia, a parte i suoi pro e contro, Atom IDE è uno dei migliori strumenti per la scienza dell'informazione che ti consente di lavorare con linguaggi di programmazione di alto livello.

### **Caratteristiche speciali:**

- Ha un modulo "Markdown Preview Plus".
  - Mostra i risultati nelle finestre di runtime.
- 

#### **● The Wing IDE**

In primo luogo, Wing IDE è open source e gratuito. Ha alcuni punti salienti in esso che possono consentire la struttura della sintassi con auto-compimento, analisi e rientri.

Ha anche una versione a pagamento, chiamata Wing Pro. Il vantaggio dell'utilizzo di Wing Pro è che fornisce debug, evidenziazioni e molto altro al tuo codice Python.

### **Caratteristiche speciali:**

- Supporta l'avanzamento remoto, il miglioramento guidato dai test insieme allo unit test.
  - È personalizzabile e può anche avere espansioni.
- 

#### **● IDE del notebook Jupyter**

Jupyter IDE è un IDE creato per la struttura server-client. Ti consente di sviluppare e controllare i report del blocco note per il tuo programma basato su Python. Jupyter Notebook è stato rilasciato per IPython nell'ottobre 2014. Si tratta di un'applicazione Web che dipende dalla struttura client-server e consente inoltre di sviluppare, personalizzare e controllare i report creati su Scratch Pad o Blocco note.

Tuttavia, Jupyter Notebook dimostra di essere una parte fondamentale della cassetta degli attrezzi di ogni ricercatore di informazioni Python. È eccezionale per lo sviluppo di prototipi ed è anche utile per impartire appunti a una forma rappresentabile.



### **Caratteristiche speciali:**

- In questo IDE, il tuo codice può essere generato e modificato facilmente.
  - I notebook Jupyter includono una funzionalità per supportare i ribassi.
  - Questo IDE è ideale per i principianti nel campo della scienza dei dati.
- 

#### **● Thonny IDE**

Se sei un principiante nella programmazione Python e vuoi imparare e padroneggiare il linguaggio di programmazione Python, Thonny IDE è uno dei modi migliori per farlo. Thonny IDE è l'IDE più utilizzato dai principianti di Python ed è considerato l'IDE più facile da capire. Possiede un ambiente di sviluppo molto comune e semplice per la comunità di data science di Python.

### **Caratteristiche speciali:**

- In questo IDE, il debug è davvero facile e diretto.
  - Thonny IDE contiene la funzionalità di completamento automatico del codice insieme a errori di debug.
- 

#### **● IDE di Microsoft Visual Studio**

Microsoft Visual Studio IDE è l'IDE più adatto per investigare e improvvisare attività basate sul web. Questo IDE è un generatore di codice open source e ha il suo mercato. Inoltre, questo IDE è stato creato da Microsoft.

### **Caratteristiche speciali:**

- L'IDE ti consente di codificare in Python nell'ambiente dello studio visivo, che è una caratteristica davvero unica di questo IDE.
  - Questo IDE è disponibile in entrambe le forme; forma a pagamento e gratuita.
- 

- **Perché gli IDE e gli editor di codice!**

La domanda sorge spontanea! Perché hai bisogno di un gestore editoriale del codice o di un IDE!

In genere! Utilizzando un gestore editoriale del codice o un IDE, puoi dare indicazioni a un terminale in modo che tu possa essere in grado di eseguire il tuo codice e progetti

Dobbiamo menzionare che l'adozione dello stesso processo per codici enormi e lunghe attività di programmazione può deluderti. Per lo più, ti deluderà se non sai come utilizzare le applicazioni di traduzione della linea di direzione.

Andando avanti, l'utilizzo di un IDE può rendere la programmazione o la codifica più semplice, facile da capire e divertente. Gli IDE sono fondamentalmente dispositivi o strumenti di codifica che consentono di scrivere, testare, eseguire, risolvere i problemi, eseguire il debug, compilare ed eseguire il codice. Gli IDE e le applicazioni di editor di codice sono l'approccio migliore per i principianti per un lavoro veloce. Questi IDE possono fornire la finitura automatica del codice e possono consentire di comprendere la struttura della frase e della sintassi.

---

- **3.3 Interpreti Python**

Ci sono molti interpreti in Python che possono essere usati per allineare, allineare e perfezionare i tuoi codici Python. Puoi eseguire Python in molti modi diversi. È impossibile eseguire la programmazione Python senza l'aiuto di un maggior numero di interpreti. Python, come linguaggio di programmazione per computer di alto livello, è davvero facile da capire ed è eseguibile con l'aiuto dei suoi interpreti.

Per quanto riguarda gli interpreti, gli interpreti sono generalmente considerati come programmi che consentono di eseguire le linee guida scritte come un codice.

---

- **CPYTHON**

L'interprete, CPYTHON, supporta le versioni più recenti di Python. Supporta fino a Python 3.7. CPYTHON è l'interprete più facilmente disponibile della programmazione Python. Inoltre, fornisce una capacità esterna per molti altri software. Di solito, CPYTHON può essere considerato anche un compilatore.

Di solito, CPYTHON supporta davvero molte altre piattaforme e fornisce un'esperienza fluida alla maggior parte degli utenti. Questo interprete è molto richiesto dai professionisti, dagli sviluppatori di software e dagli esperti di linguaggio orientati al computer.

---

- **Jython**

Jython è un noto interprete Python che una volta si chiamava JPYTHON. Jython è solitamente implementato sulla piattaforma di Java e JavaScript. È stato sviluppato e annunciato alla fine degli anni '90 in modo che fosse in grado di cambiare C con Java e JavaScript per prestazioni migliori. Jython contiene molte estensioni e funzionalità uniche. Include una funzionalità di compilazione statica e dinamica. Grazie a questa caratteristica, consente agli sviluppatori di software di eseguire più operazioni. La programmazione in Jython utilizza JavaScript e moduli invece dei moduli di Python. Un'altra caratteristica unica di Jython è che può collegare il database Python a Java Virtual Machinery.

Jython consente ai suoi utenti di importare qualsiasi classe di Java poiché è un modulo Python. Gli sviluppatori possono scrivere il codice prima in Java e poi trasformare il loro codice in Python. Grazie a questa capacità, Jython è considerato una delle migliori scelte per gli sviluppatori di software in tutto il mondo.

---

- **Stackless Python**

Stackless Python è un tipo molto efficiente di interprete Python. È stato rilasciato e annunciato nel 1998. Supporta anche le versioni più recenti di Python, fino a Python 3.7. Questo interprete evita di usare lo stack C e C. Tuttavia, utilizza lo stack C in alcuni punti, ma prima garantisce che la chiamata alle funzioni deve essere chiara e diretta.

Stackless Python ha una caratteristica unica per il micro-threading. Questa funzionalità consente all'utente di evitare l'onere dell'overhead, che è associato al threading del sistema operativo tipicamente utilizzato. Stackless Python assiste anche con la pianificazione delle attività di routine e i canali di comunicazione. Stackless Python è utilizzato principalmente dagli sviluppatori di giochi. Utilizza anche molte librerie Python.

---

- **PyPy**

PyPy è uno degli interpreti più utilizzati. È veloce e può essere utilizzato come alternativa a Python come linguaggio. PyPy è stato sviluppato e rilasciato nel 2002.

La caratteristica principale di PyPy è che mostra vicinanza a CPython nel contesto per la visualizzazione e l'esecuzione. La versione recente di PyPy è anche più efficiente di CPython. Dobbiamo ricordare che CPython funge solo da interprete e PyPy può essere utilizzato come linguaggio o compilatore. Puoi usare PyPy in quanto è più versatile, flessibile e veloce di CPython.

PyPy può supportare molti sintassi e codici per Python e vari altri linguaggi. Supporta anche molti dei framework Python in modo abbastanza efficace. A volte, PyPy può fornire supporto anche ad altri linguaggi dinamici. Di solito, PyPy viene utilizzato dagli sviluppatori per aumentare le prestazioni del loro codice, ed è anche utilizzato per lavorare con vari dialetti del codice Python.

---

- **3.4 Editor di testo Python**

- **Testo sublime**

Per la programmazione Python, sublime editor di testo è uno degli editor di testo più conosciuti per la programmazione Python. È pieno delle migliori funzionalità che un editor di testo può avere. Questo editor di testo è stato fondamentalmente sviluppato nel linguaggio di programmazione C ++, ma puoi anche usarlo per supportare Python.

La sua versione aggiornata è idonea a supportare molti linguaggi di programmazione. Jon Skinner è stato l'unico a crearlo e si è avvicinato al mercato a metà del 2007. Per sviluppare questo editor di testo, ha seguito queste tre regole di base.

- Il suo contenuto non è nascosto dalle finestre.
- Utilizza tutto lo spazio disponibile ovunque sullo schermo che potrebbe essere sensibilmente normale, poiché l'utilizzo a schermo intero lo rende migliore per un facile montaggio.
- È irrilevante e l'interfaccia discreta è uno dei punti chiave che ha tenuto a mente. In questo strumento, hai l'alternativa per dare la priorità ai contenuti e puoi evitare più strumenti.

### **Vantaggi chiave:**

- È veloce e ha un numero minimo di bug.
- È in grado di aprire record giganteschi.
- Supporta molti linguaggi di programmazione e i loro dialetti.

---

### ● **Editor di testo GNU / Emacs**

È un editor di testo GNU altamente raccomandato da molti sviluppatori. Questo editor di testo è stato sviluppato da Richard Stallman e per più di vent'anni è rimasto l'editor di testo più popolare tra gli sviluppatori di software e i professionisti della programmazione. Il signor Richard Stallman, che è il proprietario di questo editor di testo, lo ha reso gratuito per ogni utente.

L'editor di testo GNU Emacs utilizza principalmente personalizzazioni uniche per un approccio progressivo in molti linguaggi di programmazione per computer.

L'estensione Elpy è un'estensione utilizzata da questo editor di testo. Questo editor di testo ha molte caratteristiche che lo separano dagli altri editor di testo. Ad esempio, fornisce una struttura della frase che separa i segmenti dei record e gli spazi tra i testi per avere la migliore formazione in un archivio.

### **Vantaggi chiave:**

- È gratuito e fornisce un ambiente di programmazione completamente mobile.
- Fornisce l'espansione automatica dei segmenti necessari per la struttura del record.
- Supporta più sistemi operativi con un sistema di colore a 24 bit.

---

### **● CAPITOLO 4: Confronto di Python con altri linguaggi**

Python viene solitamente paragonato ad altri linguaggi di programmazione per computer di alto livello e ai loro dialetti. In questo capitolo parleremo del confronto di Python con altri principali linguaggi e dialetti di programmazione per computer. Questa relazione e correlazione di Python con altri linguaggi di programmazione ci motiverà a selezionare Python rispetto ad altri linguaggi di programmazione. Grazie a questo confronto, sarai in grado di sapere che altri linguaggi di programmazione per computer di alto livello certificabili di solito dirigono Python. Comprendiamo che questi punti sono fattori eccezionali, ma potresti imparare a utilizzare diversi linguaggi di programmazione per computer durante l'utilizzo di Python. Per questo motivo, questa relazione e correlazione sono essenziali.

---

#### **● 4.1 Confronto tra Python e C ++**

Sia Python che C ++ (C plus plus) sono i linguaggi di programmazione per computer con i propri dialetti. Entrambi sono usati per scopi ampiamente utili. Devi ricordare che sia il dialetto di Python che il dialetto C ++ variano l'uno dall'altro rispetto a molti punti di vista. C ++ è la forma avanzata del linguaggio C con molti utili aggiornamenti e upgrade. Una specialità del C ++ è che fornisce il componente per raccogliere i dati e lo utilizza in una classe.

D'altra parte, il dialetto di Python è globalmente utile ed è uno dei dialetti di programmazione di computer e software di più alto livello. In Python, possiamo utilizzare una variabile in modo semplice e chiaro senza dichiararla e descriverla durante la scrittura del codice di un programma.

Per quanto riguarda C ++, in C ++, qualsiasi codice o programma richiede di essere richiamato su ogni framework funzionante su cui il codice deve essere eseguito.

A questo proposito, Python ti consente di "scrivere una volta - eseguito in qualsiasi luogo e piattaforma". Questa caratteristica fornisce unicità a Python per continuare a essere eseguito su ogni framework funzionante. Devi solo introdurre Python in quel framework.

Il C ++ è incline alle perdite di memoria. Ciò significa che non ti fornisce la raccolta dei rifiuti. Inoltre, C ++ utilizza puntatori a un livello molto alto.

Python ti fornisce un accumulo di spazzatura integrato. Inoltre, fornisce un processo di porzione di memoria dinamica, che offre l'opportunità alla memoria competente di utilizzare la scheda.



Andando avanti, oggi, il C ++ viene utilizzato principalmente per la pianificazione delle apparecchiature e per lo sviluppo di progetti integrati. Questo concetto è stato introdotto per la prima volta con la versione aggiornata di C ++, che ha sviluppato il concetto di programmazione strutturata e ha introdotto il concetto di sistemi embedded e l'uso di C ++ in essi.

A questo proposito, potresti dire che Python è usato principalmente come linguaggio di scripting e, in molti casi, puoi usare Python anche per ragioni non di scripting. Devi tenere presente che Python possiede un'applicazione eseguibile indipendente, inclusa l'assistenza di alcuni degli strumenti che ti forniscono anche un'opportunità per la programmazione senza script.

---

## ● 4.2 Confronto tra Python e C #

Sappiamo che Python è stato introdotto per la prima volta per assomigliare alla lingua inglese. In Python, questi numeri più significativi di articolazioni sono un po 'difficili da esaminare. In particolare a questo proposito, puoi utilizzare nomi di variabili appropriati. Inoltre, a causa della sintassi grammaticale di base, non ci sono sviluppi entangled, in Python. Ad esempio, innumerevoli modificatori di parole, sezioni sintattiche C e C ++ come un framework di sviluppo e molti altri approcci per introdurre diversi fattori. Si può dire che tutto, a questo proposito, rende il codice di Python facile da capire, imparare e scrivere.

Parallelo a questo è C # (C-sharp). A causa dell'ereditarietà del linguaggio, C # ha ereditato molti concetti da C ++, JavaScript e Java. Questi concetti vengono utilizzati per la prima volta nella struttura della frase "tipo C". Inoltre, il dialetto della struttura del linguaggio C # rende importante aderire agli standard specifici quando si provano le proprie tecniche sul linguaggio o per acquisire classi, a cui si uniscono per lo più altri gruppi di programmatori ed esperti di programmazione. Non dovresti ignorare i dialetti del codice, che deve essere racchiuso tra parentesi o oggetti di scena. A differenza di C #, Python non possiede tutto questo. Python utilizza un diverso tipo di spostamento, che consente al codice di apparire perfetto.

Inoltre, per quanto riguarda la composizione dei contenuti, vale soprattutto la pena fare riferimento a qualsiasi progetto. Python dice che i contenuti sono solo contenuti, puoi semplicemente registrarli con il tuo codice che può essere efficacemente eseguito da un compilatore, IDE o un editor di testo. Puoi aprirli al tuo manager preferito e iniziare a lavorare con loro. E subito dopo, puoi eseguirlo rapidamente ancora una volta. Per questo, ti troverai in una posizione ampiamente favorevole se non ci sono compilatori o IDE nelle vicinanze. Inoltre, mentre si utilizza Python come modalità di programmazione, sarà molto semplice compilare un contenuto cross-stage e multiplatforma. Non avrai bisogno di ricompilarlo su un'altra piattaforma o livello.

Dovresti sapere che C # richiede un IDE per scrivere, compilare ed eseguire un tipico codice di programmazione. Come uno o più C #, possiede una guida abbastanza solida per diverse versioni del framework orientato al sistema operativo Windows quando si sviluppa contenuto per il sistema operativo Windows. Supponiamo che tu debba lavorare su diversi dispositivi in modo da poter lavorare con le loro librerie, WMI, WMI, ecc. C # ti consente anche di utilizzare "WinForms". Rende C # estremamente semplice creare un'interfaccia utente grafica se richiede tutte le cose considerate.

Forse potresti chiedere quale lingua è migliore! C # o Python! Non esiste una risposta del genere!

Entrambe le lingue hanno i loro pro e contro, ma Python è più semplice da imparare e da capire. Inoltre, Python ha un numero più significativo delle sue librerie open source rispetto a C #. Tuttavia, potresti dire che la libreria di base di C # è superiore e molto migliore rispetto a quella di Python. Inoltre, devi tenere presente che C # è un linguaggio più evidenziato e la presentazione, che possiede, è avanzata, molto più alta e veloce, rispetto a Python, come modalità di programmazione.

---

### ● 4.3 Confronto tra Python e Java

Se confrontiamo la velocità di esecuzione dei programmi Java con la velocità di Python! I programmi Java possiedono una maggiore velocità di esecuzione rispetto ai programmi Python. Tuttavia, lo sviluppo di programmi richiede uno sforzo minore in Python poiché il codice di Python è semplice e relativamente facile. In Python, è facile lavorare a diversi livelli e tipi di dati. Ad esempio, in Python, quando valuti l'articolazione, non cerca di raccogliere tempo. A quel tempo, Python crea un'attività di espansione adatta, che è un numero intero di produzione.

Pertanto, potresti considerare Python un linguaggio completamente migliorato, mentre la maggior parte degli sviluppatori considera Java come un linguaggio di esecuzione di basso livello. Inoltre, sia Python che Java insieme costituiscono una straordinaria partnership. La maggior parte dei segmenti di Python può essere utilizzata nella programmazione Java e, dopo essere stata combinata con i codici Java, è possibile utilizzarli per modellare la struttura delle frasi in Python. D'altra parte, Python può essere utilizzato per sviluppare e strutturare le parti in modo che la struttura Python possa essere utilizzata nella programmazione Java.

Per aiutare questo tipo di ereditarietà e avanzamento, la struttura di frasi di Python scritta in Java è un altro lavoro in corso, che consente di chiamare il codice Python durante la programmazione in Java e viceversa. In tale esecuzione, il codice sorgente di Python viene utilizzato come bytecode Java.

Se possiedi un background di programmazione, sai già che Java è considerato un linguaggio accuratamente incorporato. Significa che i nomi delle variabili in Java dovrebbero essere proclamati in modo inequivocabile. D'altra parte, abbiamo un'altra lingua che viene progressivamente composta; Python. Nella programmazione Python, non è necessaria alcuna affermazione. Ci sono molte domande sulla composizione più potente e misurata in varie sintassi e dialetti di programmazione. Ma è da notare un'idea. Python è un linguaggio di programmazione per computer ampiamente adattabile con la struttura della frase più chiara. Questo rende Python uno straordinario linguaggio di programmazione per la composizione di contenuti e lo sviluppo costante di varie applicazioni per molti campi.

Inoltre, Java consente anche di sviluppare applicazioni cross-stage e multiplatforma, mentre Python è perfetto per lo sviluppo di tutti i framework di lavoro all'avanguardia. Quindi, potresti dire che Java è molto complesso per i piedi teneri rispetto a Python.

Inoltre, quando usiamo Java con Python, semplifica semplicemente il tuo codice. Quando vuoi che il tuo codice venga eseguito da qualsiasi altra piattaforma, è meglio scegliere Java come modalità di programmazione. Il vantaggio aggiuntivo dell'utilizzo di Java, come modalità di programmazione, è che ti consente di sviluppare applicazioni organizzate mentre Python non può funzionare in questo modo.

Andando avanti, Java come linguaggio è molto più contorto di Python. Quando non si dispone di risorse specializzate per l'apprendimento di Java. Non sarà semplice da capire! Anche in questo caso, Java viene utilizzato per programmare in varie condizioni estreme e per risolvere alcuni problemi di programmazione; solo Java può aiutarti.

---

#### ● 4.4 Confronto tra Python e JavaScript

Come sappiamo, Python è un linguaggio di programmazione orientato agli oggetti. Puoi confrontare il sottoinsieme basato sugli oggetti di Python con la struttura della frase di JavaScript. Come JavaScript, Python possiede uno stile di programmazione che utilizza i limiti e i fattori fondamentali senza nemmeno manipolare le definizioni di classe.

Indipendentemente da ciò, per JavaScript ci sono così tanti problemi e complessità nella sua sintassi e formazione del codice. D'altra parte, Python ti supporta nella creazione di programmi molto più grandi e ti consente di riutilizzare meglio il tuo codice attraverso un vero e proprio dialetto di programmazione orchestrato, dove le classi e l'ereditarietà si aspettano un'attività critica.

---

- **4.5 Confronto tra Python e Go Programming language**

Il linguaggio di programmazione Go è un linguaggio abbastanza adattabile, proprio come Python. Entrambi i linguaggi non richiedono eccessivi esercizi di sintassi didattici ed entrambi sono facili da scrivere, capire ed eseguire. Il linguaggio di programmazione Go è noto anche come linguaggio Go ed è stato sviluppato e pubblicato da Google a metà del 2009.

D'altra parte, Python possiede molti modelli ideali per la programmazione di computer e ha una propria libreria molto ampia. I modelli ideali di Python includono modalità di programmazione basate sugli oggetti, di base, pratiche e procedurali.

Il linguaggio di programmazione Go sostiene una visione multi-mondo, come simultanea, pratica e procedurale. La sua struttura della frase è in qualche modo simile a C. Tuttavia, ha introdotto molti aggiornamenti per migliorare punti salienti come semplicità e sicurezza.

