

سوالات تئوری

git,github

۱. کنترل نسخه چیست و چرا توسعه‌دهندگان از آن استفاده می‌کنند؟
۲. تفاوت‌های Git با سایر سیستم‌های کنترل نسخه مانند SVN و Mercurial چیست؟
۳. چگونه می‌توان Git را روی سیستم‌عامل‌های مختلف نصب کرد؟
۴. دستور `git config --global` چه کاری انجام می‌دهد؟ چرا تنظیم نام کاربری و ایمیل در Git مهم است؟
۵. دستور `git init` چه کاری انجام می‌دهد و در چه شرایطی از آن استفاده می‌شود؟
۶. تفاوت بین `git add` و `git commit` چیست؟
۷. چگونه می‌توان وضعیت فعلی یک مخزن را بررسی کرد؟ دستور مرتبط را توضیح دهید.
۸. چرا توسعه‌دهندگان از GitHub استفاده می‌کنند؟
۹. چگونه می‌توان یک مخزن (Repository) جدید در GitHub ایجاد کرد؟
۱۰. تفاوت بین `git push` و `git pull` چیست؟
۱۱. چگونه می‌توان یک مخزن GitHub را روی سیستم خود کلون (clone) کرد؟
۱۲. چرا و چگونه باید مخزن محلی را به یک مخزن GitHub متصل کرد؟
۱۳. در یک تیم توسعه نرم‌افزار، چه مشکلاتی ممکن است هنگام همکاری روی یک مخزن مشترک پیش بیاید؟
۱۴. Merge Conflict چیست و چگونه می‌توان آن را حل کرد؟
۱۵. دستور `git remote add origin <URL>` چه کاری انجام می‌دهد؟

git,github

سوالات عملی

۱. Git را روی سیستم خود نصب کنید و بررسی کنید که به درستی نصب شده است.
۲. نام کاربری و ایمیل خود را در Git تنظیم کنید.
۳. یک مخزن جدید در سیستم خود ایجاد کنید و یک فایل متنی (README.md) در آن بسازید. سپس تغییرات را به مخزن اضافه کنید و یک کامیت انجام دهید.

۴. یک مخزن در **GitHub** ایجاد کنید و مخزن محلی خود را به آن متصل کنید. سپس تغییرات را در **GitHub** پوش کنید.
۵. یک فایل جدید به مخزن خود اضافه کنید، تغییرات را استیج (**stage**) کنید و سپس کامیت و پوش کنید.
۶. مخزن یک هم‌تیمی را کلون کنید و یک تغییر کوچک در آن ایجاد کرده و پوش کنید.
۷. دستورات **git pull** و **git fetch** را اجرا کنید و تفاوت آن‌ها را توضیح دهید.
۸. یک شاخه (**branch**) جدید در مخزن خود ایجاد کنید، یک تغییر روی آن انجام دهید، سپس به شاخه اصلی (**main**) برگردید و تغییرات را مرج کنید.
۹. یک فایل را ویرایش کرده و تغییرات را بدون کامیت، بازگردانی کنید.
۱۰. یک تغییر انجام دهید و آن را کامیت کنید، سپس کامیت آخر را ریست (**reset**) کنید.
۱۱. یک تغییر انجام دهید و بدون استفاده از **git add**، تغییرات را کامیت کنید. بررسی کنید که چه اتفاقی می‌افتد.
۱۲. یک هم‌تیمی را به مخزن **GitHub** خود اضافه کنید و با او روی یک فایل مشترک همکاری کنید.
۱۳. یک تعارض (**Merge Conflict**) ایجاد کنید و آن را حل کنید.