

प्रश्न: Git काय आहे?

Question: What is Git?

उत्तर:

Git हे एक वितरित आवृत्ती नियंत्रण प्रणाली आहे जी सॉफ्टवेअर विकास दरम्यान सोर्स कोडमधील बदल ट्रॅक करते.

Answer:

Git is a distributed version control system that tracks changes in source code during software development.

प्रश्न: GitHub काय आहे?

Question: What is GitHub?

उत्तर:

GitHub हे एक वेब-आधारित प्लॅटफॉर्म आहे ज्यावर Git रिपॉझिटरी होस्ट केल्या जातात आणि कोडवर सहकार्य केले जाते.

Answer:

GitHub is a web-based platform for hosting Git repositories and collaborating on code.

प्रश्न: Git आणि GitHub मध्ये काय फरक आहे?

Question: What is the difference between Git and GitHub?

उत्तर:

- **Git:** आवृत्ती नियंत्रण साधन.
- **GitHub:** Git रिपॉझिटरी शेअर करण्यासाठी आणि सहकार्य करण्यासाठी एक प्लॅटफॉर्म.

Answer:

- **Git:** Version control tool.

- **GitHub:** A platform for sharing and collaborating on Git repositories.
-

प्रश्न: Git चा वापर केल्याने काय फायदे होतात?

Question: What are the benefits of using Git?

उत्तर:

- कोड बदल ट्रॅक करणे.
- सहकार्य करण्याची परवानगी देते.
- ब्रॅचिंग आणि मर्जिंग प्रदान करते.
- मागील आवृत्त्यांकडे रोलबॅक करण्याची सुविधा देते.

Answer:

- Tracks code changes.
 - Allows collaboration.
 - Provides branching and merging.
 - Enables rollbacks to previous versions.
-

प्रश्न: Git मध्ये रिपॉझिटरी म्हणजे काय?

Question: What is a repository in Git?

उत्तर:

रिपॉझिटरी म्हणजे एक डायरेक्टरी किंवा संग्रहण जागा जिथे आपल्या प्रकल्पाच्या फाईल्स आणि त्यांचा इतिहास साठवला जातो.

Answer:

A repository is a directory or storage space where your project's files and their history are stored.

प्रश्न: git clone आणि git pull यामध्ये काय फरक आहे?

Question: What is the difference between git clone and git pull?

उत्तर:

- **git clone:** एक रिमोट रिपॉझिटरीची कॉपी तयार करते.
- **git pull:** रिमोट रिपॉझिटरीमधील बदल डाउनलोड करते आणि ते स्थानिक ब्रॅचमध्ये मर्ज करते.

Answer:

- **git clone:** Creates a copy of a remote repository.
 - **git pull:** Fetches changes from a remote repository and merges them into the local branch.
-

प्रश्न: Git मध्ये एक कमिट काय आहे?

Question: What is a commit in Git?

उत्तर:

एक कमिट म्हणजे रिपॉझिटरीमधील बदलांचा स्नॅपशॉट. ते प्रकल्पाच्या इतिहासातील एक विशिष्ट बिंदू दर्शवते.

Answer:

A commit is a snapshot of changes in the repository. It represents a specific point in the project's history.

प्रश्न: Git मध्ये ब्रॅच काय आहे?

Question: What is a branch in Git?

उत्तर:

ब्रॅच म्हणजे एक कमिटचा पॉइंटर, ज्यामुळे आपण वेगवेगळ्या वैशिष्ट्यांवर किंवा दुरुस्त्या स्वतंत्रपणे काम करू शकता.

Answer:

A branch is a pointer to a specific commit, allowing you to work on different features or fixes independently.

प्रश्न: Git मध्ये HEAD म्हणजे काय?

Question: What is the HEAD in Git?

उत्तर:

HEAD म्हणजे वर्तमान ब्रॅच रेफरन्सला पॉइंट करणारा पॉइंटर.

Answer:

HEAD is a pointer to the current branch reference.

प्रश्न: Git मध्ये स्टेजिंग एरिया म्हणजे काय?

Question: What is the staging area in Git?