Andando avanti, si osserva che i linguaggi di programmazione Python e Go hanno troppe differenze. Ad esempio, il linguaggio di programmazione Go non ha la caratteristica di "provare tranne", invece, consente a una funzione di risolvere i suoi problemi contemporaneamente alla sua conclusione. Pertanto, prima di utilizzare qualsiasi funzione, è necessario verificare che eventuali errori logici o di sintassi non vengano chiamati o ripetuti.

Andando avanti, Python è comunemente utilizzato nello sviluppo di applicazioni web, mentre l'obiettivo di base del linguaggio di programmazione Go è quello di essere un linguaggio di sistema. Tuttavia, possiamo utilizzare il linguaggio di programmazione Go per sviluppare alcune delle applicazioni basate sul web. Inoltre, Python non possiede la gestione della memoria, ma il linguaggio di programmazione Go fornisce ai suoi utenti una gestione efficiente della memoria. Un'altra differenza è che in Python non hai un meccanismo di concorrenza, mentre, nel linguaggio di programmazione Go, c'è un meccanismo di concorrenza integrato.

In termini di sicurezza, Python è un'opzione migliore. Python è un linguaggio fortemente tipizzato con una struttura della frase perfetta, che viene compilata, in modo che possa aggiungere un ulteriore livello di sicurezza mentre, nel linguaggio di programmazione Go, ogni fattore deve avere un ordinamento correlato ad esso. Ciò implica che uno sviluppatore non può lasciarsi sfuggire le sottigliezze, perché a lungo termine causeranno bug. Python ha un numero più ampio di librerie rispetto al linguaggio di programmazione Go. La sintassi di Python è più concisa del linguaggio di programmazione Go. Potremmo dire che Python è l'opzione più conosciuta per la programmazione e lo sviluppo di base, ma a volte diventa difficile scrivere programmi complessi utilizzando Python come modalità di programmazione. Tuttavia, il linguaggio di programmazione Go è di gran lunga migliore per la programmazione complessa.

Oltre a questo, c'è un'altra grande differenza tra il linguaggio di programmazione Python e Go. Python è un linguaggio che può essere digitato e utilizzato dinamicamente, mentre il linguaggio di programmazione Go non è dinamico.

Per questo motivo, i programmatori Python sono in grado di comprendere rapidamente la sintassi del linguaggio di programmazione Go, senza alcun problema significativo. Ecco alcune delle altre differenze tra le due lingue.

- La programmazione Python si concentra su una sintassi chiara, semplice e di facile comprensione e fatti grammaticali impeccabili, mentre il linguaggio di programmazione Go funziona correttamente solo con elevata chiarezza e precisione.
- La composizione statica del linguaggio di programmazione Go si allinea con la sintassi e la struttura delle frasi standard di Python.

Quindi, dopo tutto questo, possiamo dire che Python è l'opzione più utilizzata per sviluppatori e ingegneri di software in tutto il mondo. Ma Python è un linguaggio scritto dinamicamente; le sue prestazioni sono leggermente inferiori rispetto al linguaggio di programmazione Go. Pertanto, possiamo suggerirti di utilizzare entrambi i linguaggi di programmazione contemporaneamente. Dovresti dare la priorità al linguaggio di programmazione Go per i tuoi codici, altrimenti dovresti usare Python.

---

#### ● 4.6 Confronto tra Python e Node.js

Prima di tutto, dovremmo sapere che Node.js non è un linguaggio di programmazione per computer come Python. Possiamo considerare Node.js come un dominio di runtime per JavaScript. Quindi, scrivere il codice in Node.js significa che stai usando un linguaggio molto simile sul front-end del tuo programma e anche sul back-end.

---

#### ● Circostanze favorevoli di Python su Node.js



A un livello di sviluppo più dettagliato, JavaScript può essere difficile da usare e da comprendere per gli sviluppatori con poca esperienza con Node.js. Potrebbero esserci degli errori fondamentali, che ostacolano contemporaneamente l'avanzamento. Questa non è la situazione con Python. Poiché Python è molto semplice da usare per i programmatori meno esperti.

---

- **Più applicazioni**

Node.js viene solitamente utilizzato per la parte di un programma, che viene utilizzata per le applicazioni basate sul web, mentre gli usi e gli effetti di Python sono molto più degni di nota. Il punto da ricordare è che l'inclusività e la flessibilità di Python sono più comuni tra i motivi principali per cui Python è perfettamente adatto per i progressi. Ad esempio, scienza dei dati.

---

- **Migliore utilizzo**

I framework JavaScript e le condizioni di runtime attualizzano il linguaggio in modo straordinario e inaspettato, ad esempio Node.js non fa eccezione. Ad essere onesti, l'ecosistema e il framework di JavaScript è in qualche modo un disastro.

D'altra parte, Python non ha questo problema specifico, motivo per cui Python è più diretto e semplice da codificare. Questa specialità rende Python più veloce da codificare, anche se Node.js non è affatto lento.

Inoltre, è necessario disporre del know-how necessario di JavaScript per utilizzare Node.js come modalità di programmazione, poiché si sta lavorando su un linguaggio molto simile sul front-end e sul back-end del programma.

---

- **Ecosistema meno ostinato**

Per quanto riguarda l'ecosistema di Node.js, le librerie per Node.js sono generalmente molto semplici e si attengono a una singola causa, qualunque cosa siano state create per eseguire. Ciò consente agli sviluppatori di selezionare ciò che devono utilizzare nel loro programma e quando devono utilizzarlo.

---

- **Sviluppo rapido e una vasta rete**

Dal 2012, Python è ben noto per la sua rete affidabile e incredibile e il sistema di supporto. Al giorno d'oggi, dopo alcuni aggiornamenti, anche JavaScript è considerato affidabile. JavaScript è in continuo sviluppo senza alcun sintomo di interruzione e sta guidando il gruppo delle strutture delle frasi e dei dialetti di programmazione più potenti in crescita nel settore dello sviluppo software.

---

- **Storia dei progressi di Python e JavaScript**

JavaScript ha visto molte fasi di sviluppo. Quando è stato sviluppato per la prima volta, utilizzava solo la codifica e le frasi ei suoi vecchi adattamenti stanno creando problemi molto simili, anche ai giorni nostri.

Quindi, l'importanza dello sviluppo di JavaScript ha portato con sé un costante sciame di aggiornamenti e problemi di documentazione critici. Questa è l'unica ragione per il danno alla natura di Node.js.

Andando avanti, Python ha una posizione elevata qui rispetto a Node.js. Sia l'inclusione che la documentazione sono migliori in Python rispetto a Node.js. Questo è il motivo per cui Python ha costantemente guidato JavaScript.

---

- **Esecuzione e velocità**

Node.js cerca di combattere con l'esecuzione di una grande quantità di incarichi allo stesso tempo. Nel caso citato, se il tuo codice non è composto correttamente e perfettamente nel complesso, il tuo programma si rivelerà inefficace e funzionerà gradualmente.

Questo può anche accadere in Python. Tuttavia, i framework Python, ad esempio Django, accompagnano il supporto utente istantaneo per assistere nel codice, nella struttura delle frasi e nel programma, sopportando un carico maggiore. Questo è un altro motivo principale per cui Python rende la vita più semplice ai suoi sviluppatori.

---

- **4.7 Confronto tra Python e PHP**

Da una prospettiva evolutiva e improvvisata, PHP è un linguaggio di programmazione per computer basato sul web e orientato al web. Un'applicazione PHP è molto simile a molti contenuti individuali, magari con una sezione o un punto semantico solitario.

Anche in questo caso, Python è un linguaggio completamente adattabile che può essere utilizzato anche per sviluppare applicazioni basate sul web. In Python, un'applicazione Web dipende da un'applicazione innegabile impilata nella memoria con il suo stato interno. Per scegliere tra PHP e Python per lo sviluppo di applicazioni basate sul web, dobbiamo concentrarci su queste due qualità.

---

- **Python e PHP per la correlazione del miglioramento web**

Nella programmazione di computer, prevalenza e modelli significano molto di questi tempi. Circa tutti gli utenti e i proprietari di articoli devono utilizzare i progressi più famosi e pubblicizzati per i loro prodotti. In tal caso, puoi dimostrarti un programmatore di talento senza clienti e occupazioni, solo con l'uso della tua immaginazione.

---

- **Frameworks**

Anche l'assortimento della tua macchina, utensile e dispositivo è fondamentale mentre prendi in considerazione il tuo progetto. Caratterizza la chiarezza, la semplicità, la leggibilità, la comprensibilità e la sistemazione del progetto. È possibile che un progetto offra molte strutture per vari incarichi. Uno sviluppatore di software o web dovrebbe essere certo di non dover fare tutto senza alcun framework. Questi framework possono consentirti di creare applicazioni sorprendenti ed eccezionali.

Di nuovo, Python non ha così tanti framework. I framework più utilizzati per lo sviluppo di applicazioni web-based sono Django e Flask. Ma possiamo assicurarti che questo cambierà presto a causa della rete in via di sviluppo e dei futuri framework di Python.

---

- **Capitolo 5: Variabili, operatori e tipi di dati di Python**

Tipi di dati, variabili e operatori sono i concetti fondamentali di Python, in quanto linguaggio di programmazione, sono considerati gli elementi costitutivi di base. Di solito li usiamo per programmare i nostri progetti e per ottenere i risultati richiesti attraverso la loro funzionalità. I tipi di dati sono il concetto più essenziale e nessuno può comprendere la programmazione Python senza prima capirne i meccanismi che sono molto simili ad altri linguaggi ma molto più flessibili, ma lo vedremo più avanti. In questo capitolo, discuteremo alcune variabili essenziali, operatori e tipi di dati di Python, come linguaggio di programmazione per computer.

---

- **5.1 Variabili di Python**

Variabile è un altro nome per gli identificatori Python. Variabile è un termine utilizzato per indicare una zona di memoria di una macchina o di un dispositivo. In Python, non è necessario decidere questo tipo di fattori poiché Python è una sorta di linguaggio di programmazione dedotto ed è abbastanza astuto da ordinare le sue variabili.

Inoltre, possiamo dire che le variabili in Python sono locazioni di memoria, con diversi tipi di dati, come numeri interi o caratteri. Le variabili in Python sono modificabili e manipolabili perché utilizzano un insieme di varie operazioni.

In ogni caso, le variabili necessitano di una lettera o di un trattino basso per essere inizializzate. Si consiglia di utilizzare lettere minuscole come nomi di variabili. In Python, sledge e Mallet, entrambi sono due elementi eccezionali.

---

- **La denominazione di variabili o identificatore**

I fattori sono le situazioni degli identificatori. Una variabile viene utilizzata per concepire i coefficienti letterali e gli interi utilizzati nel programma. Per Python, gli standard per nominare una variabile sono riportati di seguito.

- Il carattere essenziale di un identificatore deve essere una lettera o un trattino basso "\_".
- Ciascuno dei caratteri oltre ai caratteri essenziali può essere una lettera disposta da lettere minuscole "az", maiuscole "AZ", trattini bassi o cifre "0-9".
- Il nome di una variabile non deve contenere alcuna zona vuota o vuota, o alcun carattere speciale o straordinario, come "!", "@", "#", "%", "^ and, \*".

- Il nome di una variabile non deve assomigliare ad alcun slogan rappresentato nella sintassi del programma Python.
- In Python, le variabili fanno distinzione tra maiuscole e minuscole. Ad esempio, sono cool e sono cool non è proporzionato.
- Istanze di identificatori considerevoli: n696, \_v, v\_69, ecc.
- Istanze di identificatori non validi: 5a, v% 69, x69, ecc.

---

- **Assegnazioni multiple**

Python consente di mantenere un incentivo a vari identificatori in un'unica spiegazione, che di solito è chiamata assegnazioni varie. Può essere applicato in due modi diversi dichiarando un incentivo solitario per più identificatori contemporaneamente o relegando varie qualità a numerose variabili in momenti diversi.

### **Esempio 1:**

Apri la console Python o l'IDE e scrivi il comando per dichiarare le variabili.

```
>>> n=v=w=69
>>> print
>>> print (n, v, w)
```

### **Produzione:**

Quando digiti il comando per stampare il valore delle variabili, l'output sarà qualcosa del genere.

```
>>> 69, 69, 69
>>>
```

### **Esempio - 2:**

```
>>> n, v, w = 69, 74, 36
```

```
>>> print
>>> print (n)
>>> print (v)
>>> print (w)
```

### **Produzione:**

Per l'output,

Quando digiterai il tuo comando

```
>>> print (n)
```

La tua console stamperà "69"

Quando digiterai il tuo comando

```
>>> print (v)
```

La tua console stamperà "74"

Quando digiterai il tuo comando

```
>>> print (w)
```

La tua console stamperà "36"

---

## ● 5.2 Operatori in Python

In generale, gli operatori sono i token sintattici specifici della lingua che richiedono alcune azioni per essere eseguiti. Gli operatori derivano principalmente dai concetti di matematica. Ad esempio, "Segno di moltiplicazione (\*)" è un operatore utilizzato nella programmazione Python. È usato per moltiplicare due numeri.

In Python, gli operatori sono delineati come una rappresentazione simbolica di una funzione che compie un'azione particolare tra due operandi per ottenere alcuni risultati specifici e desiderati. In pratica gli operatori sono delle espressioni, delle strutture di codice, che ci permettono di eseguire dei procedimenti logici o matematici. L'assortimento di operatori fornito da Python è rappresentato come segue. Di seguito sono riportati alcuni operatori comunemente utilizzati per eseguire operazioni specifiche:

- Operatori aritmetici
- Operatori di confronto
- Operatori di assegnazione
- Operatori logici
- Operatori bit per bit
- Operatori di appartenenza
- Operatori di identità

In questa parte discuteremo alcuni degli operatori sopra menzionati.

---

- **Operatori aritmetici**

Gli operatori aritmetici vengono utilizzati per eseguire particolari operazioni aritmetiche per ottenere i risultati desiderati. In questo caso, vengono presi due operandi e, tra di essi, viene eseguita l'attività tramite un operatore, risultante in un valore desiderato, specifico e assoluto.

Ecco alcuni degli operatori aritmetici critici e utili, comunemente usati in Python.

- Aggiunta "+"



- Sottrazione "-"
- Divisione "/"
- Moltiplicazione "\*"
- Resto "%"

Una descrizione dettagliata di questi operatori:

- **Operatore addizione "+"**

Questo operatore viene utilizzato per eseguire la funzione di addizione o somma tra due operandi.

**Esempio:**

```
>>> n, v = 25, 69
```

```
>>> n + v
```

La tua console stamperà "94", in questo caso.

- **Operatore di sottrazione "-"**

Questo operatore viene utilizzato per prendere il primo operando e sottrae il secondo operando dal primo.

**Esempio:**

```
>>> n, v = 69, 25
```

```
>>> n - v
```

La tua console stamperà "44", in questo caso.

- **Operatore di divisione "/"**

Questo operando prende il secondo operando e divide il primo operando sul secondo operando e fornisce il quoziente, come output.

**Esempio:**

```
>>> n, v = 4, 2
```

```
>>> n / v
```

La tua console stamperà "2.0", in questo caso.

- **Operatore di moltiplicazione "\*"**

Come spiegato in precedenza, questo operatore esegue l'operazione di moltiplicazione tra il primo operando e il secondo.

**Esempio:**

```
>>> n, v = 4, 2
```

```
>>> n * v
```

La tua console stamperà "8", in questo caso.

- **Operatore rimanente "%"**

Questo operatore è responsabile dell'operazione di divisione e ottiene il resto come output.

**Esempio:**

```
>>> n, v = 4, 2
```

```
>>> n % v
```

La tua console stamperà "0", in questo caso.

---

- **Operatore di confronto in Python**

Gli operatori di confronto, in Python, sono usati per confrontare due operandi e restituiscono un tipo booleano, cioè TRUE o FALSE, rispettivamente.

- **==**

True: questo operatore viene utilizzato se e solo se i valori sono logicamente uguali e veri.

- !=

True: questo operatore viene utilizzato quando i valori sono veri ma disuguali.

- <=

True: questo operatore viene utilizzato quando il primo operando è minore o uguale al secondo operando.

- >=

True: questo operatore viene utilizzato quando il primo operando è maggiore o uguale al secondo operando.

- <>

True: questo operatore viene utilizzato se e solo se i valori non sono uguali.

- >

True: questo operatore viene utilizzato quando il primo operando è maggiore del secondo operando.

- <

True: questo operatore viene utilizzato quando il primo operando è inferiore al secondo.

---

- **Operatori di assegnazione in Python**

In Python, usiamo operatori di assegnazione per assegnare il valore all'operando sinistro, dell'espressione sul lato destro.

● =

Spesso, questo operatore viene utilizzato per assegnare il valore dell'espressione destra all'operando sinistro.

● +=

Questo operatore viene utilizzato per costruire la stima dell'operando sinistro mediante la stima dell'operando corretto e nominare l'alterato un incentivo all'operando sinistro.

**Esempio:**

```
>>> n = 2, v = 4
```

```
>>> n += v
```

Questo sarà equivalente a

```
>>> n = n + v
```

```
>>> print (n)
```

E la tua console potrebbe stampare il valore se n come "6".

● -=

Per quanto riguarda questo operatore, diminuisce la stima dell'operando sinistro stimando l'operando corretto e distribuisce l'incentivo modificato all'operando sinistro.

**Esempio:**

```
>>> n, v = 4, 2
```

```
>>> n -= v
```

Questo sarà equivalente a

```
n = n - v
```

```
>>> print (n)
```

E la tua console potrebbe stampare il valore se n come "2".

● \*=

Aumenta la stima dell'operando sinistro tramite la stima dell'operando corretto e nomina l'alterato un incentivo di nuovo all'operando sinistro.

**Esempio:**

```
>>> n, v = 4, 2
```

```
>>> a * = b
```

Questo sarà equivalente a

```
n = n * v
```

```
>>> print (n)
```

E la tua console potrebbe stampare il valore se n come "8".

- %=

Questo operatore è responsabile di Divide la stima dell'operando di sinistra per quella dell'operando corretto e assegna l'aggiornamento all'operando di sinistra.

**Esempio:**

```
>>> n, v = 4, 2
```

```
>>> a %= b
```

Questo sarà equivalente a

```
n = n % v
```

```
>>> print (n)
```

E la tua console potrebbe stampare il valore se n come "0".

---

- **Operatori logici in Python**

Per quanto riguarda le nostre vite reali, a volte, dobbiamo fare scelte difficili basate su dati logici, vale a dire, vero o falso. Ad esempio, diciamo se qualcuno ti chiama e ti chiede: "Sei a casa?" Avresti due scelte: "Sì! Sono a casa" o "No! Non lo sono". Questo sarebbe sotto 0 (falso) e 1 (vero), nella programmazione. Questo è noto come dati logici.

In Python, gli operatori logici vengono utilizzati per valutare le espressioni per ottenere alcune decisioni specifiche. Questi operatori sono molto utili per scrivere ragionevolmente qualsiasi logica. Ecco l'elenco degli operatori logici con una breve descrizione per costruire una migliore comprensione con questi operatori in Python.

Descrizione dell'operatore logico

- **Operatore and**

Se un'espressione "n" è vera e anche un'altra espressione "m" è vera, il risultato sarà vero. In ogni altro caso, il risultato sarà falso.

Questa tabella può aiutarti a capire "operatore and", in un modo migliore.

| <b>n</b> | <b>v</b> | <b>n and v</b> |
|----------|----------|----------------|
| Vero     | Vero     | Vero           |
| Vero     | Falso    | Falso          |
| Falso    | Vero     | Falso          |
| Falso    | Falso    | Falso          |

- **Operatore or**

Ciò risulterà falso, se e solo se entrambi gli operandi sono falsi. Considerando che un'espressione "n" è vera e un'altra espressione "v" è falsa, il risultato sarà vero.

Questa tabella può aiutarti a capire "operatore or", in un modo migliore.

| <b>n</b> | <b>m</b> | <b>n or m</b> |
|----------|----------|---------------|
| Vero     | Vero     | Vero          |
| Vero     | Falso    | Vero          |
| Falso    | Vero     | Vero          |
| Falso    | Falso    | Falso         |

---

### ● 5.3 Tipi di dati in Python

Un tipo di dati definisce un insieme di operazioni e valori che possono essere applicati ai valori interessati. Ad esempio, un interruttore di una lampadina può essere paragonato a un sistema informatico in quanto ha due valori diversi; Vero come acceso e Falso come spento. Poiché l'interruttore della lampadina contiene solo questi due valori, possiamo considerare le sue dimensioni come due. Ci sono solo due operazioni che possono essere eseguite con un interruttore della lampadina:

- Possiamo accenderlo

- Possiamo spegnerlo

In Python, Factors può contenere stime di vari tipi di dati. Python è quindi un linguaggio di programmazione per computer composto progressivamente. Pertanto, non è necessario caratterizzare il tipo di tutte le variabili durante la dichiarazione e la definizione. Le variabili legano l'incentivo al loro tipo.

Inoltre, Python consente ai suoi sviluppatori di controllare il tipo di variabile utilizzata nel programma. Python ci fornisce il framework "type()", che restituisce il tipo di variabile, che viene passato. Puoi considerare la guida di accompagnamento per caratterizzare i tipi di numerosi tipi di dati e per verificarne il tipo.

