Overview of Git/GitHub

In questo video viene presentata una panoramica di Git e GitHub, **ambienti molto diffusi tra gli sviluppatori e i data scientist per eseguire il controllo di versione dei file di codice sorgente e dei progetti** e per collaborare con altri. Non si può parlare di Git e GitHub senza una comprensione di base di cosa sia il **controllo di versione.**

**Un sistema di controllo delle versioni** **consente di tenere traccia delle modifiche apportate ai documenti**. In questo modo è facile recuperare le versioni precedenti del documento in caso di errore e la collaborazione con gli altri è molto più semplice. Ecco un esempio che illustra il funzionamento del controllo di versione. Supponiamo che abbiate una lista della spesa e che vogliate che i vostri coinquilini confermino le cose di cui avete bisogno e ne aggiungano altre. Senza il controllo delle versioni, si crea un gran casino da sistemare prima di andare a fare la spesa. Con il controllo delle versioni, invece, sapete esattamente cosa vi serve dopo che tutti hanno contribuito con le loro idee.

Git è un software libero e open source distribuito sotto la GNU General Public License. Git è un sistema di controllo di versione distribuito, il che significa che gli utenti di tutto il mondo possono avere una copia del vostro progetto sul proprio computer; quando apportano modifiche, possono sincronizzare la loro versione su un server remoto per condividerla con voi. Git non è l'unico sistema di controllo delle versioni esistente, ma l'aspetto distribuito è uno dei motivi principali per cui è diventato uno dei sistemi di controllo delle versioni più diffusi. I sistemi di controllo delle versioni sono ampiamente utilizzati per il codice, ma è possibile controllare anche immagini, documenti e qualsiasi altro tipo di file. È possibile utilizzare Git senza un'interfaccia web utilizzando l'interfaccia della riga di comando, ma GitHub è uno dei servizi più popolari ospitati sul web per i repository Git. Altri includono GitLab, BitBucket e Beanstalk. Prima di iniziare, è necessario conoscere alcuni termini di base. Il protocollo SSH è un metodo di accesso remoto sicuro da un computer a un altro. Un repository contiene le cartelle del progetto impostate per il controllo di versione. Un fork è una copia di un repository. Una richiesta di pull è il modo in cui si richiede che qualcuno riveda e approvi le modifiche prima che diventino definitive. Una directory di lavoro contiene i file e le sottodirectory del computer associati a un repository Git. Ci sono alcuni comandi Git di base che si useranno sempre. Quando si inizia con un nuovo repository, è sufficiente crearlo una sola volta: localmente, e poi fare il push su GitHub, oppure clonando un repository esistente usando il comando "git init".

"git add" sposta le modifiche dalla directory di lavoro all'area di staging. "git status" permette di vedere lo stato della directory di lavoro e l'istantanea delle modifiche. "git commit" prende l'istantanea delle modifiche e le inserisce nel progetto. "git reset" annulla le modifiche apportate ai file della cartella di lavoro. "git log" consente di sfogliare le modifiche precedenti a un progetto. "git branch" consente di creare un ambiente isolato all'interno del repository per apportare modifiche. "git checkout" consente di vedere e modificare i rami esistenti. "git merge" consente di rimettere tutto insieme. Per imparare a usare Git in modo efficace e iniziare a collaborare con i data scientist di tutto il mondo, è necessario imparare i comandi essenziali. Per nostra fortuna, GitHub mette a disposizione risorse straordinarie per aiutarvi a iniziare. Andate su try.github.io per scaricare i fogli informativi e seguire le esercitazioni. Nei moduli che seguono, vi forniremo un corso accelerato sulla configurazione dell'ambiente locale e sull'avvio di un progetto.