### **Esempio:**

```
>>> n = 69
>>> v = "Ciao Python"
>>> w = 17,5
>>> print (type (n));
>>> print ( type (v));
>>> print ( type (w));
```

### **Produzione:**

Quando digiterai il comando

```
>>> print (type (n));
```

La tua console stamperà

```
<type 'int'>
```

Quando digiterai il comando

```
>>> print (type (v));
```



La tua console stamperà

```
<type 'str'>
```

Quando digiterai il comando

```
>>> print (type (w));
```

La tua console stamperà

```
<type 'float'>
```

---

- **Tipi di dati standard**

In Python, gli identificatori possono resistere a vari tipi di qualità. Ad esempio, il nome di qualsiasi individuo deve essere messo da parte come una stringa mentre il suo "id" deve essere messo via come un numero intero.

Python ci fornisce diversi tipi di dati standard che caratterizzano la tecnica di capacità su ciascuno di essi. I tipi di dati normalmente utilizzati caratterizzati in Python sono denominati di seguito.

- Numbers
- String
- List
- Tuple
- Dictionary

Ora spiegheremo alcuni di loro con esempi di seguito.

- **Numbers**

In Python, Number, come tipo di dati, memorizza i valori numerici. Python genera un oggetto numerico ogni volta che un numero viene assegnato a una variabile o identificatore.

**Esempio:**

```
>>> n = 69
```

```
>>> v = 96
```

Nell'esempio sopra citato, n e v sono gli oggetti numerici

Quattro diversi tipi di dati numerici sono supportati da Python

- **Integer (int)**

Int firma numeri interi, come 69, 96, 22, ecc.

- **Long Integers (long)**

I numeri interi lunghi vengono utilizzati per un intervallo relativamente più elevato di valori interi, come -0x19292L, 908800L e così via.

- **Float (float)**

In Python, float viene utilizzato per memorizzare valori in virgola mobile, come 1.9, 69.1902, 262.2, ecc.

- **Complex**

Questo tipo, in Python, supporta numeri complessi, come 12.14j, 2.0 + 12.3j, ecc.

Andando avanti, Python consente di utilizzare una "l" minuscola da utilizzare con i numeri interi lunghi. Ma è necessario assicurarsi che venga sempre utilizzata una "L" maiuscola per chiarezza e comprensione di base.

Inoltre, un numero complesso è sempre costituito da una coppia ordinata, come  $n + iv$  dove  $n$  e  $v$  denotano rispettivamente le parti reale e immaginaria.

- **String**

In Python, possiamo descrivere la stringa come una sequenza di caratteri rappresentati tra virgolette. Inoltre, è possibile utilizzare virgolette singole, doppie o triple per definire una stringa.

La gestione delle stringhe è un'attività semplice, comprensibile e molto chiara poiché ci sono molte funzioni e operatori incorporati forniti da Python per eseguire questa attività. Per la gestione delle stringhe in Python, l'operatore "+" viene utilizzato per concatenare due stringhe come operazione

**"Ciao" + "signor Faheel Nasir"**

Riporterà

**"Ciao signor Faheel Nasir".**

Inoltre, l'operatore "\*" è comunemente noto come operatore di ripetizione come operazione

**"Specchio" \* 2**

Ritorna,

**"Specchio specchio"**

Puoi comprendere la gestione delle stringhe in Python, con l'aiuto del seguente esempio:

**Esempio:**

```
>>> string1 = "Ciao signor Faheel Nasir"
```

```
>>> string2 = "Come stai"
```

```
>>> print (string1 [0: 2]) #stampa i primi due caratteri usando l'operatore  
slice  
>>> print (string1 [4]) #stampa il quarto carattere della stringa  
>>> print (string1 * 2) #stampa la stringa due volte  
>>> print (string1 + string2) #stampa la concatenazione di stringa1 e  
stringa2
```

### **Produzione:**

Quando scrivi il primo comando e premi Invio, es

```
print (string1 [0:2])
```

La tua console stamperà i primi due caratteri della stringa utilizzando un operatore slice. Quindi, in questo caso, la tua console stamperà "Ciao".

Quando scrivi il secondo comando, es

```
print (string1 [4])
```

La tua console stamperà il quarto carattere della tua stringa. Quindi, in questo caso, la tua console stamperà "r".

Quando scrivi il terzo comando e premi invio, es

```
print (string1 * 2)
```

La tua console stamperà la tua prima stringa due volte. Quindi, in questo caso, la tua console stamperà "Ciao signor Faheel Nasir Ciao signor Faheel Nasir".

Quando scrivi il quarto comando e premi Invio, ad es

```
print (string1 + string2)
```

La tua console stamperà la concatenazione di string1 e string2. Quindi, in questo caso, la tua console stamperà "Ciao signor Faheel Nasir, come stai".

- **Operatori di stringa**

### **Operatore addizione "+":**

In Python, l'operatore di aggiunta "+" viene utilizzato per unire le stringhe in un programma.

### **Operatore di moltiplicazione "\*":**

In Python, l'operatore di moltiplicazione con il simbolo "\*" viene utilizzato per la generazione di più copie della stessa stringa per eseguire e chiamare una funzione.

### **Operatore di sezione "[]":**

In Python, l'operatore slice viene utilizzato per rendere disponibile una sottostringa, da una stringa specificata.

### **Operatore sezione intervallo "[:]":**

In Python, possiamo usare l'operatore range slice per eseguire una funzione e per chiamare o ottenere caratteri.

### **In Operatore "In":**

Questo è un operatore di appartenenza. Restituisce un valore in base alla presenza di una sottostringa specifica nella stringa principale.

### **Operatore rimanente "%":**

L'operatore resto, in Python, viene utilizzato per eseguire la formattazione delle stringhe.

- **Lists**

In Python, usiamo gli elenchi come usiamo gli array in C o C ++. Tuttavia, l'elenco può contenere dati di vari tipi. Gli elementi memorizzati in un elenco sono separati da una virgola "," e racchiusi tra parentesi quadre "[]"

Gli operatori di sezione "[:]" possono essere utilizzati per accedere agli elementi della lista. L'operatore di addizione o concatenazione "+" e l'operatore di ripetizione o moltiplicazione "\*" funzionano con l'elenco nello stesso modo in cui funzionano con le stringhe.

### **Esempio:**

```
>>> l = [1.5, "Hello", "Python", 7]
>>> print (l [3 :]);
>>> print (l [0:2]);
>>> print (l);
>>> print (l + l);
>>> print (l * 3);
```

### **Produzione:**

Quando scrivi il primo comando e premi Invio, es  
print (l [3:]);

La tua console stamperà, "[7]".

Quando scrivi il secondo comando e premi invio, es  
print (l [0: 2]);

La tua console stamperà "[1.5, 'Hello']".

Quando scrivi il terzo comando e premi invio, es  
stampa (l);

La tua console stamperà "[1.5, 'Hello', 'Python', 7]".

Quando scrivi il quarto comando e premi Invio, ad es  
stampa (l + l);

La tua console stamperà "[1.5, 'Hello', 'Python', 7, 1.5, 'Hello', 'Python', 7]"

Quando scrivi il quinto comando e premi Invio, es  
stampa (l \* 3);

La tua console stamperà: "[1.5, 'Hello', 'Python', 7, 1.5, 'Hello', 'Python', 7, 1.5, 'Hello', 'Python', 7]"

Python List Funzioni integrate Descrizione:

| Funzione incorporata di Python | Descrizione delle funzioni                      |
|--------------------------------|---|
| len(list):                     | Lunghezza della lista.                          |
| max(list):                     | Elemento massimo della lista.                   |
| min(list):                     | Elemento minimo della lista.                    |
| cmp(list1, list2):             | Confrontando gli elementi di entrambe le liste. |
| list(seq):                     | Sequenza all'elenco.                            |

- **Tuple**

Tuple, in Python, è identica all'elenco in diversi modi. Analogamente agli elenchi, anche le tuple possiedono la raccolta di diversi elementi di numerosi tipi di dati. I componenti della tupla sono separati con l'aiuto di una virgola "," e racchiusi tra parentesi "()".

Inoltre, il sistema non può modificare da solo la dimensione, il valore e il numero degli elementi in una tupla.

**Esempio:**

```
>>> t = ("Ciao", "Il mondo di Python", 69)
>>> print (t [1:]);
>>> print (t [0: 1]);
>>> print (t);
>>> print (t + t);
>>> print (t * 3);
>>> print (type (t))
>>> t [2] = "ciao";
```

### **Produzione:**

Quando scrivi il tuo primo comando e premi Invio, es  
print (t [1:]);

La tua console stamperà, "('Il mondo di Python', 69)".

Quando scrivi il tuo secondo comando e premi Invio, ad es  
print (t [0: 1]);

La tua console stamperà "('Ciao',)".

Quando scrivi il tuo terzo comando e premi Invio, ad es  
stampa (t);

La tua console stamperà, "('Ciao', 'Il mondo di Python', 69)".

Quando scrivi il quarto comando e premi Invio, ad es  
stampa (t + t);

La tua console stamperà, "('Ciao', 'Il mondo di Python', 69, 'Ciao', 'Il mondo di Python', 69)".

Quando scrivi il tuo quinto comando e premi invio, ad es  
stampa (t \* 3);

La tua console stamperà "('Ciao', 'Il mondo di Python', 69, 'Ciao', 'Il mondo di Python', 69, 'Ciao', 'Il mondo di Python', 69)".

Quando scrivi il tuo sesto comando e premi Invio, es



stampa (type (t))

La tua console stamperà "<type 'tuple'>".

Quando scrivi il tuo settimo comando e premi Invio, es

```
t [2] = "ciao";
```

La tua console stamperà stamperà;

Traccia a ritroso (la chiamata più recente per ultima):

File "main.py", riga 8, in <module>

```
t [2] = "Ciao";
```

Errore di tipo: l'oggetto "tuple" non supporta l'assegnazione

---

- **Capitolo 6: Espressioni regolari, dichiarazioni di espressioni, cicli in Python**

Come altri linguaggi di programmazione per computer, Python possiede anche espressioni regolari, istruzioni e cicli. Ma le espressioni e i cicli di Python sono unici e rappresentano la totalità della programmazione Python. Tutti questi metodi, funzioni, cicli e istruzioni svolgono un ruolo vitale nella creazione di un programma efficace per l'analisi dei dati e per l'esecuzione di altre attività in Python. C'è una serie di ragioni dietro le aggiunte e gli aggiornamenti di questi runner operativi nelle librerie di Python. In questo capitolo, discuteremo l'importanza e le funzionalità di questi metodi, mentre si utilizza Python, come modalità di programmazione.

---

- **6.1 Espressioni regolari di Python**

Le espressioni regolari in Python sono anche conosciute come regex. Regex funziona per analizzare il pattern in una stringa. Esistono vari numeri di funzionalità regex che possono essere ereditate, chiamate o importate per essere utilizzate. Per importare queste funzioni, possiamo usare il comando `import re`

---

- **Funzioni Regex**

- **Split**

Split è usato per dividere la nostra stringa desiderata.

- **Sub**

Usiamo il sub per sostituire le corrispondenze in una stringa.

- **Match**

In Python, Match valuta il pattern regex e lo restituisce come tipo booleano, ovvero True o False.

- **Findall**

Findall viene utilizzato in Python per ripristinare tutte le corrispondenze in una stringa.

- **Search**

La ricerca viene utilizzata per trovare tutte le corrispondenze in una stringa.

---

- **6.2 Espressioni di istruzioni di Python**

Il token '='; "il segno di uguale" viene utilizzato per rappresentare un'istruzione di assegnazione. Nella struttura della frase e nel dialetto di Python, l'istruzione di assegnazione funziona in modo diverso rispetto ad altri linguaggi e dialetti di programmazione per computer primari e questo sistema di base, contando l'idea della forma dei fattori di Python, illumina numerosi punti salienti del linguaggio Python. Se consideriamo questo compito nel linguaggio di programmazione C, sarà, diciamo,  $v = 2$ , il che significa che esiste una variabile composta denominata "v" e ha un duplicato di valore numerico 2. In questo caso, il lato destro o stima viene replicato in un'area di stoccaggio assegnata per la quale il nome della variabile sul lato sinistro è l'ubicazione emblematica. La memoria allocata alla variabile "v" è sufficientemente ampia per l'ordinamento pronunciato.

Nell'istanza più esplicita dell'attività Python, usiamo un modello molto simile,  $v = 2$ , significa "(non esclusivo) il nome v ottiene un riferimento a un diverso oggetto assegnato progressivamente di tipo" int "numerico o intero di valore significativo 2. "

In questo, discuteremo le istruzioni di base utilizzate da Python, nella routine.

### **Dichiarazione if**

In Python, l'istruzione if esegue in modo restrittivo un quadrato di codice, insieme a else ed elif, che è una compressione di else-if.

### **Dichiarazione for**

In Python, l'istruzione for sottolinea un articolo. Cattura ogni componente in un fattore vicino per essere utilizzato dal quadrato collegato.

### **Dichiarazione while**

L'istruzione While, in Python, esegue un quadrato di codice fintanto che la sua condizione è valida.

### **Dichiarazione try**

In Python, l'istruzione try consente ad alcuni casi speciali sollevati nel loro rispettato e connesso code square di essere acquisiti e gestiti eccetto i requisiti; questo allo stesso modo garantisce che il riordino del codice e alla fine il quadrato verrà eseguito costantemente prestando poca attenzione a come il quadrato esce.

### **Dichiarazione the raise**

Python ha utilizzato l'istruzione raise per sollevare un'esenzione specificata o rilanciare nuovamente molti casi speciali.

### **Dichiarazione the class**

In Python, a causa dell'istruzione class, puoi eseguire un quadrato di codice e aggiungere il suo spazio dei nomi vicino a una classe, in modo da poterlo utilizzare nella programmazione organizzata per elementi.

### **Dichiarazione def**

In Python, l'istruzione def è un'istruzione che descrive una capacità o una tecnica.

### **Dichiarazione pass**

In Python, l'istruzione `pass` è un'istruzione che viene compilata come NOP, ovvero "nessuna operazione". A causa di questa affermazione, si prevede linguisticamente di creare un quadrato di codice libero.

### **Dichiarazione `assert`**

L'istruzione `Assert` viene solitamente utilizzata durante la risoluzione dei problemi o il processo di debug per verificare le condizioni da applicare.

### **Dichiarazione `yield`**

In Python, una dichiarazione `yield` è una dichiarazione che ripristina un incentivo da un lavoro generatore. Da Python 2.5, `yield` viene inoltre considerato come amministratore. Questa forma della dichiarazione è comunemente usata per aggiornare le co-routine.

### **Dichiarazione `import`**

In Python, l'istruzione `import` è un'istruzione generalmente utilizzata per importare i moduli le cui capacità o fattori possono essere utilizzati nel presente programma. Esistono tre diversi modi di utilizzo

```
>>> Import:
```

```
>>> import <module name>
```

```
>>> [as <alias>].
```

```
>>> Print statement
```

L'istruzione `print` è stata modificata nel framework `print ()` in Python con il rilascio di Python 2.7

La spiegazione di alcune delle affermazioni più utilizzate è la seguente

- 
- **Istruzioni if-else in Python**

Di solito, qualsiasi dichiarazione di decisione in un computer è nota come selezione a due vie. La decisione viene presentata al computer, una dichiarazione condizionale e il computer può rispondere come vero o falso. In tali condizioni, se la risposta è vera, viene eseguita un'azione o un insieme di azioni. D'altra parte, se la risposta è falsa, viene eseguita un'azione o un insieme di azioni differenti.

In Python, il processo decisionale è una parte primaria dei componenti dialettali e linguistici. Come suggerisce il nome, il processo decisionale consente al codice di eseguire un blocco di codice specifico per una decisione specifica. Tuttavia, principalmente sulla convalida di una particolare condizione, le scelte vengono effettuate. Gli atti di verifica delle condizioni sono la spina dorsale del processo decisionale. In Python, possiamo eseguirli con le seguenti istruzioni.

- **Istruzione If**

L'istruzione If viene utilizzata per verificare una condizione specifica. Ad esempio, se questa condizione (gelato) è valida, la funzione procede.

- **If - else Statement**

L'istruzione If - else è in qualche modo identica all'istruzione if tranne per il fatto che fornisce anche le intuizioni e controlla la validità del codice, sia vero che falso. Per questo motivo, l'istruzione else verrà eseguita se la condizione fornita nell'istruzione if è falsa.

- **Istruzione if nested**

Le istruzioni if nested ci consentono di utilizzare l'istruzione if-else all'interno di un'istruzione if esterna.

Abbiamo imparato che le affermazioni if-else possono essere qualsiasi affermazione; può includere anche altre istruzioni if-else. Quindi, quando un'istruzione if-else include un'altra istruzione if-else, questa è nota come un'istruzione if nidificata. Non esiste un limite specifico per quanti livelli in un if-else possono essere annidati.

---

- **Rientro in Python**

Per la semplicità della programmazione Python e per ottenere chiarezza, Python non consente l'uso di contenitori per il codice a livello quadrato. In Python, il rientro viene utilizzato per pronunciare un quadrato. Nella remota possibilità che due affermazioni siano a un livello di rientro simile, a quel punto, sono entrambe il pezzo dello stesso quadrato.

A quel punto, vengono assegnati quattro spazi per indentare le istruzioni, che sono una misura comune di indentazione in Python. L'indentazione è il componente più utilizzato del linguaggio Python poiché proclama il quadrato del codice. Ciascuna delle affermazioni di un quadrato è proposta a un livello di rientro simile. Percepiremo come avvenga il vero rientro nella leadership di base e in altre cose in Python.

---

- **L'istruzione if**

Un'istruzione if viene utilizzata per testare una condizione specifica e, se la condizione è valida, esegue un codice noto come blocco if. La condizione, se l'affermazione, può essere qualsiasi articolazione sostanziale e coerente che può essere valutata come vera o falsa.

La sintassi di if else è la seguente:

### **Esempio 1:**

```
>>> numero = int(input("Inserisci il numero desiderato ="))
>>> if number% 2 == 0:
>>> print ("Il numero inserito è un numero pari")
```

### **Produzione:**

Quando eseguirai questo codice, lo schermo della tua console stamperà una riga come,

Immettere il numero desiderato =

Dopo aver inserito il numero, se il numero inserito è un numero pari, il programma produrrà un risultato,

Il numero inserito è un numero pari:

### **Esempio - 2:**



```
>>> n = int (input ("Inserisci il primo numero intero"));
>>> v = int (input ("Inserisci il secondo intero"));
>>> w = int (input ("Inserisci terzo numero intero"));
>>> if n> ve n> w:
>>> print ("n è il più grande");
>>> if v> ne v> w:
>>> print ("v è il più grande");
>>> if w> ne w> v:
>>> print ("w è il più grande");
```

### **Produzione:**

Dopo aver compilato ed eseguito questo programma, il computer lo stamperà sullo schermo della console,

Immettere il primo numero intero

Supponiamo che tu abbia inserito 69 come primo numero intero, cioè "n".

Il tuo computer ora stamperà un'altra dichiarazione sullo schermo della tua console,

Immettere il secondo numero intero

Supponiamo che tu abbia inserito 70 come secondo numero intero, cioè "v".

Ora, il tuo computer stamperà un'altra dichiarazione sullo schermo della console,

Immettere il terzo numero intero

Supponiamo che tu abbia inserito 69 come terzo numero intero, cioè "w".

Nel caso sopra menzionato, mostrerà questa dichiarazione come nostro output.

v è il più grande

---

- **L'istruzione if-else**

L'istruzione if-else fornisce un altro blocco oltre all'istruzione if, che viene eseguita nel caso falso della prima condizione. Quando la condizione è vera, viene eseguito il blocco if. In caso contrario, il blocco else verrà considerato vero.

In Python, la sintassi per l'istruzione if else è la seguente

```
>>> if condition:
>>> (blocco di dichiarazioni)
>>> Else:
>>> (Blocco di dichiarazioni else)
```

### **Esempio 1:**

```
>>> votage = int (input ("Inserisci la tua età ="))
>>> if votage >= 18:
>>> print ("Sei idoneo a votare!");
>>> else:
>>> print ("Scusa! Devi aspettare!");
```

### **Produzione:**

Quando eseguirai questo programma, il tuo sistema stamperà la seguente riga sullo schermo della tua console.

Inserisci la tua età =

Se inserisci un numero intero, che è uguale o maggiore di 18, lo schermo visualizzerà un messaggio,

Hai diritto a votare!

Se inserisci un numero intero, che è inferiore a 18, il tuo schermo stamperà questo messaggio,

Spiacente! Devi aspettare!

### **Esempio - 2:**

```
>>> numero = int (input ("Inserisci il numero desiderato:"))
>>> if numero% 2 == 0:
>>> print ("Il numero inserito è un numero pari!")
>>> else:
>>> print ("Il numero inserito è un numero dispari!")
```

### **Produzione:**

Quando esegui questo programma, il tuo schermo stamperà un messaggio,  
Inserisci il numero desiderato:

Se inserisci un numero pari, verrà stampato,

Il numero inserito è un numero pari!

Se inserisci un numero dispari, verrà stampato,

Il numero inserito è un numero dispari!

---

#### **● La dichiarazione elif**

L'istruzione elif ci aiuta a gestire più livelli di condizioni contemporaneamente. Deve avere livelli if-and-if per eseguire le attività, specificate dal programma. Funziona solo assumendo una serie di condizioni "Vere".

La sintassi dell'istruzione elif è la seguente

```
>>> if (espressione 1):
>>> (blocco di dichiarazioni);
>>> elif (espressione 2):
>>> (blocco di dichiarazioni);
>>> elif (espressione 3):
>>> (blocco di dichiarazioni);
```

```
>>> else:
>>> (blocco di dichiarazioni);
```

---

### ● Istruzione break di Python

L'istruzione break ha la sua importanza unica nella programmazione orientata ai cicli Python. Sposta il modello di esecuzione alle righe successive suddividendo il ciclo dai codici precedenti. Con una sintassi semplice, restituisce il controllo ai cicli richiesti nello stesso programma più grande. La sua sintassi è semplice.

**break;**

---

### ● Istruzione continue di Python

In Python, questa istruzione fornisce il controllo di un programma all'inizio di un ciclo. Consente a un individuo di saltare il resto dei codici e l'esecuzione torna all'inizio. Ha un ruolo importante nel saltare e nell'esecuzione in condizioni speciali.

La sintassi dell'istruzione continue è la seguente

```
>>> istruzioni del ciclo
>>> continue;
>>> (il codice da saltare);
```

#### **Esempio 1:**

```
>>> v = 0;
>>> while v!=10:
>>> print ("%d"%v);
>>> continue;
>>> v=v+1;
```

#### **Produzione:**

In questo caso, l'output sarà un ciclo infinito.

#### **Esempio - 2:**

```
>>> n=1;
>>> per n nell'intervallo (1,10):
>>> if n == 15;
>>> continue:
>>> print ("%d"%n);
```

### **Produzione:**

In questo caso, l'output sarà un elenco di numeri interi.

```
10
11
12
13
14
16
17
18
19
20
```

---

### ● Istruzione Pass di Python

In Python, l'istruzione pass è considerata come una parte non eseguibile del programma. Sembra solo giustificare la sintassi e fornisce solo un'operazione nulla. Viene generalmente utilizzato quando il codice non fa parte di un programma, ma è scritto da qualche parte al di fuori del programma. Usiamo questa dichiarazione, come sua sintassi.

### **Pass**

**Esempio:**

```
>>> for v in [1,2,3,4,5]:  
>>> if v == 4:  
>>> pass  
>>> print "(passa quando il valore è)", v  
>>> print (v);
```

**Produzione:**

```
>>>  
1 2 3 Passa quando il valore è 4  
4 5  
>>>
```

---

- **Dichiarazione di importazione in Python**

Questa istruzione è considerata la più preziosa nella programmazione Python. Rende possibile l'accesso delle funzionalità di un modulo a un altro modulo. Senza l'istruzione import, Python non è in grado di eseguire fino al segno. Usiamo questa istruzione, come sintassi dell'istruzione import.

**modulo di importazione;**

**Esempio:**

```
>>> import document;  
>>> cognome = input ("inserisci il tuo cognome:")  
>>> document.displayMsg (cognome)
```

**Produzione:**

Quando esegui questo programma, il tuo sistema stamperà un messaggio sullo schermo del tuo computer,

Inserisci il tuo cognome:

Ad esempio, se hai inserito "Nasir", verrà visualizzato:

Ciao Nasir

---

- **6.3 Cicli in Python**

La programmazione del computer è incentrata sul flusso di alcuni comandi e funzioni più e più volte. A volte, lo stesso codice deve essere ripetuto molte volte per ottenere i risultati desiderati. È una pratica molto comune nel mondo della programmazione generale. Per rendere questo più facile per i programmatori Python, ci sono molti loop che vengono utilizzati dai professionisti per risparmiare tempo e mantenere la sintassi facile da leggere e capire. Questi cicli ripetono il codice richiesto più volte con solo un piccolo blocco di codice. In Python, questi cicli sono molto necessari per costruire programmi modello predittivi e per ottenere risultati.

In Python, i loop sono molto utili per ridurre la complessità del codice. La sintassi di questi cicli è molto facile da capire ed è necessaria per mantenere il flusso del programma. Evita la ripetizione dello stesso codice e, attraverso un semplice loop, si può facilmente ripetere più volte lo stesso codice.

Ecco alcuni loop importanti in Python.

1- ciclo for

2- ciclo while

3- ciclo fai-da-te

4- Ciclo "for" di Python

---

- **For Loop**

Per quanto riguarda il ciclo for in Python, la sua sintassi è la seguente

```
>>> for iterating_variable in sequenza:
```

```
>>> {statements};
```

**Esempio:**

```
>>> v = 1;
```

```
>>> number = int (input ("Immettere il numero desiderato:"));
```

```
>>> for v in range(1,11):
```

```
>>> print ("%d X %d = %d"% (number,v,number*v));
```

**Produzione:**

Quando esegui questo programma, la tua console stamperà,

Immettere il numero desiderato

Supponiamo che tu abbia inserito "10" come numero. La tua console stamperà,

10 X 1 = 10

10 X 2 = 20

10 X 3 = 30

10 X 4 = 40

10 X 5 = 50

10 X 6 = 60

10 X 7 = 70

10 X 8 = 80



$10 \times 9 = 90$

$10 \times 10 = 100$

---

- **Nidificato per ciclo**

In Python, il ciclo for annidato riguarda l'annidamento di un "ciclo for" all'interno di un altro "ciclo for" per eseguirlo più volte.

La sintassi del ciclo for annidato è la seguente

```
>>> per iterating_variable1 in sequenza:
```

```
>>> per iterating_variable2 in sequenza:
```

```
>>> (blocco di dichiarazioni)
```

```
>>> (Altre dichiarazioni)
```

### **Esempio:**

```
>>> v = int(input("Inserisci il numero di righe desiderato"))
>>> v,w=0,0
>>> for v in range(0,n):
>>> print ()
>>> for w in range(0,i+1):
>>> print("*",end="")
```

### **Produzione:**

Quando esegui questo programma, lo schermo della tua console lo stamperà,

Immettere il numero di righe desiderato

Diciamo che hai inserito "6". Stamperà,

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

---

- **Istruzione Else con ciclo for**

L'istruzione else è una parte essenziale di molte affermazioni condizionali. Viene anche utilizzato in molti altri linguaggi di programmazione per la soddisfazione di una condizione. In Python, l'istruzione else può essere eseguita all'interno di un "ciclo for".

**Esempio 1:**

```
>>> for v in range(0,8):  
>>> print(v)  
>>> else:print("Esclusa istruzione break quindi ciclo completamente esaurito.");
```

**Produzione:**

In questo caso, verrà stampata la schermata della console,

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

Poiché non c'è interruzione, il ciclo for è completamente esaurito

**Esempio - 2:**

```
>>> for v in range(0,5):  
>>> print(v)  
>>> break;  
>>> else:print("il ciclo for è esaurito adesso!");  
>>> print("viene usata l'istruzione break quindi il ciclo viene interrotto")
```

# L'istruzione break interrompe l'esecuzione dell'istruzione else.

### **Produzione:**

In questa condizione, la tua console stamperà,

0

viene utilizzata l'istruzione break, quindi il ciclo viene interrotto

---

### ● **Python while loop**

In generale, un ciclo while è responsabile dell'abilitazione di una parte del codice da eseguire fintanto che la condizione data è vera. Di solito viene utilizzato nel caso in cui la quantità dell'iterazione non sia nota in anticipo.

La sintassi del ciclo While è la seguente.

```
>>> while espressione:
```

```
>>> (dichiarazioni);
```

L'espressione di istruzione deve essere qualsiasi espressione Python valida che si concluda con vero o falso. Il True è qualsiasi valore diverso da zero, in questo caso.

### **Esempio 1:**

```
>>> v=1;
```

```
>>> while v<=11:
```

```
>>> print(v);
```

```
>>> v=v+1;
```

### **Produzione:**

In questa condizione, la tua console ti mostrerà un elenco di tutti i numeri interi, fino a 11.

1

2

3

4

5  
6  
7  
8  
9  
10  
11

---

- **Ciclo while infinito in Python**

Se la condizione fornita nel ciclo while non diventa falsa, il ciclo while non finirà mai e il risultato sarà un ciclo while infinito. Per avere una condizione vera, utilizziamo un valore diverso da zero nel ciclo while e un valore zero per indicare la condizione falsa

**Esempio:**

```
>>> variable = 1
>>> while variable != 2:
>>> v = int(input("Inserisci il numero desiderato:"))
>>> print("Il valore inserito è% d"% (v))
```

**Produzione:**

In questo risultato sarà,  
Immettere il numero desiderato: 67  
Il valore inserito è 67  
Immettere il numero desiderato: 69  
Il valore inserito è 69  
Immettere il numero desiderato: 68  
Il valore inserito è 68  
Immettere il numero desiderato: 76 ...  
Sarà un ciclo infinito.

---

- **Else con Python while loop**

Python consente ai suoi sviluppatori di utilizzare anche il ciclo while con il ciclo while al suo interno. Esegue il quadrato else quando la condizione data nell'articolazione while risulta essere falsa. Come il ciclo for, nella remota possibilità che il ciclo while venga interrotto utilizzando l'istruzione break. A quel punto, la casella else non verrà eseguita e verrà eseguito l'annuncio presente dopo un'altra casella.

**Esempio 1:**

```
>>> v=1;
>>> while v<=4:
>>> print(v)
>>> v=v+1;
>>> else:print("Il ciclo while è interrotto");
```

**Produzione:**

In questo caso, la tua console stamperà,

```
1
2
3
4
```

Il ciclo while è interrotto

### **Esempio - 2:**

```
>>> v=1;
>>> while v<=5:
>>> print(v)
>>> v=v+1;
>>> if(v==3):
>>> break;
>>> else:print("Il ciclo while è interrotto");
```

### **Produzione:**

In questo caso, la tua console stamperà,

1

2

---

- **CAPITOLO 7: Funzioni Python e gestione dei file**

Le funzioni Python e la gestione dei file sono la parte più importante della programmazione Python. Senza utilizzare queste funzionalità, nessun programma può ottenere i risultati desiderati. Le funzioni sono codici di facile comprensione che possono essere richiamati ovunque nel corpo principale del programma.

---

- **7.1 Funzioni in Python**

Le funzioni di Python sono piccoli pacchetti di codici molto utili che possono essere chiamati per eseguire un'attività specifica. Sono utilizzati nei programmi per svolgere ruoli speciali. Fondamentalmente, sono alcune affermazioni uniche racchiuse tra {}. In Python, puoi chiamare le funzioni tutte le volte che vuoi.

---

- **Vantaggio delle funzioni Python**

Di seguito sono riportati alcuni dei principali vantaggi delle funzioni python:

- Utilizzando le funzioni, possiamo evitare la ripetizione del codice. Con una singola istruzione è possibile chiamare l'intera funzione. Risparmia molto tempo!
- Sono riutilizzabili e la loro riutilizzabilità è una caratteristica molto interessante. Possono essere chiamati più volte in un programma.
- Utilizzando queste funzioni, un programma più grande può essere suddiviso in più funzioni. Migliora l'utilizzo.



---

- **Funzioni di Python**

Ci sono così tante funzioni nel linguaggio di programmazione Python. Possono essere richiamati dal pacchetto interprete per essere utilizzati in qualsiasi programma. Senza queste funzioni, questo Python non ha alcuna attrazione per la comunità del software. Al giorno d'oggi, vengono utilizzati in tutto il mondo per eseguire tutte le principali attività di programmazione.

---

- **La funzione abs ()**

Questa funzione viene utilizzata principalmente per i valori numerici. Restituisce un valore assoluto quando inseriamo un numero intero nel nostro programma. È specificamente per ottenere valori assoluti contro un singolo argomento. Ecco alcuni esempi di numeri assoluti per comprendere il concetto.

**Esempio:**

```
# int number
>>> integer = -25
>>> print(' abs value of -25 is', abs(integer))
#float number
>>> float = -55
>>> print('abs value of -55', abs(float))
```

**Produzione:**

il valore abs di -25 è: 25

il valore abs di -55 è: 55

Questi risultati mostrano come funziona la funzione abs ().

---

- **La funzione bin ()**

Questa funzione, bin (), restituisce i risultati binari di un numero intero. L'uscita binaria ha il prefisso 0b all'inizio del valore.

### **Esempio:**

```
>>> n=20  
>>> v= bin(n)  
>>> print(v)
```

### **Produzione:**

In questo caso, la tua console stamperà,  
0b2020

---

- **La funzione bool ()**

Questa funzione fornisce un output in valore di tipo booleano utilizzando metodi di verifica della verità. È una funzione incorporata essenziale di Python. Se è presente un valore in ingresso, il risultato è vero; in caso contrario, stampa false.

### **Esempio:**

```
>>> v1=[5]  
>>> print(v1, 'is',bool(v1))  
>>> v1= No-value  
>>> print(v1, 'is',bool(v1))
```

### **Produzione:**

[5] è vero  
Nessun valore è falso

---

- **La funzione bytes ()**

In Python, la funzione bytes () è molto utile per ottenere oggetti in byte. Appartiene al comando byte-array. Per lo più, i professionisti della programmazione Python ottengono aiuto generando oggetti tramite questo comando.

### **Esempio:**

```
>>> string = "Ciao amico."
>>> array= bytes(string, 'utf-8')
>>> print(array)
```

### **Risultati:**

Ciao amico

---

- **La funzione callable ()**

La funzione interessata indaga e mostra True quando un oggetto sembra richiamabile, altrimenti mostra False. Questa funzione può far risparmiare tempo notificando all'utente la disponibilità di un oggetto con un solo comando.

### **Esempio:**

```
>>> v= 12
>>> print (callable(v))
```

### **Produzione:**

False

---

- **La funzione compile ()**

Questa funzione, compile (), lavora sul codice sorgente utilizzando i compilatori di Python e infine genera un oggetto con quel codice. Successivamente, potremmo eseguire questo codice utilizzando un'altra funzione exec () nello stesso programma.

### **Esempio:**

```
>>> yourcode_str= v=10\n w=15\n print("sum=",v+w)'
>>> yourcode=compile(yourcode_str, "sum.py",'exec')
>>> print(type(yourcode))
>>> exec(yourcode)
>>> exec(c)
```

### **Produzione:**

sum = 25

---

- **La funzione exec ()**

La funzione exec () ha un po 'di importanza in più all'interno delle funzioni integrate di Python. Esegue i programmi di python e produce l'output. Senza questa funzione, i programmi python non possono essere eseguiti.

**Esempio:**

```
>>> v = 12
>>> exec('print(v==12)')
>>> exec('print(v+4)')
```

**Produzione:**

```
True
16
```

---

- **La funzione sum ()**

In Python, quando lavoriamo con operazioni aritmetiche mentre usiamo dati numerici, la funzione Sum () diventa inevitabile. Utilizziamo questa funzione per eseguire l'aggiunta di valori disponibili nell'elenco.

**Esempio:**

```
>>> v = sum([2, 5,4 ])
>>> print(v)
>>> v = sum([4, 2, 4], 10)
>>> print(v)
```

**Produzione:**

```
11
20
```

---

- **La funzione any ()**

La funzione any () in Python fornisce il risultato o l'output in valore di tipo booleano che può essere vero o falso.

Stampa true quando è presente un valore True nell'elenco. Ma se non c'è valore vero, dà Falso. È anche una funzione molto utile per i data scientist che lavorano su progetti di big data.

**Esempio:**

```
>>> V=[4, False,9]
>>> Print(any(V))
>>> V=[]
>>> Print(any(V))
```

**Risultato:**

True

False

---

- **La funzione ascii ()**

La funzione ascii() ha un ruolo importante in Python. Il valore di output di questa funzione è sempre una stringa. Dobbiamo ricordare che non stampa altri caratteri ASCII.

**Esempio:**

```
>>> vQ = 'Buona giornata'
>>> print(ascii(vQ))
>>> wQ = 'Buona giornata'
>>> print(ascii(wQ))
>>> print("Buona giornata")
```

**Produzione:**

'Buona giornata'

"Passa una \xf6n buona giornata"

'Buona giornata'

---

- **La funzione bytearray ()**

La funzione bytearray () gioca un ruolo importante nella programmazione Python. Per creare un oggetto, questa funzione aiuta utenti o professionisti del software senza perdite di tempo.

**Esempio:**

```
>>> string1 = "Programmazione computer"
#string1 with encode 'utf-8'
>>> array1= bytearray(string, 'utf-8')
>>> print(array1)
```

**Risultato:**

```
bytearray(b'Programmazione del computer ')
```

---

- **La funzione eval ()**

La funzione eval () gioca un ruolo aggiuntivo nella programmazione Python. Questa funzione viene eseguita in un programma in esecuzione per aiutare il gestore del codice a svolgere il lavoro in velocità.

**Esempio:**

```
>>> w = 6
>>> print(eval('w+1'))
```

**Produzione:**

7

---

- **La funzione format ()**

Questa funzione format () in Python semplifica la codifica per ogni programmatore. Formattando i valori e altri dati dati, si risparmia il tempo di un programmatore.

### **Esempio:**

```
>>> d, f and b are a type
# integer
>>> print(format(515, "d"))
# float arguments
>>> print(format(515.7898, "f"))
# binary format
>>> print(format(15, "b"))
```

### **Produzione:**

```
245
363.790
35
```

---

#### ● **La funzione frozenset ()**

La funzione frozenset (), nella programmazione Python, fornisce un oggetto congelato modificabile. Questa è una funzione molto utile per la programmazione Python.

### **Esempio:**

```
>>> letter = ('j', 'k', 'l', 'm', 'p')
>>> frozset = frozenset(letter)
>>> print('Frozen set:', frozSet)
>>> print('set with no value:', frozenset())
```

### **Output:**

```
Frozen set: ({'k', 'p', 'j', 'm', 'l'})
Set with no val: frozenset()
```

---

#### ● **La funzione getattr ()**

Questa funzione gioca un ruolo molto importante nel linguaggio Python. Con l'aiuto di questa funzione, l'utente è in grado di ottenere l'attributo dell'oggetto. I programmatori di software utilizzano questa funzione per assegnare nomi agli oggetti.

**Esempio:**

```
>>> class Details:
>>> age = 25
>>> name = "faheel"
>>> detail = Details()
>>> print('age:', getattr(detail, "age"))
>>> print('age:', detail.age)
```

**Output:**

```
age: 25
age: 25
```

---

- **La funzione globals ()**

La funzione globals () consente all'utente di ottenere la tabella dei simboli globali, con tutte le informazioni di variabili e metodi. È una funzione obbligatoria per avere tutti i simboli pronti per l'uso in qualsiasi programma python.

**Esempio:**

```
Id = 72
globals()['Id'] = 72
print(' My id :', Id)
```

**Output:**

```
My id : 72
```

---

- **La funzione hasattr ()**



La funzione `Hasattr ()` si basa sui ritorni di tipo booleano, ovvero `True` e `False`.

**Esempio:**

```
>>> v = [0, False, 5]
>>> print(any(v))
>>> v = []
>>> print(any(v))
```

**Results:**

`True`

`False`

---

- **La funzione `iter ()`**

La funzione `iter ()` è comunemente usata poiché si occupa dei valori all'interno di un oggetto principalmente `list`. Stampa i valori in una lista uno per uno.

**Esempio:**

```
# elenco di numeri
>>> list = [6,7,8,9,}
>>> listIter = iter (elenco)
# scrivi "6"
>>> print (next (listIter))
# scrivi "7"
>>> print (next (listIter))
# scrivi "8"
>>> print (next (listIter))
```

**Produzione:**

6

7

8

---

- **La funzione len ()**

È una funzione molto semplice ma estremamente importante per la programmazione Python. Gli utenti oi programmatori possono misurare la lunghezza degli articoli utilizzando questa funzione.

**Esempio:**

```
>>> stringW = '... Data...'  
>>> print (len (stringW))
```

**Risultato:**

4

---

- **La funzione list ()**

La funzione list () è una delle funzioni più comunemente utilizzate per generare un elenco completo di un insieme di istruzioni fornite.

**Esempio:**

```
>>> print(list())  
#for empty list  
# string  
>>> String = 'abcde'  
>>> print(list(String))  
# tuple  
>>> tuple = (1,2,3,4,5)  
>>> print(list(Tuple))  
# list  
>>> list = [1,2,3,4,5]  
>>> print(list(list))
```

---

- **La funzione locals ()**

Locals () fornisce un risultato di tipo booleano rispetto all'input (True o False). Richiede due input e restituisce vero o falso in base al programma definito.

**Esempio:**

```
>>> def localsBeta():
>>> return locals()
>>> def localstSenior():
>>> Alpha = True
>>> return locals()
>>> print('localsNoAuthority:', localsBeta())
>>> print('localsHighAuthority:', localsAlpha())
```

**Risultato:**

```
localsBeta: {}
localsAlpha: {'present': True}
```

---

- **La funzione map ()**

La funzione map () è molto importante in quanto puoi ottenere l'elenco di un elemento elaborato con questa funzione.

**Esempio:**

```
>>> def calculateAddition(n):
>>> return n+n
>>> numbers = (1, 2, 3, 4)
>>> result = map(calculateAddition, numbers)
>>> print(result)
# converting map object to set
```

```
>>> numbersAddition = set(result)
>>> print(numbersAddition)
```

**Risultato:**

```
<map object at 0x7fb04a6bec18>
```

---

- **La funzione delattr ()**

Questa funzione è la più importante rispetto a tutte le altre funzioni. Ad ogni passaggio, uno sviluppatore o un utente deve eliminare gli attributi da una classe e mostra un errore nel chiamare lo stesso attributo.

**Esempio:**

```
>>> class jobholder:
>>> ID= 76
>>> name= "Hiziya"
>>> email= "hiziya@company.com"
>>> def getinfo(self):
>>> print( self.id, self.name, self.email)
>>> J=jobholder()
>>> e.getinfo()
>>> delattr(jobholder, 'Job Description')
>>> e.getinfo()
```

**Risultato:**

```
76 Hiziya hiziya@company.com
```

---

- **La funzione divmod ()**

Questa funzione esegue un'operazione numerica sui valori dati. Gli argomenti utilizzati da questa funzione sono valori numerici. In tutte le operazioni numeriche. In Python, questa funzione è usata frequentemente e preferita.

**Esempio:**

```
>>> V = divmod(30,5)
>>> print(v)
```

**Risultato:**

```
(6, 0)
```

---

- **La funzione enumerate ()**

La funzione enumerate () si basa sulla sequenza dei numeri di indice. Utilizzando la sequenza e l'indice dell'elemento, può generare un oggetto con valori numerici.

**Esempio:**

```
>>> W = enumerate([4,5,6])
>>> print(W)
>>> print(list(W))
```

**Output:**

```
[(0, 4), (1, 5), (2, 6)]
```

---

- **La funzione dict ()**

Questa funzione restituisce un dizionario. Questa funzione genera tre tipi di dizionario:

- Dizionario vuoto: quando non viene passato alcun argomento.
- Dizionario coppia chiave-valore identico: quando viene fornito un potenziale argomento.
- Dizionario parola chiave e valore aggiunto: quando è presente un argomento parola chiave.

**Esempio:**

```
>>> N = dict()
>>> V = dict(c=4,d=5)
```

```
>>> print(result)
>>> print(result2)
```

### **Risultato:**

```
>>> {} #empty dictionary
>>> {'c': 4, 'd':5} #dictionary with values
```

---

#### ● **La funzione filter ()**

La funzione filter () viene utilizzata per filtrare i valori fornendo due argomenti, funzioni e variabili. In caso di (nessuna) funzione, restituisce solo TRUE.

### **Esempio:**

```
>>> def filterdata(w):
>>> if w>4:
>>> return w
>> result = filter(filterdata,(1,2,7))
>>> print(list(result))
```

### **Produzione:**

```
[7]
```

---

#### ● **La funzione hash ()**

Hash () genera il valore numerico tramite l'algoritmo hash, in Python. Questi valori possono essere numeri interi utilizzati per il confronto delle chiavi del dizionario.

### **Esempio:**

```
>>> V = hash(35)
>>> W = hash(35.6)
>>> print(v)
```

```
>>> print(W)
```

### **Produzione:**

```
35
```

```
756783388388221
```

---

- **La funzione help ()**

La funzione help () è responsabile di chiamare help per assistere il processo di passaggio dell'oggetto. Attraverso parametri aggiuntivi, questa funzione condivide i dati della guida con te.

### **Esempio:**

```
Information = help()
```

```
print(Information)
```

### **Output:**

```
Help Centre!
```

---

- **La funzione min ()**

La funzione min () aiuta a ottenere gli elementi più piccoli o più elementari prendendo due argomenti come input, l'elenco degli elementi e l'elenco delle parole chiave.

### **Esempio:**

```
>>> V = min(2100,221,225)
```

```
>>> W = min(1000.25,2025.35,5625.36,10052.50)
```

```
>>> print(V)
```

```
>>> print(W)
```

### **Output:**

221

1000.25

---

- **La funzione set ()**

La funzione set () genera un oggetto utilizzando oggetti iterabili. Questa funzione della programmazione Python è considerata come la base dei programmi.

**Esempio:**

```
>>> v = set('25')
>>> w= set('python')
>>> print(v)
>>> print(w)
```

**Output:**

```
{'2', '5'}
{'y', 'o', 't', 'h', 'p', 'n'}
```

---

- **La funzione hex ()**

La funzione hex () converte l'argomento intero in un valore stringa esadecimale. Questa funzione rende la conversione facile per tutti i programmatori, sviluppatori di software, ingegneri ed esperti professionisti di Python.

**Esempio:**

```
>>> n = hex(4)
>>> v= hex(140)
>>> print(n)
>>> print(v)
```

**Output:**

```
0x2
```



0x70

---

- **La funzione id ()**

La funzione id () genera un intero identità mediante l'uso di un argomento.

**Esempio:**

```
>>> N = id("Python")
>>> V = id(1500)
>>> W= id([95,236,92,3225])
>>> print(N)
>>> print(V)
>>> print(W)
```

**Output:**

```
59696771728
66864236539
19945047867
```

---

- **La funzione setattr ()**

Questa funzione è responsabile dell'impostazione di un attributo di un oggetto. Accetta valori diversi e dopo l'esecuzione della funzione non restituisce nulla.

**Esempio:**

```
>>> RollNo = 0 #RN- roll number
>>> Name = ""
>>> def _init_(my, RollNo, Name):
>>> my.RollNo = RollNo
>>> self.Name = Name
>>> V= Student(33,"Faheel")
>>> print(V.RollNo)
```

```
>>> print(V.Name)
>>> print(V.email) product error
>>> setattr(V, 'email', 'faheelnasir@gcaill.com') # adding new attribute
>>> print(V.email)
```

### **Output:**

```
33
Faheel
faheelnasir@gcail.com
```

---

- **La funzione slice ()**

La funzione slice () permette di creare delle sotto sequenze, in modo da ottenere più elementi da una lista. Inizialmente richiede un singolo argomento, ma una seconda funzione richiede tre argomenti per procedere.

Esempio:

```
>>> V = slice(7)
>>> W = slice(0,7,3)
>>> print(V)
>>> print(W)
```

---

- **La funzione Sort ()**

Usiamo questa funzione per l'ordinamento degli elementi. L'ordinamento, dovuto a questa funzione, è in ordine crescente. Per procedere, questa funzione utilizza normalmente quattro valori.

### **Esempio:**

```
V = "python"
W = sorted(V) # stringa di ordinamento
print(W)
```

---

- **La funzione next ()**

La funzione `next ()` consente di ottenere l'elemento successivo dal gruppo specificato. Attraverso due argomenti, questa funzione produce con un unico elemento.

**Esempio:**

```
>>> V = iter([128, 16, 42])
>>> W= next(V)
>>> print(W)
>>> W = next(V)
>>> print(W)
>>> W= next(V)
>>> print(W)
# V è un Numero
# W è un oggetto
```

**Produzione:**

```
128
16
42
```

---

- **La funzione `input ()`**

La funzione `input ()` è usata specialmente per ricevere istruzioni dal programmatore, dallo sviluppatore del software o dall'utente. Dopo aver ottenuto le informazioni, converte quel valore nel formato dati richiesto dal programma.

**Esempio:**

```
>>> valore = input("Inserisci il valore desiderato:")
>>> print("Hai inserito:",value)
```

**Output:**

Si prega di inserire il valore desiderato: 22

Hai inserito: 22

---

- **La funzione int ()**

La funzione int () è progettata per ottenere numeri interi, normalmente gli utenti la usano per convertire stringhe e altre strutture di dati in valori integrali specificati.

**Esempio:**

```
>>> n = int(15) # intero
>>> v = int(15.52) # float
>>> w = int('15 ') # string
>>> print("Int val:",a,b,c)
```

**Output:**

```
Int val: 15 15 15
```

---

- **La funzione pow ()**

Usiamo la funzione pow () per calcolare la potenza di number per definirlo per alcuni risultati specifici necessari per il progetto o programma. È davvero una funzione importante eseguire molte soluzioni algebriche per i numeri.

**Esempio:**

```
>>> Positive v, Positive w (v**w)
>>> print(pow(2, 3))
>>> Negative v, Positive w (-v**w)
>>> print(pow(-2, 3))
>>> Positive v, Negative w (v** -w)
print(pow(2, -3))
>>> Negative v, Negative w (-v** -w)
print(pow(-2, -3))
```

### **Output:**

8  
8

---

- **La funzione print ()**

La funzione print () responsabile di stampare un elemento o un oggetto sullo schermo.

### **Esempio:**

```
>>> print ("Ciao! Al mondo della programmazione Python!")  
>>> v = 7  
>>> print ("v =", v)  
>>> w = v  
>>> print ('v =', v, '= w')
```

### **Output:**

Ciao! Nel mondo della programmazione Python!  
v = 7  
v = 7 = w

---

- **La funzione range ()**

La funzione range () fornisce la sequenza che inizia da 0 in generale, aumenta di 1 e si ferma su un numero specifico.

### **Esempio:**

```
>>> print(list(range(9,12)))  
>>> intervallo(inizio, fine)
```

**Output:**

[10,11]

---

- **La funzione reversed ()**

La funzione reversed () restituisce la sequenza inversa di una data sequenza.

**Esempio:**

```
>>> string = 'Hello'
>>> print(list(reversed(string)))
>>> tuple = ('H', 'e', 'l', 'l', 'o')
>>> print(list(reversed(tuple)))
>>> range = range(10, 12)
>>> print(list(reversed(range)))
>>> List = [1, 2, 7, 7, 9]
>>> print(list(reversed(List)))
```

**Output:**

['o', 'l', 'l', 'e', 'H']

---

- **La funzione round ()**

La funzione round () viene utilizzata principalmente quando sono presenti decimali nell'elenco dei numeri.

**Esempio:**

```
>>> print(round(8))
>>> print(round(10.4))
>>> print(round(6.6))
```

### **Output:**

8  
10  
7

---

- **La funzione str ()**

Str () trasforma qualsiasi valore in stringa. Questa funzione di conversione aiuta l'utente a fare le cose rapidamente.

### **Esempio:**

```
>>> str('6')
```

### **Produzione:**

```
"6"
```

---

- **La funzione tuple ()**

La funzione tuple () genera un oggetto tramite qualsiasi funzione. Questa funzione consente agli utenti di ottenere l'oggetto richiesto semplicemente scrivendo una semplice sintassi.

### **Esempio:**

```
>>> n = tuple()
>>> print('n=', n)
>>> v = tuple([2, 8, 10])
>>> print('v=', v)
>>> n = tuple('Python')
>>> print('n=',n)
>>> n = tuple({4: 'four', 5: 'five'})
>>> print('n=',n)
```

### **Output:**

```
n = ()
v= (2, 8, 10)
```

```
n= ('P', 'y', 't', 'h','o','n')
n= (4, 5)
```

---

- **La funzione type ()**

La funzione type () viene normalmente applicata per comprendere il tipo di dati su un elemento. Con tre argomenti, la funzione type fornisce un oggetto.

**Esempio:**

```
>>> V = [4, 5] #LIST
>>> print(type(V))
>>> W = {4: 'four', 5: 'five'} #Dictionary
>>> print(type(W))
>>> class Python:
>>> n = 0
>>> InstanceOfPython = Python()
>>> print(type(InstanceOfPython))
```

**Output:**

```
<class 'V'>
<class 'W'>
<class '__main__.Python'>
```

---

- **La funzione vars ()**

La funzione vars () restituisce gli attributi che appartengono al dizionario. È la funzione più comunemente usata di Python.

**Esempio:**

```
>>> class Python:
>>> def _init_(my, v = 7, w = 9):
```



```
>>> my.v = v
>>> my.w = w
>>> InstanceOfPython = Python()
>>> print(vars(InstanceOfPython))
```

**Output:**

```
{'w': 9, 'v': 7}
```

---

- **La funzione zip ()**

La funzione zip () fornisce un oggetto con lo stesso indice con diversi contenitori. Attraverso questa funzione, i risultati possono essere prodotti in una forma raggiunta o zip.

**Esempio:**

```
>>> numericalList = [4,5, 6]
>>> stringList = ['four', 'five', 'six']
>>> V = zip()
>>> VList = list(V)
>>> print(VList)
>>> V= zip(numericalList, stringList)
>>> VSet = set(result)
>>> print(VSet)
```

---

- **7.2 Gestione dei file di Python**

Python supporta la gestione dei file e consente inoltre ai client di gestire la lettura e la scrittura di qualsiasi documento, insieme a vari metodi per gestire il file o il documento disponibile. Il concetto di gestione dei file è presente in molti linguaggi di programmazione di computer diversi, tuttavia l'utilizzo della gestione dei file è confuso o prolungato in altri linguaggi. In Python, l'idea della gestione dei file è simile ma diversa. In Python, la gestione dei file è davvero semplice.

Python fornisce molteplici caratteristiche e funzioni uniche per gestire i file. Distingue altri linguaggi di programmazione per computer di alto livello sulla base dell'organizzazione strutturale della gestione e della gestione dei file. È facile apprendere e implementare il modulo di codifica in Python. Dovremmo iniziare con la lettura e la scrittura dei file. Per questo, la sintassi è menzionata di seguito.

### **open(nome file, modalità)**

---

- **Apertura di file tramite la funzione open ()**

Usiamo la funzione open () per leggere e scrivere il file. Come accennato in precedenza, ripristina un oggetto in formato file. In generale, open () funziona insieme a due contestazioni, che riconoscono la gestione dei file. La sua sintassi è menzionata di seguito.

**Object File= open(<name>, <mode>, <buffering>)**

---

- **Chiusura del file utilizzando la funzione close ()**

Dopo il completamento del programma, il programmatore deve chiudere il file utilizzando lo script di python, close ().

Protegge il file da minacce esterne e impedisce la manipolazione delle funzionalità. La sua sintassi è menzionata di seguito.

**file.close ()**

**Esempio:**

```
Filenw = open("file.txt", "c")
```

```
se filenw:
```

```
print("aperto con successo")
```

```
filenw.close()
```

---

- **Capitolo 8: Concetti OOP in Python**

I concetti di programmazione orientata agli oggetti Python stanno giocando un ruolo vitale nell'industria del software. Ha tutti i concetti della programmazione orientata agli oggetti. Esistono molti altri linguaggi della stessa famiglia di programmazione principale, ma Python si basa sui concetti OOP sin dall'inizio. Qui, un esperto di software ha la libertà di chiamare funzioni, oggetti e classi per eseguire qualsiasi attività di programmazione. Questo linguaggio è altamente consigliato per i concetti di scienza dei dati.

---

- **8.1 Concetti di base OOP**

Parliamo di alcune parti importanti di OOPs Python:

- Framework degli oggetti - Qualità e metodi in Python
- Class - Collezione di oggetti
- Method - capacità di un oggetto
- Inheritance - eredita le qualità dell'oggetto genitore

---

- **Framework degli oggetti**

Questo framework ha un concetto simile nella programmazione come nel mondo reale. Qualsiasi sostanza esistente con una certa qualità è un oggetto. In python, ci sono ovunque approcci orientati agli oggetti e tutti questi oggetti hanno alcune qualità e funzioni specifiche. Avendo una certa capacità definita, l'oggetto contiene tutte le informazioni importanti che vengono utilizzate per ricavarne una parte di informazioni complete orientate ai risultati.

---

- **Class**

La classe riguarda un gruppo di oggetti. Queste classi hanno elementi con attributi specifici. Come nella vita reale, definiamo le classi anche nel mondo della programmazione. Ad esempio, possiamo avere una classe di studenti, lavoratori, ufficiali e molto altro ancora. Tutte le classi hanno alcuni tratti simili all'interno della classe. La sintassi per creare una classe è la seguente.

```
>>> class Nome della classe:  
>>> (istruzione-1)  
>>> (istruzione-2)  
>>> (istruzione-X)
```

---

- **Method**

Il metodo non è qualcosa di simile al modo di eseguire in linguaggio Python; piuttosto, si tratta della capacità di un oggetto definito in un programma. Si basa su quanti metodi può avere un oggetto. Viene spesso utilizzato nella programmazione Python.

---

- **Inheritance**

È parte integrante del linguaggio di programmazione Python. In OOP, è simile al sistema ereditario tradizionale dell'esistenza biologica umana. L'oggetto più giovane ha tutti i tratti e i metodi. Attraverso questo framework, possiamo sviluppare classi per utilizzare le proprietà l'una dell'altra. Aiuta a ottenere risultati utilizzando un unico codice per ogni classe. Inoltre fa risparmiare tempo e semplifica la sintassi.

---

- **8.2 Class e oggetti in Python**

Una classe è un elemento presunto che contiene un numero di oggetti. È una cosa virtuale che ci dà significato quando la guardiamo in riferimento agli oggetti e alle loro proprietà. Ad esempio, supponi un edificio ospedaliero. Dispone di stanze, letti, attrezzature mediche e così via. L'edificio dell'ospedale è una classe e tutte le parti dell'edificio sono i suoi oggetti.

In quest'area dell'esercizio didattico, parleremo della creazione di classi e oggetti in Python. Allo stesso modo discuteremo come ottenere una caratteristica usando l'oggetto classe.

---

- **Creazione di classi in Python**

Python ha una sintassi semplice per la creazione di classi. Un individuo non tecnico può anche creare una classe semplicemente digitando semplici comandi. La sintassi per creare una classe è la seguente.

```
>>> class ClassName:
```

```
>>> (statement_suite)
```

Considera la seguente guida per creare una classe Employee che contiene due campi come Employee id e name.

La classe contiene anche una capacità show (), che viene utilizzata per mostrare le informazioni sul titolare del lavoro.

**Esempio:**

```
>>> titolare di un posto di classe:
```

```
>>> id = 78;
```

```
>>> nome = "hiziya"
```

```
>>> def display (self):
```

```
>>> print(self.id,self.name)
```

Qui, il `self` è utilizzato come fonte di una variabile prospettica che allude all'attuale oggetto di classe. È costantemente l'argomento principale nella definizione della capacità. Comunque sia, l'uso di `self` è discrezionale nella chiamata di capacità.

---

- **Creazione di un'istanza della classe**

Una classe dovrebbe essere istanziata nella remota possibilità che abbiamo bisogno di utilizzare le caratteristiche della classe in un'altra classe. Può essere istanziato chiamando la classe utilizzando il nome della classe.

**Esempio:**

```
>>> id number = 78;
>>> name = "Nasir"
>>> print("ID number: %d \nname: %s"%(self.id,self.name))
>>> emp = Employee()
>>> emp.display()
```

**Output:**

```
ID number: 78
Name: Nasir
```

---

- **8.3 Python Constructor**

È un tipo speciale di metodo (funzione) utilizzato per inizializzare i membri specificati nella classe.

Esistono due tipi di costruttori:

- Uno è il costruttore parametrizzato
- Il secondo è il costruttore non parametrizzato

La sua definizione viene eseguita quando creiamo un oggetto di questa classe. I costruttori verificano che ci siano risorse misurabili per l'oggetto per eseguire un'attività per l'avvio.

---

- **Creazione del costruttore in python**

In python, il metodo `__init__` ha generato il Costruttore della classe. Questo metodo viene utilizzato quando viene creata un'istanza della classe. Possiamo passare un numero di argomenti al momento della creazione dell'oggetto classe, usando la definizione `__init__`. Ogni classe dovrebbe avere un costruttore, anche se è semplicemente il costruttore predefinito.

**Esempio:**

```
>>> class Student:
```



```
>>> count = 0
>>> def __init__(self):
>>> Student.count = Student.count + 1
>>> s1=Student()
>>> s2=Student()
>>> s3=Student()
>>> print("Il numero di studenti:", Student.count)
```

### **Produzione:**

Il numero di studenti: 3

---

### **● Funzioni di classe integrate in Python**

Python ha più funzioni di classe integrate. Proviamo a capirne le funzionalità attraverso un esempio.

### **Esempio:**

```
>>> class Workers:
>>> def __init__(my,name,age):
>>> my.name = name;
>>> my.age = age
>>> W = worker("Cindy",115,22)
>>> print(getattr(W,'name'))
>>> setattr(W,"age",27)
>>> print(getattr(s,'age'))
>>> delattr(s,'age')
>>> print(s.age)
```

### **Output:**

Cindy

27

True

AttributeError: non è presente l'attributo "età" nell'oggetto Studente ".

---

- **Attributi di classe incorporati**

Una classe in Python contiene anche alcuni attributi di classe (incorporati) che forniscono informazioni sulla classe.

Ecco l'elenco degli attributi di classe incorporati:

**`__dict__`**

Serve a fornire il dizionario contenente le informazioni sullo spazio dei nomi della classe.

**`__doc__`**

Deve contenere una stringa che ha la documentazione della classe

**`__name__`**

Accede al nome della classe.

**`__module__`**

Accede al modulo in cui è definita questa classe.

**`__bases__`**

Viene utilizzato per ottenere una tupla.

**Esempio:**

```
>>> def __init__(my,name,roll number,age):
>>> my.name = name;
>>> my.rollbumber = roll number;
>>> m.age = age
>>> def display_details(my):
>>> print("Name:%s, Roll Number:%d, age:%d"%(my.name,my.roll
number))
>>> Y = Student("Jack",10,17)
>>> print(y.__doc__)
```

```
>>> print(y.__dict__)
>>> print(y.__module__)
```

### **Output:**

```
None
{'name': 'Jack', 'Roll number': 10, 'age': 17}
__main__
```

---

## ● **8.4 Ereditarietà di Python**

L'ereditarietà di Python è una caratteristica davvero unica del linguaggio di programmazione. Migliora l'usabilità del programma e dello sviluppo. In questo framework, una classe figlia può accedere alle qualità e alle funzionalità della classe genitore. La sintassi per l'ereditarietà è la seguente.

```
>>> class derived-class(base class):
>>> <class-suite>

or

>>> class derive-class(<base class 1>, <base class 2>, ..... <base class n>):
>>> <class - suite>
```

### **Esempio:**

```
>>> class Animal:
>>> def speak(self):
>>> print("Parlare di animali")
# La classe figlio Dogs eredita la classe base Animal
>>> class Dog(Animal):
>>> def bark(self):
>>> print("cane che abbaia")
>>> d = Dog()
>>> d.bark()
>>> d.speak()
```

## **Produzione:**

cane che abbaia

Parlare di animali

---

### **● Ereditarietà multilivello Python**

Questa eredità ha più livelli in Python come simili in altri linguaggi di programmazione. Questa funzionalità orientata agli oggetti è molto utile per derivare dati da una classe e per usarli in un'altra. La sintassi dell'ereditarietà multilivello è la seguente.

```
>>> class class1:
>>> <class-suite>
>>> class class2(class1):
>>> <class suite>
>>> class class3(class2):
>>> <class suite>
```

## **Esempio:**

```
>>> class Animal:
>>> def speak(self):
>>> print("Animale parlante")
# La classe figlio Dog eredita la classe base Animal
>>> class Dog(Animal):
>>> def bark(self):
>>> print("cane che abbaia")
# La classe figlio Dogchild eredita un'altra classe figlio Dog
>>> class DogChild(Dog):
>>> def eat(self):
>>> print("Mangiare il pane...")
```

```
>>> d = DogChild()
>>> d.bark()
>>> d.speak()
>>> d.eat()
```

### **Output:**

cane che abbaia

Animale parlante

Mangiare il pane...

---

## ● **Capitolo 9: Moduli, eccezioni e array in Python**

I moduli, le eccezioni e gli array di Python sono i componenti più importanti della programmazione Python orientata agli oggetti. In Python, li usiamo entrambi, di volta in volta per avere una migliore comprensione dell'uso del codice in modo molto logico. Questi metodi di programmazione sono utilizzati anche in altri linguaggi di programmazione per computer di alto livello e possiedono molti framework popolari. A causa del loro utilizzo è facile trasformare le complessità della programmazione in semplice codifica. Discutiamoli uno per uno.

---

### ● **9.1 Moduli in Python**

I moduli Python sono alcuni programmi che sono costituiti da codici di programmazione di Python. Contengono tutte le variabili, classi e funzioni del linguaggio Python. I moduli consentono agli sviluppatori di organizzare i propri codici in modo corretto e logicamente valido. I moduli possono essere importati per utilizzare la funzionalità di un modulo per un altro.

### **Esempio:**

Ecco un modulo, chiamato mod.py. Contiene una funzione “func” che ha un codice per stampare qualche messaggio sullo schermo della console.

Quindi generiamo il modulo, mod.py.

#displayMsg stampa un messaggio con il nome.

```
>>> def displayMsg(nome)
```

```
>>> print("Ciao" + nome);
```

Ora è necessario aggiungere questo modulo nel modulo principale per chiamare la funzione del metodo. displayMsg() definito nel modulo denominato file.

---

- **Caricamento del modulo in un codice python**

Per poter utilizzare la funzionalità del codice python, vengono caricati i moduli. Generalmente, Python fornisce due tipi di istruzioni come indicato di seguito.

- La dichiarazione di importazione
- L'istruzione from-import

---

- **Moduli incorporati della libreria standard di Python**

C'è un numero illimitato di moduli incorporati di Python, discuteremo alcuni dei moduli più importanti, come,

- Random
- Statistics
- Math
- Datetime
- CSV

Per importare qualsiasi modulo tra di loro, usiamo la seguente sintassi

```
>>> Import[nome_modulo]
```

Per una migliore comprensione, potresti dire, ad esempio, `Import[math]`.

---

- **Modulo casuale in Python**

In Python, questo modulo viene generalmente utilizzato per generare numeri. Usando il comando `random ()`, possiamo generare i numeri in virgola mobile. L'intervallo di questi numeri in virgola mobile è compreso tra 0,0 e 1,0.

Di seguito sono riportate alcune delle funzioni casuali più importanti utilizzate nel modulo casuale!

- **La funzione `random.randint ()`**

Viene utilizzato per generare numeri interi casuali.

- **La funzione `random.randrange ()`**

Viene utilizzato per generare gli elementi selezionati in modo casuale.

- **La funzione `random.choice ()`**

Questo modulo viene utilizzato per selezionare in modo casuale elementi da moduli non vuoti.

---

- **Il modulo Statistiche di Python**

Il modulo statico è un modulo molto utile in python. Fornisce dati numerici elaborati dopo aver eseguito funzioni statistiche su di esso. Di seguito è riportato un elenco di alcune funzioni molto importanti e più utilizzate del modulo statico.

- **La funzione `mean()`**

La funzione `mean ()` viene utilizzata per prendere la media aritmetica della lista.

**Esempio:**

```
>>> import statistics
>>> listofdata = [5, 2, 7, 4, 2, 6, 8]
>>> v= statistics.mean(listofdata)
>>> print("La media anatematica sarà:", v)
```

**Output:**

La media anatematica sarà: 4,857142857142857

- **La funzione mediana ()**

Questa funzione viene utilizzata per trovare il valore medio della tua lista.

**Esempio:**

```
>>> import statistics
>>> setofdata = [4, -5, 6, 6, 9, 4, 5, -2]
>>> print("La mediana del set di dati è:% s"
>>>% (statistics.median (setofdata)))
```

**Produzione:**

La mediana del set di dati è: 4.5

- **La funzione mode ()**

Questa funzione fornisce dati comuni dall'elenco.

**Esempio:**

```
>>> statistiche di importazione
>>> setofdata =[2, 4, 7, 7, 2, 2, 3, 6, 6, 8]
>>> print("Modalità calcolata% s"% (statistics.mode (setofdata)))
```

**Output:**

Modalità calcolata 2

- **La funzione stdev ()**



Questa funzione viene utilizzata per calcolare la deviazione standard dei dati.

**Esempio:**

```
>>> import statistics
>>> data = [7, 8, 9, 10, 11]
>>> print("La deviazione standard dei dati del campione è: % s "
>>> % (statistics.stdev(data)))
```

**Output:**

La deviazione standard dei dati del campione è: 1.5811388300841898

- **Il median\_low ()**

La funzione median\_low () viene utilizzata per restituire la mediana bassa dei dati numerici nell'elenco fornito.

**Esempio:**

```
>>> import statistics
# semplice elenco di un insieme di numeri interi
>>> setofdata1 = [4, 6, 2, 5, 7, 7]
# Stampa la mediana bassa del set di dati
>>> print ("La mediana bassa del set di dati è % s "
>>> % (statistics.median_low(setofdata1)))
```

**Output:**

La mediana bassa del set di dati è 5

- **Median\_high ()**

La funzione median\_high () viene utilizzata per calcolare la mediana alta dei dati numerici nell'elenco inserito.

### **Esempio:**

```
>>> import statistics
# elenco di set di interi
>>> setofdata = [2, 1, 7, 6, 1, 9]
>>> print("La mediana alta del set di dati è %s "
>>> % (statistics.median_high(setofdata)))
```

### **Output:**

La mediana alta del set di dati è 6

---

#### **● Il modulo Math di Python**

Il modulo Math di Python contiene le funzioni matematiche per eseguire la maggior parte dei calcoli matematici.

Ci sono anche due costanti:

#### **● Pi ( $\pi$ )**

Pi è una costante matematica ben nota ed è definita come il rapporto tra le circonferenze di un cerchio e il suo diametro. Il suo valore è o approssimativamente 3,141592653589793.

#### **● Numero di Eulero (e)**

Il numero di Eulero è la base del logaritmo naturale e il suo valore è approssimativamente 2,718281828459045.

Di seguito vengono elaborati alcuni moduli matematici più utilizzati:

#### **● La funzione `math.log10()`**

Questa funzione viene utilizzata per calcolare il logaritmo in base 10 del numero.

### **Esempio:**

```
>>> import math
```

```
>>> v=13 # small value of x
>>> print('log10(v) is :', math.log10(v))
```

**Output:**

log10(v) is : 1.1139433523068367

- **La funzione `math.sqrt()`**

Questa funzione calcola la radice del numero.

**Esempio:**

```
>>> import math
>>> n = 20
>>> v = 14
>>> w = 17,8995
>>> print('radice quadrata di 20 è ', math.sqrt(n))
>>> print('radice quadrata di 14 è ', math.sqrt(v))
>>> print('radice quadrata di 17,8995 è ', math.sqrt(w))
```

**Output:**

la radice quadrata di 20 è 4,47213595499958  
radice quadrata di 14 è 3.7416573867739413  
la radice quadrata di 17,8995 è 4,230780069916185

- **La funzione `math.expm1()`**

Questa funzione viene utilizzata per calcolare "e" elevato alla potenza di qualsiasi numero meno 1. "e" è la base del logaritmo naturale.

- **La funzione `math.cos()`**

Questa funzione viene utilizzata per calcolare il coseno di qualsiasi numero in radianti.

**Esempio:**

```
>>> import math
>>> angleInDegree = 60
>>> angleInRadian = math.radians(angleInDegree)
>>> print(Angolo dato :, angleInRadian)
>>> print('cos(x) is :', math.cos(angleInRadian))
```

**Output:**

```
>>> Angolo dato : 1.0471975511965976
>>> cos(x) is : 0.50000000000000001
```

- **La funzione `math.sin ()`**

Questa funzione viene utilizzata per calcolare il seno di qualsiasi numero, in radianti.

**Esempio:**

```
>>> importa matematica
>>> angleInDegree = 60
>>> angleInRadian = math.radians (angleInDegree)
>>> print ('Angolo dato:', angleInRadian)
>>> print ('sin (v) is:', math.sin (angleInRadian))
```

**Produzione:**

```
Angolo dato: 1.0471975511965976
sin (v) è: 0,8660254037844386
```

- **La funzione `math.tan ()`**

Questa funzione viene utilizzata per restituire la tangente di qualsiasi numero, in radianti.

**Esempio:**

```
>>> import math
>>> angleInDegree = 60
>>> angleInRadian = math.radians(angleInDegree)
>>> print(Angolo dato :, angleInRadian)
>>> print('tan(v) is :, math.tan(angleInRadian))
```

**Output:**

```
Angolo dato : 1.0471975511965976
tan(v) is : 1.7320508075688767
```

---

- **Il modulo Sys di Python**

Il modulo sys fornisce l'accesso a funzioni specifiche del sistema. Modifica il Python Runtime Environment per consentire all'utente di ottenere parametri, variabili e articoli.

- **Importazione della funzione sys**

Prima di tutto, dobbiamo importare il modulo sys nel programma prima di iniziare l'uso delle funzioni e prima di chiamare qualsiasi funzione.

- **La funzione sys.modules**

Queste funzioni vengono utilizzate per eseguire alcune attività davvero importanti sul sistema e Python Runtime Environment nella programmazione python.

- **Funzione di sys.argv:**

Questa funzione viene utilizzata per gli argomenti.

- **Funzione di sys.base\_prefix:**

Questa funzione viene utilizzata per l'avvio.

- **Funzione di sys.byteorder**

Questa funzione viene utilizzata per ottenere byterorder.

- **Funzione di sys.maxsize**

Questa funzione viene utilizzata per ottenere un numero intero di grandi dimensioni.

- **Funzione di sys.path**

Questa funzione viene utilizzata per impostare il percorso.

- **Funzione di sys.stdin**

Questa funzione viene utilizzata per ripristinare i file.

- **Funzione di sys.getrefcount**

Questa funzione viene utilizzata per ottenere il conteggio dei riferimenti di un oggetto.

- **Funzione di sys.exit**

Questa funzione viene utilizzata per uscire dal prompt dei comandi di Python.

- **Funzione dell'eseguibile sys**

Questa funzione viene utilizzata per individuare il python nel sistema.

- **sys.platform**

Questa funzione viene utilizzata per identificare la piattaforma.

- 
- **Il modulo Collection in Python**

Il modulo di raccolta gioca un ruolo importante poiché raccoglie i principali formati di dati o strutture di dati come elenco, dizionario, insieme e tupla. Questa funzione è responsabile del miglioramento della funzionalità della versione corrente di python.

È definito come un contenitore utilizzato per conservare raccolte di dati, ad esempio list.

- **La funzione di namedtuple() nel modulo di raccolta**

Questa funzione produce un oggetto tupla senza causare alcun problema di indicizzazione.

**Esempio:**

```
>>> Faheel = ('Faheel', 25,'Maschio')
>>> print(Faheel)
```

**Output:**

```
('Faheel', 25 anni, 'Maschio')
```

- **Funzione OrderedDict ()**

Questa funzione genera un oggetto dizionario con chiave che può sovrascrivere i dati all'interno.

**Esempio:**

```
>>> import collections
>>> d1=collections.OrderedDict()
>>> d1['N']=15
>>> d1['V']=20
>>> d1['W']=25
>>> d1['T']=30
>>> for k,v in d1.items():
>>> print (k,v)
```

**Output:**

N 15

V 20

W 25

T 30

- **Funzione defaultdict ()**

Questa funzione produce un oggetto simile al dizionario.

**Esempio:**

```
>>> from collections import defaultdict
>>> number = defaultdict(int)
>>> number['one'] = 1
>>> number['two'] = 2
>>> print(number['three'])
```

**Output:**

0

- **Funzione Counter ()**

Questa funzione conta gli oggetti hashable dopo aver esaminato gli elementi della lista.

**Esempio:**

```
>>> A = Counter()
>>> Vlist = [1,2,3,4,5,7,8,5,9,6,10]
>>> Counter(Vlist)
>>> Counter({1:5,2:4})
>>> Wlist = [1,2,4,7,5,1,6,7,6,9,1]
>>> c = Counter(Wlist)
```



```
>>> print(A[1])
```

**Result:**

3

- **La funzione deque ()**

Questa funzione facilita l'aggiunta e la rimozione di elementi da entrambe le estremità.

**Esempio:**

```
>>> from collections import deque
>>> list = ["n","v","m"]
>>> deq = deque(list)
>>> print(deq)
```

**Output:**

```
deque(['n', 'v', 'm'])
```

---

- **Modulo OS Python**

Il modulo OS di Python fornisce funzioni utilizzate per interagire con il sistema operativo e ottiene anche i relativi dati su di esso. Il sistema operativo rientra nei moduli di utilità standard di Python. Il modulo del sistema operativo Python ci consente di lavorare con file, documenti e directory. Alcune delle funzioni del modulo OS sono le seguenti,

- **os.name**

Questa funzione fornisce il nome del modulo del sistema operativo che importa.

Può registrare "posix", "nt", "os2", "ce", "java" e "riscos".

**Esempio:**

```
>>> importa os  
>>> print(os.name)
```

**Output:**

posix

- **os.getcwd()**

Questa funzione ripristina la directory di lavoro corrente (CWD) del file.

**Esempio:**

```
>>> importa os  
>>> print (os.getcwd())
```

**Produzione:**

C:\Users\Python\Desktop\ModuleOS

- **os.error**

Le funzioni in questo modulo definiscono gli errori a livello di sistema operativo in caso di nomi di file non validi o percorso di file non valido.

**Esempio:**

```
>>> import os  
>>> filename1 = 'PythonData.txt'  
>>> f = open(filename1, 'rU')  
>>> text = f.read()  
>>> f.close()  
>>> print('Difficult read: ' + filename1)
```

**Output:**

Difficult read: PythonData.txt

- **os.popen ()**

Questa funzione apre un file e restituisce un oggetto file che contiene la connessione con pipe.

---

- **Il modulo datetime**

Questo è un modulo importato che permette di creare oggetti data e ora. Funziona per svolgere molte funzioni relative a data e ora.

**Esempio:**

```
>>> import datetime;  
# restituisce l'oggetto datetime corrente  
>>> print (datetime.datetime.now ())
```

**Produzione:**

2018-12-18 16: 16: 45.462778

---

- **Il file CSV (Comma Separated Values)**

Questo è un semplice formato di file che organizza i dati tabulari. Viene utilizzato per memorizzare i dati in forma tabulare o foglio di calcolo che possono essere scambiati quando necessario. Questo è un modulo dati supportato da Microsoft Excel.

- **Le funzioni del modulo CSV in Python**

Questo modulo aiuta a leggere e scrivere file CSV. Prende i dati dalle colonne e li archivia per utilizzarli in futuro.

- **La funzione csv.field\_size\_limit**

Questa funzione viene utilizzata per massimizzare le dimensioni del campo.

- **La funzione csv.reader**

Questa funzione viene utilizzata per leggere informazioni o dati da un file csv.

- **La funzione csv.writer**

Questa funzione viene utilizzata per scrivere le informazioni o i dati in un file csv. Queste funzioni hanno un ruolo importante nel modulo CSV.

---

- **9.2 Le eccezioni nella programmazione Python**

Le eccezioni sono fondamentalmente le interruzioni che interrompono un programma in esecuzione. Sono errori, bug o errori in un codice. In Python, vengono gestiti in modo diverso.

- **Le eccezioni comuni in Python**

Di seguito sono riportate alcune eccezioni comuni che possono verificarsi in python. Ogni programmatore Python professionista conosce molto bene questi errori o eccezioni.

- **L'eccezione di ZeroDivisionError**

Questa eccezione si verifica quando un numero viene diviso per zero.

- **L'eccezione di NameError**

Questa eccezione si verifica quando non viene trovato un nome.

- **L'eccezione di IndentationError**

Questa eccezione si verifica quando viene fornito un rientro errato.

- **L'eccezione di IOError**

Questa eccezione si verifica quando l'operazione di input output non riesce.

- **L'eccezione di EOFError**

Questa eccezione si verifica quando viene raggiunta la fine del file e vengono ancora eseguite le operazioni.

---

- **Eccezioni non gestite**

**Esempio:**

```
>>> n = int(input("Enter a:"))
>>> v = int(input("Enter b:"))
>>> w = a/b;
>>> print("x / y =% d"% c)
>>> print("Ciao, sono un insegnante")
```

**Produzione:**

Immettere a:10

Immettere b:0

Traceback (la chiamata più recente per ultima):

File "exception-test.py", riga 3, in <module>

c = a/b;

ZeroDivisionError: divisione per zero

---

- **Il blocco finale**

Questa eccezione viene utilizzata per eseguire un codice prima dell'istruzione try. La sua sintassi è la seguente,

```
>>> try:
```

```
>>> (blocco di codice)
>>> (questo potrebbe generare un'eccezione)
>>> finally:
>>> (blocco di codice)
>>> (questo verrà sempre eseguito)
```

### **Esempio:**

```
>>> try:
>>> fileptr = open("file.txt", "r")
>>> try:
>>> fileptr.write("Hi I am good")
>>> finally:
>>> fileptr.close()
>>> print("file closed")
>>> except:
>>> print("Error")
```

### **Output:**

file closed

Error

---

#### ● **L'eccezione che si solleva in Python**

La clausola raise in python viene usata per sollevare un'eccezione. La sua sintassi è la seguente,

**Raise exception\_class,<value>**

---

#### ● **L'eccezione personalizzata in Python**

Questa eccezione consente ai programmatori di generare eccezioni che sono già state avviate con il programma.

### **Esempio:**

```
>>> class ErrorInCode(Exception):
```

```
>>> def __init__(self, data):
>>> self.data = data
>>> def __str__(self):
>>> return repr(self.data)
>>> try:
>>> raise ErrorInCode(2000)
>>> except ErrorInCode as ae:
>>> print("Received error:", ae.data)
```

**Output:**

Received error: 2000

---

- **9.3 Array Python**

In Python e in generale, array è un insieme di elementi che vengono utilizzati per lavorare su valori di dati specifici. La programmazione di livello avanzato consente agli sviluppatori di gestire più funzionalità su strutture di dati. Utilizzando un array, il codice di Python può essere semplificato. Inoltre, consente di risparmiare molto tempo.

---

- **Elemento array**

Gli elementi di matrice sono elementi di dati, che vengono memorizzati in un array.

---

- **Indice array**

L'indice dell'array specifica la posizione di un elemento in un array.

---

- **Dichiarazione di matrice**

La dichiarazione di array può essere eseguita in molti modi diversi. Dobbiamo tenere a mente questi passaggi, durante la dichiarazione di un array, in Python.

- L'indice dell'array inizia con 0.
- Gli elementi possono essere individuati con l'aiuto del rispettivo numero di indice.
- La lunghezza di un array definisce la capacità di archiviazione di tale array.

---

- **Operazioni sugli array in Python**

Alcune delle operazioni di base in un array sono le seguenti:

- **Traverse**

Questa operazione serve per stampare tutti gli elementi uno per uno.

- **Insertion**

Questa operazione viene utilizzata per aggiungere elementi in Index.

- **Deletion**

Questa operazione viene utilizzata per eliminare un elemento in index.

- **Search**

Questa operazione viene utilizzata per cercare un elemento.

- **Update**

Questa operazione viene utilizzata per aggiornare un elemento all'indice specificato.



---

- **Generazione di array**

```
>>> importazione di array *  
>>> MyarrayName = array(typecode, [initializers])
```

---

- **Accesso agli elementi dell'array**

L'accessibilità agli elementi dell'array può essere eseguita utilizzando i rispettivi indici degli elementi interessati.

**Esempio:**

```
>>> importa l'array come arr  
>>> a = arr.array('i', [1, 3, 5, 87])  
>>> print ("Primo elemento:", a [0])  
>>> print ("Secondo elemento:", a [1])  
>>> print ("Penultimo elemento:", a [-1])
```

**Output:**

```
Primo elemento: 1  
Secondo elemento: 3  
Penultimo elemento: 8
```

- Dobbiamo ricordare che gli array sono modificabili e gli elementi possono essere modificati in modo simile agli elenchi.
- Una combinazione di array rende il processo veloce e fa risparmiare tempo. L'array può ridurre la dimensione del codice.
- L'eliminazione può essere eseguita utilizzando l'istruzione del, in Python.

---

- **Lunghezza di un array**

La lunghezza di un array può essere descritta come il numero di elementi in un array. Restituisce un valore intero che è uguale al numero totale degli elementi presenti nell'array interessato. La sua sintassi è la seguente,

**len(nome\_array)**

**Esempio:**

```
>>> n = arr.array('d', [1.2, 2.2, 3.2,3,6,7.8])
>>> v = arr.array('d', [4.5,8.6])
>>> w = arr.array('d')
>>> w = n+v
>>> print("Array w =", w)
```

**Produzione:**

```
Array w = array ('d', [1.2, 2.2, 3.2, 3.6, 7.8, 4.5, 8.6])
```

**Esempio:**

```
>>> importa l'array come arr
>>> x = arr.array('i', [5, 10, 15, 20])
>>> print("Primo elemento:", x [0])
>>> print("Secondo elemento:", x [1])
>>> print("Secondo ultimo elemento:", x [-1])
```

**Produzione:**

```
Primo elemento: 5
```

```
Secondo elemento: 10
```

```
Penultimo elemento: 15
```

---

- **Capitolo 10: Librerie in Python**

Nei capitoli precedenti abbiamo studiato i concetti importanti di Python, come strutture dati, funzioni incorporate, variabili, eccezioni, metodi, cicli for e istruzioni. Nel capitolo seguente, studieremo a fondo i moduli, le librerie e i pacchetti di Python che possono essere importanti per qualsiasi progetto. Importare è il concetto più importante in Python. Quindi, utilizzando questo concetto, possiamo chiamare milioni di librerie che possono essere utilizzate in Python.

Inoltre, la programmazione Python e la scienza dei dati sono parallele tra loro. Python è un linguaggio incredibile per la scienza dei dati ed è importante per le persone che hanno bisogno di iniziare nel campo della scienza dei dati. Fornisce innumerevoli librerie e sistemi di cluster per decidere se lavorare con la scienza dei dati in modo impeccabile e molto produttivo. I diversi sistemi e librerie accompagnano un motivo particolare per l'uso e dovrebbero essere scelti in base ai prerequisiti.

---

- **10.1 Librerie generali Python**

Python è noto come "linguaggio di programmazione incluso nelle batterie". Python essenzialmente implica e accompagna varie librerie preconfezionate. In ogni caso, ci sono un certo numero di librerie differenti accessibili per lo stato tradotto, anormale. Inoltre, Python è un linguaggio di programmazione universalmente utile.

Tra i diversi elementi che vengono aggiunti alla prevalenza di Python, è degno di nota un enorme raccolta di librerie. La maggior parte delle librerie e dei bundle sono un bundle o un gruppo di molti linguaggi di programmazione, che fornisce l'accesso agli sviluppatori. Ecco alcuni dei linguaggi più importanti utilizzati in Python.

Una delle più importanti librerie Python generali è "Richieste". Intende rendere la richiesta HTTP sempre più adatta all'uomo e meno difficile. Questa libreria è autorizzata con il permesso Apache2 e scritta in Python; Requests è un vero standard utilizzato dagli sviluppatori Python per fare richieste HTTP durante l'utilizzo di Python.

Utilizzando la libreria Richieste per inviare sollecitazioni HTTP a un server, consente inoltre di includere informazioni sulla struttura, contenuto, intestazione, documenti in più parti e così via. Con la libreria, i progettisti non hanno bisogno di aggiungere una domanda all'URL o la struttura codifica fisicamente le informazioni POST.

La libreria Requests astrae le varie complessità delle richieste HTTP in un'API di base in modo che gli sviluppatori possano concentrarsi maggiormente sulla comunicazione con gli amministratori. Questa libreria, inoltre, offre supporto di autorità per Python 2.7, Python 3.4 o più e funziona perfettamente anche con PyPy.

### **Caratteristiche speciali:**

- Consente il trasferimento di record in più parti e il trasferimento di download.
- Sostanza automaticamente la decompressione districante e programmabile.
- Lo stile del browser è la conferma SSL.
- In questa libreria, le funzionalità possono essere modificate e migliorate in base ai prerequisiti.
- Questa libreria mantiene viva e fornisce il pooling supporta domini internazionali e URL.

---

- **PIL**

PIL o Python Imaging Library è una libreria Python gratuita che aggiunge una capacità di immagine alle variabili Python. In termini di base, PIL consente di controllare, aprire e risparmiare diversi record di immagini che vengono organizzati in Python. Questa libreria è stata sviluppata da Alex Clark e contributori. Pillow è un sottoinsieme della libreria PIL.

Inoltre, questa libreria offre incredibili capacità di gestione delle immagini; PIL offre una potente rappresentazione interiore e un ampio supporto per l'organizzazione dei record. La libreria Python centrale in questione ha lo scopo di offrire un rapido accesso alle informazioni immagazzinate in un paio di pixel design fondamentali.

### **Caratteristiche speciali:**

- Questa libreria è efficace nell'indagare sui sostegni utilizzando la strategia show ().
- Questa libreria è ideale per le applicazioni di gestione di gruppo.
- Nelle variabili di libreria in questione e analizza una vasta gamma di progetti di documenti illustrati.
- PIL offre inoltre interfacce BitmapImage, PhotoImage e Windows DIB.
- Questa libreria supporta le modifiche discrezionali relative, le trasformazioni dello spazio di ombreggiatura, la separazione con molte parti di convoluzione implicite, il ridimensionamento e la rotazione delle immagini e le attività puntuali.
- In questa libreria, la tecnica dell'istogramma è consentita per estrarre alcune misurazioni da un'immagine, può essere utilizzata per aggiornamenti differenziati programmati e indagini fattuali a livello mondiale.

---

### **● Scrapy**

Scrapy è anche una struttura Python gratuita e open source ampiamente utilizzata per il web e vari incarichi diversi, tra cui test meccanizzati e mining di informazioni. Inizialmente, Scrapy è stato sviluppato per lo scrapping basato sul Web, ma è stato aggiornato per soddisfare diversi scopi nel corso del suo corso. Questa libreria offre anche una strategia di stato rapida e anormale per i siti striscianti e per separare le informazioni organizzate dai siti web.

Questa libreria è scritta in Python. Scrapy viene aggirato su errori e bug che sono essenzialmente crawler indipendenti, che vengono forniti con troppe linee guida. Conforme allo standard DRY, Scrapy semplifica l'assemblaggio e la scalabilità di innegabili imprese di scorrimento del web.

### **Caratteristiche speciali:**

- Scrapy è facile da comporre un bot per strisciare un sito e concentrare le informazioni.
  - Segue la regola DRY.
  - Questa libreria Python offre una shell per il web che consente agli ingegneri di testare la condotta di un sito web.
  - Questa libreria supporta l'invio di informazioni graffiate utilizzando la linea di direzione
- 

- **Tkinter**

Quando lavoriamo con Tkinter, Python offre un percorso semplice e veloce per lo sviluppo di applicazioni GUI. Questa libreria è considerata la libreria GUI standard per la programmazione Python. Offre un'incredibile interfaccia situata negli oggetti per la casella degli strumenti della GUI di Tk. Dobbiamo ricordare che lo sviluppo di un'applicazione GUI durante l'utilizzo di Tkinter è semplice. Devi solo perseguire questi progressi di base:

### **Caratteristiche speciali:**

- Questa libreria include una serie di gadget che aiutano a definire le strategie del dirigente.
  - Fornisce facilità durante la creazione di applicazioni GUI.
  - Tkinter supporta una potente interfaccia basata sugli oggetti.
- 

- **Six**



Dobbiamo ammettere il fatto che sei è la più semplice delle librerie Python. È una straordinaria libreria Python che ha lo scopo di appianare i contrasti tra le forme Python 2 e Python 3.

Six è stato sviluppato per supportare basi di codice che possono funzionare sia su Python 2 che su Python 3 senza alcun requisito di regolazione.

Le Sei librerie erano considerate super semplici da usare perché offerte come un documento Python solitario. Di conseguenza, Six è assurdamamente semplice duplicare una libreria in un'impresa Python. Il nome Six riflette Python 2 x Python 3.

### **Caratteristiche speciali:**

- Six è semplice da utilizzare le capacità per rendere il codice Python perfetto sia con Python 2 che con Python 3.
- Questa libreria supporta ogni versione di Python, a partire da Python 2.6.
- È troppo facile usare Six contenuto in un documento Python solitario.

---

### **● Pygame**

Pygame è una libreria Python open-source e gratuita pensata per il miglioramento delle applicazioni basate su video e audio in Python. In particolare supporta iniziative di gioco bidimensionali. Quindi, viene utilizzato principalmente da sviluppatori di giochi Python nuovi ed esperti.

In un compilatore o IDE Python, Pygame utilizza la libreria SDL, ovvero Simple DirectMedia Layer. Come la libreria SDL, la libreria Pygame è estremamente comoda e offre aiuto per un ampio numero di fasi e framework di lavoro.

Pygame è concepibile per scortare applicazioni create utilizzando Pygame su gadget basati su Android, come telefoni cellulari e tablet. Per molte ragioni, dovresti preferire pgs4a, che è un sottoinsieme Pygame per Android.

### **Caratteristiche speciali:**

- Questa libreria non richiede OpenGL.
- Ti aiuta in un modo semplice a utilizzare una CPU multicentrica.
- Durante l'utilizzo di questa libreria, non è richiesta alcuna GUI per utilizzare ogni singola capacità accessibile.
- Questa libreria fornisce supporto per un'ampia gamma di fasi e framework di lavoro.
- Questa libreria è semplice da utilizzare.
- Pygame utilizza il codice assembly e il codice C avanzato per aggiornare le capacità centrali.

---

### ● **Bokeh**

Bokeh è una libreria di rappresentazione istintiva per il linguaggio di programmazione Python; Garantisce l'immaginazione dei dati in un corso straordinario e critico all'interno dei programmi web contemporanei. La libreria di rappresentazione dei dati incoraggia la creazione di dashboard, applicazioni di dati e grafici dettagliati.

Nonostante offra un breve e delizioso miglioramento di piani versatili, la libreria Bokeh estende in modo simile la sua capacità con la migliore conoscenza su fuoriuscite o enormi set di dati e database.

### **Caratteristiche speciali:**

- Durante l'utilizzo di questa libreria, è possibile creare facilmente trame autentiche con intestazioni chiare senza alcuna complessità.
- I ritratti di bokeh possono essere introdotti con successo in due dei più standard. Framework Python, Django e Flask.
- Fornisce la capacità di effettuare riconoscimenti di dati sorprendenti e naturali in più dialetti del linguaggio di programmazione per computer.

---

- **Asyncio**

Questa libreria viene utilizzata per comporre codice simultaneo utilizzando l'asincronia e anticipa la grammatica da parte degli sviluppatori. Nella parte più ampia del programma, la libreria asyncio è perfetta per codice sorgente di sistema organizzato di alto livello e legato a IO.

Asyncio è stato utilizzato per la differenza strutturale di framework non concorrenti Python che offrono associazione di database con le librerie, linee di impresa circolate, sistemi d'élite e server web e molto altro ancora. La libreria in questione accompagna varie API di alto e basso livello.

### **Caratteristiche speciali:**

- Questa libreria viene utilizzata per l'implementazione dei protocolli utilizzando il trasporto.
- Durante l'utilizzo di questa libreria, troverai i codici Python facili e diretti.
- Asyncio aiuta nella generazione di molti loop diversi.

## **10.2 Librerie Python per Data Science**

Le librerie Python stanno diventando sempre più accessibili e utili giorno dopo giorno. Poiché sono un insieme di linguaggi open source, ci sono milioni di data scientist che stanno arricchendo Python con strumenti e librerie attraverso la codifica avanzata. Ora, ci sono pacchetti e librerie estremamente avanzati che i data scientist utilizzano per più attività di analisi dei dati. Di seguito viene fornita una breve descrizione di alcune delle migliori librerie di Python Data Science

---

- **Numpy**

NumPy è una libreria Python Data Science molto importante, implicita per la registrazione logica. Fornisce supporto per uno straordinario oggetto da esposizione N-dimensionale e capacità di trasmissione.

Inoltre, questa libreria offre modifiche di Fourier, capacità di numeri arbitrari e dispositivi per il coordinamento di codice C e C ++ e FORTRAN. Disporre di informazioni di lavoro su NumPy è necessario per gli sviluppatori full-stack associati a iniziative di intelligenza artificiale che utilizzano Python.

Numpy è il pacchetto fondamentale e perfetto per lavorare con le informazioni in Python. Poiché diversi pacchetti per l'indagine delle informazioni si basano su Numpy e sul pacchetto sci-kit-learn, che viene utilizzato per assemblare applicazioni di intelligenza artificiale e funziona perfettamente anche con Numpy.

Inizialmente, Numpy fornisce la fenomenale serie di oggetti, ammassi n-dimensionali. In un oggetto "ndarray", altrimenti noto come "esposizione", è possibile memorizzare varie cose con un tipo di informazione simile. Gli uffici intorno all'oggetto della mostra rendono Numpy così vantaggioso per l'esecuzione di calcoli matematici e per il controllo delle informazioni.

**Caratteristiche speciali:**

- Numpy è una libreria molto interattiva ed è anche facile.
  - Durante l'utilizzo di Numpy, i problemi matematici vengono risolti con facilità.
- 

- **Panda**

In Python, utilizziamo tabelle bidimensionali per analizzare i dati come in SQL o Excel. Inizialmente, Python non aveva questa funzionalità. Ma poi è stato introdotto Panda. Senza dubbio, Pandas è l'"SQL di Python". In breve, Pandas è la libreria che può aiutarci a gestire tabelle di dati bidimensionali in Python. In molti casi, è simile a SQL, però.

Inoltre, Pandas si basa sul bundle NumPy, il che significa che una grande quantità della struttura di NumPy viene utilizzata o duplicata per sviluppare Pandas. Le informazioni nei panda vengono spesso utilizzate per rafforzare l'esame fattuale in SciPy, le capacità di tracciamento da Matplotlib e i calcoli di apprendimento automatico in Scikit-learn.

Inoltre, i notebook Jupyter offrono una situazione decente per utilizzare i panda per eseguire indagini e dimostrazioni di informazioni, ma allo stesso modo i panda possono essere utilizzati semplicemente negli strumenti di contenuto.

Jupyter Notebooks consente a uno sviluppatore di eseguire il codice in una cella specifica invece di eseguire l'intero record. Ciò consente di risparmiare un sacco di tempo quando si lavora con enormi set di dati e modifiche complesse. Anche Scratchpad offre un metodo semplice per immaginare DataFrame e grafici dei panda. Infatti.

Pandas è noto soprattutto per fornire informazioni in Python. Questa è una biblioteca eccezionale per l'esame delle informazioni, in contrasto con altri dialetti espliciti dell'area come R. Usando Pandas; è più semplice gestire le informazioni mancanti, rafforza il lavoro con informazioni archiviate in modo contrastante e raccolte da numerose risorse diverse e supporta la disposizione programmata delle informazioni.

Inoltre, fornisce ai dispositivi l'esame delle informazioni e le strutture delle informazioni come il consolidamento, lo stampaggio o il taglio di set di dati, ed è inoltre eccezionalmente praticabile nel lavorare con le informazioni identificate con la disposizione del tempo fornendo apparati sostanziosi per impilare informazioni da Excel, documenti di livello e database.

L'utilizzo della libreria Pandas rende più semplice e istintivo per i programmatori lavorare con dati denominati o social. Offre inoltre strutture di dati espressive, rapide e adattabili. I panda sono fatti per riempire come la struttura di livello elevato essenziale chiusa per eseguire l'esame dei dati, che viene utilizzata durante l'utilizzo di Python come modalità di programmazione.

Una delle caratteristiche più importanti e dominanti di Pandas è interpretare complesse attività di dati utilizzando un paio di direzioni trascurabili. Inoltre, la libreria di intelligenza artificiale non ha carenza di lavoro nelle tecniche per consolidare, separare e raccogliere dati. Inoltre Evidenzia l'utilità della disposizione del tempo.

### **Caratteristiche speciali:**

- Durante l'utilizzo di Panda, le operazioni di tipo personalizzato possono essere completate in modo significativo.

- Con l'uso di Panda, la manipolazione dei dati diventa più semplice e facile.
  - Quando Pandas viene utilizzato con altre librerie e strumenti Python, fornisce risultati eccellenti.
- 

- **Il Matplotlib**

Matplotlib è una libreria di plottaggio bidimensionale con straordinari moduli di rappresentazione per Python. È equipaggiato per fornire figure di prim'ordine in varie organizzazioni di versioni stampate e condizioni intelligenti trasversali. Oltre ad essere utilizzato nella shell Python, nei contenuti Python e nella shell IPython, Matplotlib può essere utilizzato anche in:

- Jupyter Notebook
- Server di applicazioni Web
- Cassette degli strumenti della GUI

Come detto dal sito ufficiale di Matplotlib, questa libreria Python cerca di "rendere le cose semplici semplici e le cose difficili concepibili". La libreria Python per il plottaggio 2D consente di produrre grafici a barre, diagrammi degli errori, istogrammi, grafici, grafici a dispersione e così via con meno righe di codice.

Forse, il miglior vantaggio della rappresentazione è che ci consente l'accesso visivo a enormi quantità di informazioni in immagini effettivamente assorbibili. Matplotlib comprende alcuni grafici come linea, barra e istogramma disperso e così via.

Matplotlib è una libreria di grafici matematici in Python. È una libreria che è, per la maggior parte, utilizzata per la rappresentazione delle informazioni, inclusi grafici 3D, istogrammi, grafici, grafici a dispersione, grafici a barre e spettri di potenza con luci luminose per lo zoom e la ricerca dell'oro in vari modelli di copie stampate. Supporta praticamente tutte le fasi, ad esempio Windows, Mac e Linux. Anche questa libreria si integra come un ampliamento per la libreria NumPy. Matplotlib ha un modulo pyplot che viene utilizzato nelle rappresentazioni, che è spesso in contrasto con MATLAB.

Queste librerie sono le migliori per i dilettanti per iniziare la scienza dell'informazione utilizzando il linguaggio di programmazione Python. Ci sono numerose altre librerie Python accessibili, ad esempio, NLTK per la preparazione del linguaggio standard, Pattern per il web mining, Theano per l'apprendimento profondo, IPython, Scrappy per il web scratching, Mlpy, Stats models, e il cielo è il limite da lì.

### **Caratteristiche speciali:**

- Ha proprietà utili, proprietà dei caratteri, stili di linea, ecc. Attraverso un'interfaccia orientata agli oggetti.
- L'interfaccia MATLAB supporta il semplice tracciamento dei dati.
- Ha il supporto dell'asse x / y secondario per rappresentare due dimensioni.
- Questa libreria supporta più sistemi operativi.

---

- **Scikit-Learn**



Scikit-learn offre un ambito di calcoli di apprendimento amministrato e da solo per mezzo di un'interfaccia prevedibile in Python. È autorizzato con un permesso BSD riorganizzato indulgente ed è distribuito in numerose disseminazioni di Linux, consentendo l'uso scolastico e aziendale. La libreria è basata su SciPy (Scientific Python) che deve essere introdotto prima di poter utilizzare sci-kit-learn.

Ci sono alcune librerie Python che danno una forte esecuzione dell'ambito dei calcoli di machine learning. Eccezionale tra gli altri noti è Scikit-Learn, un pacchetto che offre adattamenti abili di innumerevoli calcoli di base. Scikit-Learn è descritto da un'API perfetta, uniforme e semplificata, proprio come una documentazione online estremamente utile e completa. Un vantaggio di questa coerenza è che una volta compreso l'uso fondamentale e la struttura linguistica di Scikit-Learn per un tipo di modello, passare a un altro modello o calcolo è estremamente diretto.

Indubbiamente le cose più fantasiose in Python sono Machine Learning e Prescient Investigation. Inoltre, la migliore libreria per questo è Scikit-Learn, che essenzialmente si caratterizza come "Machine Learning in Python". Scikit-Learn ha alcune tecniche, che coprono fondamentalmente tutto ciò che potresti richiedere in un paio di lunghi periodi della tua professione dell'informazione: strategie di ricaduta, strategie di caratterizzazione e raggruppamento, proprio come l'approvazione del modello e la determinazione del modello.

Questa libreria prevalente viene utilizzata per l'intelligenza artificiale nella scienza dell'informazione con un ordine diverso, ricadute e calcoli di raggruppamento, che offre macchine vettoriali di aiuto, Bayes innocenti, aumento dell'angolo e ricaduta sensibile. SciKit è concepito per interagire con SciPy e NumPy.

### **Caratteristiche speciali:**

- Capacità di estrarre caratteristiche da immagini e testo
  - Può essere riutilizzato in diversi contesti
- 

- **Scipy**

Ci sono scipy library e scipy stack. La stragrande maggioranza delle librerie e dei bundle sono una parte dello stack Scipy (ovvero per l'elaborazione logica in Python). E una di queste parti è semplicemente la libreria Scipy, che fornisce risposte competenti per le pianificazioni numeriche (le cose matematiche dietro i modelli AI). Questi sono incorporazione, introduzione, miglioramento e così via. Scipy fornisce al centro strategie scientifiche per realizzare le forme di IA imprevedibili in Scikit-learn.

È una libreria open-source utilizzata per registrare diversi moduli, ad esempio, preparazione di immagini, unione, inserimento, capacità uniche, miglioramenti, matematica basata su variabili dirette, trasformata di Fourier, raggruppamento e numerose imprese diverse. Questa libreria viene utilizzata con NumPy per eseguire un calcolo numerico efficiente.

### **Caratteristiche speciali:**

- Gestisce comodamente le operazioni matematiche

- Fornisce routine numeriche efficaci ed efficienti, come l'integrazione e l'ottimizzazione numerica, utilizzando sottomoduli
  - Supporta l'elaborazione del segnale
- 

- **TensorFlow**

Chiunque sia impegnato con attività di apprendimento automatico di intelligenza artificiale utilizzando Python deve, in ogni caso, conoscere TensorFlow. Creata da Google, è una libreria matematica rappresentativa open source per il calcolo numerico che utilizza i diagrammi del flusso di informazioni. Le attività scientifiche in un normale diagramma del flusso di informazioni di TensorFlow vengono lette dagli hub del grafico. I bordi del grafico, poi di nuovo, parlano alle informazioni multidimensionali esibite, alias Tensors, che fluiscono tra gli hub del diagramma.

TensorFlow sfoggia un design adattabile. Consente agli ingegneri Python di trasmettere il calcolo a una o più CPU o GPU in un'area di lavoro, un telefono cellulare o un server senza la necessità di rivedere il codice. Tutte le librerie realizzate in TensorFlow sono scritte in C e C ++. Gli elementi di Google ampiamente utilizzati come Google Foto e Google Voice Search sono costruiti utilizzando TensorFlow. La libreria ha un front-end contorto per Python. Il codice Python verrà accumulato e, successivamente, eseguito sul motore di esecuzione trasmesso da TensorFlow.

### **Caratteristiche speciali:**

- Consente di preparare vari sistemi neurali e numerose GPU, rendendo i modelli eccezionalmente produttivi per framework di

enorme scala

- Facilmente addestrabile su CPU e GPU per il calcolo diffuso
- Flessibilità nella sua operabilità, il che significa che TensorFlow offre la scelta di estrarre le parti di cui hai bisogno e lasciare ciò che non ti serve
- Ottimo livello di supporto di rete e designer
- A differenza di altre librerie Python della scienza dell'informazione, TensorFlow migliora il modo in cui immaginare ogni singolo pezzo del diagramma.

---

- **Keras**

È riconosciuta come una delle librerie Python AI (Algorithm) più interessanti; Keras offre uno strumento più semplice per comunicare i sistemi neurali. Dispone inoltre di straordinarie utilità per accumulare modelli, preparare set di dati, immaginare grafici e molto altro ancora. Scritto in Python, Keras può continuare a funzionare su CNTK, TensorFlow e Theano. La libreria Python AI è stata creata con un focus essenziale sul consentire una rapida sperimentazione. Tutti i modelli Keras sono compatti. In contrasto con altre librerie AI di Python, Keras è moderato. Ciò è dovuto al modo in cui crea un diagramma computazionale utilizzando prima il framework di backend e, successivamente, utilizza l'equivalente per eseguire le attività. Keras è estremamente espressivo e adattabile alla ricerca creativa.

### **Caratteristiche speciali:**

- Essendo totalmente basato su Python, è più semplice risolvere i problemi e indagare

- Modulare in natura.
  - I modelli del sistema neurale possono essere uniti per sviluppare modelli sempre più complessi
  - Funziona facilmente su CPU e GPU
  - Supporta praticamente tutti i modelli di un sistema neurale, inclusi convoluzionale, inseribile, completamente associato, pooling e ripetitivo.
- 

- **Seaborn**

Fondamentalmente una libreria di percezione delle informazioni per Python, Seaborn, si basa sulla libreria Matplotlib. Inoltre, è saldamente integrato con le strutture informative di Pandas. La libreria di percezione delle informazioni di Python offre un'interfaccia di stato anormale per disegnare grafici fattuali altrettanto interessanti.

Il punto principale di Seaborn è rendere la rappresentazione un elemento fondamentale per indagare e ottenere informazioni. Le sue capacità di tracciatura organizzate per set di dati funzionano su mostre e bordi informativi contenenti interi set di dati. La libreria è perfetta per controllare le connessioni tra numerosi fattori. Seaborn all'interno fa tutta la mappatura semantica significativa e la raccolta misurabile per la creazione di trame educative. La libreria di rappresentazione delle informazioni di Python dispone inoltre di dispositivi per la selezione tra le tavolozze di ombreggiatura che guidano nello scoprire i progetti in un set di dati.

### **Caratteristiche speciali:**

- Stima automatica così come il tracciamento di modelli di ricaduta diretta.

- Prospettive confortevoli sulla struttura generale di set di dati complessi.
- Facilita la costruzione di rappresentazioni complesse utilizzando deliberazioni di stato anormali per organizzare matrici multi-trama.
- Opzioni per la rappresentazione di disseminazioni bivariate o univariate.
- Supporto specializzato per l'utilizzo di fattori chiari.

---

- **Natural Language Toolkit (NLTK)**

Un prezioso per la preparazione del linguaggio comune e le iniziative di riconoscimento del design e che può essere utilizzato per creare modelli intellettuali, tokenizzazione, etichettatura, pensiero e diversi compiti utili alle applicazioni di intelligenza artificiale.

**Caratteristiche speciali:**

- Viene fornito con una struttura linguistica tagger
- Supporta la valutazione lessicale

## **Capitolo 11: Il futuro di Python**

### **11.1 Opportunità di carriera associate a Python**

In questo potente mondo di oggi in cui tutto cambia rapidamente, la prevalenza di Python sembra non fermarsi mai. Oggi, la certificazione Python è molto popolare. Ha molte librerie che aiutano l'indagine, il controllo e la rappresentazione dei dati. In questo modo, è diventato il linguaggio più preferito e visto come la "grande cosa che segue" e una "necessità assoluta" per i professionisti.

Con una vasta gamma di linguaggi di programmazione accessibili, Python ha superato enormemente i diversi dialetti. Le aperture di vocazione relative a Python si sono ulteriormente sviluppate in quanto la sua fama è aumentata del 40%. Numerose organizzazioni IT stanno cercando più candidati con esperienza e attitudini nei linguaggi di programmazione Python. Ciò ha mostrato il migliore ambito di vocazione per gli ingegneri del software Python il prima possibile.

## **11.2 Crescente popolarità di Python**

Python sta dominando il mondo della tecnologia moderna e, grazie alla sua unicità, ha lasciato altri linguaggi come C ++, Java, ecc. Molto indietro. Contiene caratteristiche dinamiche e versatili grazie alle quali è altamente utilizzato in tutto il mondo. Python, con la sua grande utilità, promette un futuro molto luminoso. Python ha pensato a 25 anni di continue modifiche con versioni migliorate e aggiornate in modo da poter fungere da linguaggio di programmazione più veloce e affidabile. Python offre la migliore qualità, ecco perché attira l'attenzione di ogni sviluppatore. È noto che quasi 126.424 siti Web sono stati realizzati utilizzando Python. Una pletora di associazioni decisionali ha sviluppato applicazioni utilizzando Python. Questo spiega che Python è il linguaggio di oggi e anche del futuro.

La crescita di Python si ottiene con un enorme sforzo. Il duro lavoro continuo svolto con l'aggiornamento di questo linguaggio ha portato risultati fruttuosi e ha reso giustizia a tale sforzo. Python ha un ruolo primario in condizioni future come ML e AI. Le funzionalità di Python continueranno ad aumentare nelle prossime versioni e diventerà un software più affidabile e veloce che alla fine diventerà sicuramente una domanda per il futuro.

---

- **Profili degli sviluppatori Python**

Il gruppo di persone Python è vario quanto la lingua e le sue applicazioni. I clienti Python hanno un'età molto ampia, ma la maggior parte degli intervistati ha 20 anni e un quarto ha 30 anni. Sorprendentemente, quasi un quinto dei clienti Python ha meno di 20 anni. Nella remota possibilità che mettiamo a confronto le fasce di età dei designer Python e le estensioni di età dello sviluppatore generale riconosciute da Stack Overflow nella sua recensione più recente, l'appropriazione sembra essere comparativa, con la panoramica di Python che dettaglia un po' più di ingegneri di età inferiore ai 18 anni. Questo può essere chiarito dal modo in cui numerosi sostituti usano Python nelle scuole e nei college, ed è una tipica prima lingua.

La maggior parte degli intervistati lavora tutto il giorno come progettisti e uno su cinque ingegneri duplica come investigatore dell'informazione, disegnatore o responsabile di gruppo. Confrontando le nostre informazioni con la panoramica di Stack Overflow, possiamo vedere che i designer Python sono più stravaganti degli ingegneri in generale per essere utilizzati a tempo pieno (52% degli ingegneri Python contro il 74% a parità di condizioni) e destinati ad agire naturalmente utilizzati o specialisti (13 % di ingegneri Python contro il 9,7% a parità di condizioni).



Gli intervistati alla revisione di Python riportano un ampio ambito di esperienza; Il 22% ha meno di quello che un anno di esperienza, un tasso simile ha oltre 11 anni di esperienza e c'è un trasporto regolare nel centro. Ciò dimostra che esiste una discreta armonia tra i nuovi arrivati e gli ingegneri esperti, rendendo Python un linguaggio gestibile.

È importante che Python abbia un livello molto più alto di nuovi arrivati rispetto a dialetti diversi in generale. Secondo le informazioni di Stack Overflow, il 30% degli ingegneri ha meno di due anni di esperienza da esperto, in contrasto con il 41% nella revisione dei designer di Python.

Enormi incontri di designer non hanno precedenti in Python. Nella nostra panoramica, il 56% degli ingegneri Python afferma di provare liberamente le imprese e il 40% lavora in un gruppo di solito da due a sette persone. Circa una gran parte degli intervistati si dedica a un'impresa principale con una serie di attività collaterali, mentre circa un quarto si concentra su ciascuna attività a turno.

---

- **Utilizzo generale di Python**

Circa quattro designer Python su cinque affermano che è il loro linguaggio fondamentale, un'espansione del quattro percento dal 75% nella panoramica 2016 di JetBrains.

Diverse ricerche dimostrano la quantità di ingegneri Python, proprio come la porzione di designer che utilizzano Python come linguaggio principale, si sta sviluppando costantemente passo dopo passo. Nella recensione di Stack Overflow, la fama di Python è passata dal 32% a metà 2017 al 38,8% verso la fine dell'anno. Questo può essere chiarito dalla notorietà in rapido sviluppo di Python per la scienza dell'informazione, poiché questa sezione di client si sta sviluppando molto più rapidamente delle altre.

Python è regolarmente unito a diversi dialetti:

La metà di tutti gli ingegneri che utilizzano Python come linguaggio principale utilizza anche JavaScript. JavaScript è utilizzato dal 79% degli ingegneri web, ma solo dal 39% di quelli coinvolti nell'indagine delle informazioni o nell'intelligenza artificiale.

La suddivisione è in qualche modo unica per le persone che usano Python come linguaggio ausiliario, poiché meno di loro usano JavaScript (46%) e molti di più usano C / C ++ (42%), Java (41%) e C # (24% ).

---

- **Alcune importanti aziende che utilizzano Python come Data Science**

- **Google**

Google, considerato il più grande gigante IT, ha supportato Python sin dall'inizio. Google utilizza Python nel proprio crawler web.

- **Facebook**

Gli ingegneri di produzione di Facebook sono molto entusiasti di utilizzare Python nel reparto di ingegneria di produzione.

- **Instagram**

Instagram engineering team ha rivelato nel 2016 che il [La più grande distribuzione al mondo del framework web Django guidato da loro è completamente scritta in Python.](#)

- **Netflix**

Netflix utilizza Python in un modo molto simile a Spotify, a seconda della lingua per alimentare la sua analisi dei dati sul lato server.

- **Dropbox**

Questo sistema di archiviazione basato su cloud utilizza Python nel suo client desktop.

### **11.3 Fattori alla base della crescita di Python nel mondo moderno**

La crescita di Python sta diventando importante e sta migliorando giorno dopo giorno. Gli ingegneri del software e gli sviluppatori preferiscono questo linguaggio per la sua versatilità e facilità d'uso. Vari altri fattori che stanno dietro la sua crescita sono i seguenti.

---

- **Buon supporto e comunità**

I linguaggi di programmazione affrontano molto spesso problemi di supporto. Inoltre, mancano di documentazione completa a causa della quale si verificano difficoltà quando vengono utilizzati dai programmatori. Python non ha questo problema ed è ben supportato. È disponibile una pletora di tutorial e documentazione per assistere i programmatori nel miglior modo possibile.

In aggiunta a ciò, ha una buona e attiva comunità la cui funzione è quella di fornire assistenza agli sviluppatori. I programmatori esperti aiutano principianti e studenti in questo senso.

---

- **Facile da codificare e scrivere**

Se confrontiamo Python con altri linguaggi di programmazione come Java, C o C ++, Python possiede un codice semplice e leggibile. La codifica è espressa in un modo relativamente semplice per farla capire completamente ai principianti.

Anche se per apprendere il livello avanzato della programmazione Python, è necessario molto tempo e impegno, ma per i principianti è un compito facile. Gli utenti possono identificare facilmente lo scopo del codice anche dopo averlo guardato.

---

- **Python è il linguaggio dell'educazione**

L'utilizzo dei PC nell'istruzione è cambiato radicalmente negli ultimi tempi. In precedenza, i sostituti potevano aver capito come digitare insieme ad attitudini introduttive essenziali nella remota possibilità che fossero fortunati. Oggigiorno, l'uso dell'innovazione include numerosi tipi di istruzione, con la codifica che assume la sua posizione legittima in numerose aule di studio in tutto il mondo.

Python è un linguaggio facile da usare. Possiede funzioni, espressioni, variabili e tutti gli altri elementi che gli studenti possono facilmente comprendere e praticare. Vicino alla lucidità che il linguaggio porta; è il linguaggio di programmazione standard per il Raspberry Pi, un PC strutturato in vista dell'allenamento. Questo amplia anche la formazione degli anni più giovani. I college mostrano Python, nelle scienze del computer e anche per gli studenti aritmetici. Inoltre, Matplotlib (un'importante libreria Python) viene utilizzato in soggetti a tutti i livelli per esprimere dati complessi. Python è anche uno dei dialetti che si sviluppano più rapidamente su Codecademy, e quindi è tutt'altro che difficile da adattare da remoto.

---

- **Semplice da codificare e scrivere**

Python ha un codice semplice e distinguibile se confrontato con altri linguaggi di programmazione come Java, C o C ++. Il codice viene comunicato in modo semplice, che può essere sostanzialmente decifrato anche da un ingegnere informatico alle prime armi.

Nonostante il fatto che per superare la programmazione Python, richiederà un sacco di fatica e tempo, tuttavia, accettare questo linguaggio senza alcuna preparazione è semplice per un tenero piede. Tuttavia, dando un'occhiata al codice, può determinare cosa dovrebbe fare il codice.

---

- **Python è perfetto per la creazione di prototipi**

Python non solo consente agli utenti di scrivere meno codici, ma fornisce anche l'utilità per costruire prototipi e idee molto rapidamente

Il brainstorming o l'ideazione è un aspetto importante dello sviluppo web, che viene per lo più trascurato, quindi la capacità di pensare a prototipi che possono funzionare più velocemente diventa molto più fondamentale.

---

- **L'integrazione e l'esecuzione sono rapide**

Python è considerato un linguaggio di alta classe. È il linguaggio più veloce quando si tratta di esecuzione e integrazione e fa risparmiare molto tempo ai programmatori.

Con progetti come PyPy e Numba, la velocità aumenta sempre di più e tali progetti tendono a renderlo il linguaggio più veloce ogni giorno che passa

---

- **Python ha una libreria standard**

Python contiene librerie che eliminano l'onere di comporre un codice da parte del programmatore. Queste librerie possiedono un'enorme quantità di funzioni integrate e codici già disponibili, quindi ogni volta non dovrebbe essere obbligatorio generare un codice.

---

- **Linguaggio multiplatforma**

Un altro dei migliori inclusi su Python è che è molto vario per quanto riguarda il framework di lavoro. In parole semplici, Python può essere utilizzato per un framework funzionante; ad esempio, Linux, Windows, Ubuntu e così via.

Pertanto, si può senza dubbio continuare a scappare da un prodotto senza tormentarsi per il supporto del framework. Può essere tradotto molto bene nella lingua con l'assistenza di un componente conveniente che lo rende redditizio da utilizzare. Per riassumere, componi il codice sullo stage Mac ed esegilo facilmente anche su Windows.

Poiché i punti salienti e le motivazioni per utilizzare Python sono chiari, dacci la possibilità di andare avanti verso il destino finale di Python. Non c'è dubbio che Python abbia capito come battere diversi dialetti che lo stanno portando al top.

Successivamente, ha una vasta gamma di possibilità che possono essere prese in considerazione in futuro. Nel caso in cui qualcosa, si è espanso nel tempo ed è andato oltre il 45% in solo un paio d'anni. Esistono quantità di organizzazioni soggette a sviluppatori Python per aggiornare l'esperienza del cliente.

---

- **Fornisce una pletora di strumenti**

Contiene un'enorme raccolta di librerie standard, che riduce lo sforzo per la scrittura di codici o funzioni. Le librerie in Python contengono sempre codici pre-scritti.

Alcuni degli strumenti sono i seguenti: Tkinter (uno sviluppo GUI), formato file, funzione incorporata, interprete Python personalizzato, protocolli Internet e supporto, modulo, ecc. Questa ampia raccolta aumenta l'utilità di Python come strumento di programmazione per i dati scienza.

---

- **Python è gratuito**

Dire che Python è gratuito potrebbe non essere di buon auspicio sin dall'inizio. Tutto considerato, indipendentemente dalla lingua che scegli di cercare, il semplice utilizzo della lingua non costa nulla.

Poiché Python è open source, è anche gratuito. Python sin dalla sua creazione è ancora gestito da Guido van Rossum. È totalmente Open Source e GPL. L'impegno di Python a queste convinzioni probabilmente non ti colpirà come significativo. Tuttavia, la programmazione open source ha ufficialmente cambiato il mondo. Lo stato senza obblighi nascosti di Python lo rende un dispositivo ideale da utilizzare per tutti.

## Conclusione

Congratulazioni! Se sei arrivato così lontano. Ci auguriamo che tu abbia veramente iniziato a comprendere i concetti di base e le complessità della programmazione Python. A questo punto, supponiamo, dovresti essere in grado di leggere quasi tutto il codice scritto in Python con sicurezza e comprensione.

Abbiamo cercato di coprire un buon numero di importanti concetti e funzionalità di Python in questo libro, inclusi alcuni dei concetti di programmazione orientata agli oggetti, nota anche come "OOP". Inoltre, abbiamo cercato di rendere la scrittura chiara e facile da capire, e questo libro include molti esempi teorici, pratici e spiegati.

Nel mondo dell'informatica e della programmazione informatica, potresti trovare molti libri accademici. Molti di loro sono progettati per gli studenti, e lì si possono trovare altri libri che sono appositamente destinati agli sviluppatori professionisti che hanno bisogno di consigli personali su come risolvere problemi di sintassi e problemi di runtime, nel codice, quando sviluppano un programma. Se impari codificando, infatti, in questo libro, poche pagine non hanno codice sorgente per Python, ma ogni concetto è dimostrato da almeno un esempio di codifica.

Gli esempi di codice sono molto ben formattati, facili da leggere e puliti in modo che tu possa trovare facile programmare in Python.

Andando avanti, se sei un principiante nell'apprendimento di Python, leggi questo libro perché contiene otto capitoli in modo che tu possa avere una migliore comprensione di Python entro una settimana.

Alcune persone potrebbero dire che questo libro non è per principianti! Guarda le sue dimensioni! Questo libro è semplicemente troppo travolgente per un principiante come te!



Non ascoltarli!

Dovresti sceglierne uno più facile se ne trovi uno. Bene, non possiamo dirti se questo libro è giusto o sbagliato. Immaginiamo che la comprensione di tutti sia abbastanza diversa. Possiamo suggerire che questo libro è adatto a una persona che è un principiante in programmazione di computer, ma un laureato in economia!

Ti ringrazio per aver acquistato questo libro, spero sia stato utile e ti sia piaciuto.

Se vuoi lasciare una recensione su Amazon, grazie mille, ti voglio bene.

Tony Chan.

## Riferimenti

- Tutorial - Javatpoint. (nd). Recuperato da <https://www.javatpoint.com/>
- Lutz, M. (2007). Imparare Python. Farnham: O'Reilly.
- Lutz, M. (2009). Riferimento tascabile di Python. Sebastopoli: O'Reilly Media, Inc.
- Slatkin, B. (2015). Python efficace: 59 modi specifici per scrivere meglio in Python. Upper Saddle River, NJ: Addison-Wesley.
- VanderPlas, J. (nd). Un rapido tour della sintassi del linguaggio Python. Recuperato da <https://jakevdp.github.io/WhirlwindTourOfPython/02-basic-python-syntax.html>
- 7. Dichiarazioni semplici. (nd). Recuperato da [https://docs.python.org/3/reference/simple\\_stmts.html](https://docs.python.org/3/reference/simple_stmts.html)
- Matthes, E. (2015). Corso intensivo di Python: un'introduzione pratica alla programmazione basata su progetti. San Francisco: No Starch Press.
- Pestunov, A. e Perov, A. (2017). Corso e-learning "Linguaggi di programmazione". Un corso di lezioni "Capacità di insegnamento dell'informazione e dell'ambiente educativo dell'Università". doi: 10.12731 / ofernio.2017.23146
- Shukla, XU e Parmar, DJ (2016). Python - Un linguaggio di programmazione completo ma gratuito per gli statistici. Journal of Statistics and Management Systems, 19 (2), 277-284. doi: 10.1080 / 09720510.2015.1103446

- Hetland, ML (2014). Algoritmi Python Padroneggiare gli algoritmi di base nel linguaggio Python. Berkeley, CA: Apress.