उत्तर:

स्टेजिंग एरिया एक जागा आहे जिथे बदल कमिट करण्यापूर्वी जोडले जातात.

Answer:

The staging area is a space where changes are added before committing them to the repository.

प्रश्न: git fetch आणि git pull यांमध्ये काय फरक आहे?

Question: What is the difference between git fetch and git pull?

उत्तर:

- **git fetch:** रिमोट रिपॉझिटरीमधून बदल डाउनलोड करते परंतु मर्ज करत नाही.
- **git pull:** git fetch आणि git merge यांचा समावेश करतो.

Answer:

- **git fetch:** Downloads changes from a remote repository without merging.

- **git pull:** Combines git fetch and git merge.
-

प्रश्न: Git मध्ये मर्ज कॉन्फ्लिक्ट काय आहे?

Question: What is a merge conflict in Git?

उत्तर:

मर्ज कॉन्फ्लिक्ट तेव्हा होतो जेव्हा दोन वेगवेगळ्या ब्रॅचमधून बदल एकाच फाईलमध्ये होतात आणि Git त्यांना आपोआप मर्ज करू शकत नाही.

Answer:

A merge conflict occurs when changes from different branches conflict and Git cannot automatically resolve them.

प्रश्न: Git मध्ये मर्ज कॉन्फ्लिक्ट कसा सोडवावा?

Question: How do you resolve merge conflicts in Git?

उत्तर:

- कॉन्फ्लिक्ट झालेल्या फाईल्सची ओळख पटवा.
- मॅन्युअली कॉन्फ्लिक्ट संपादित करा.
- git add वापरून सोडवलेले कॉन्फ्लिक्ट चिन्हांकित करा.
- बदल कमिट करा.

Answer:

- Identify conflicting files.
 - Manually edit conflicts.
 - Mark resolved conflicts using git add.
 - Commit the changes.
-

प्रश्न: GitHub मध्ये फोर्क काय आहे?

Question: What is a fork in GitHub?

उत्तर:

फोर्क म्हणजे एक रिपॉझिटरीची कॉपी, ज्यामुळे आपण मूळ रिपॉझिटरीपासून स्वतंत्रपणे बदल करू शकता.

Answer:

A fork is a copy of a repository that allows you to make changes independently from the original repository.

प्रश्न: git reset आणि git revert यांमध्ये काय फरक आहे?

Question: What is the difference between git reset and git revert?

उत्तर:

- **git reset:** ब्रॅच पॉइंटर हलवते आणि काम करणाऱ्या डायरेक्टरीला बदलू शकते.
- **git revert:** इतिहास बदलले न करता बदल उलटवण्यासाठी एक नवीन कमिट तयार करते.

Answer:

- **git reset:** Moves the branch pointer and optionally modifies the working directory.
 - **git revert:** Creates a new commit to undo changes without modifying history.
-

प्रश्न: Git मध्ये टॅग काय आहे?

Question: What is a Git tag?

उत्तर:

टॅग म्हणजे एक विशिष्ट कमिट, जो प्रामुख्याने रिलीज वर्शनसाठी वापरला जातो.

Answer:

A tag marks a specific commit, often used for versioning releases.

प्रश्न: .gitignore फाइलचे उद्देश काय आहे?

Question: What is the purpose of .gitignore?

उत्तर:

.gitignore फाइल जी फाईल्स किंवा डायरेक्टरीज निर्दिष्ट करते ज्या Git रिपॉझिटरीमध्ये समाविष्ट कराव्यात.

Answer:

The .gitignore file specifies files or directories to ignore in Git repositories.

प्रश्न: git stash काय आहे?

Question: What is git stash?

उत्तर:

git stash अस्थायीपणे बदल साठवतो, ज्यामुळे तुम्ही दुसऱ्या कार्यावर काम करू शकता आणि त्यानंतर ते बदल पुन्हा लागू करू शकता.

Answer:

git stash stash change temporarily, allowing you to work on other tasks without committing them.

प्रश्न: GitHub मध्ये Pull Request काय आहे?

Question: What is a pull request in GitHub?

उत्तर:

Pull Request म्हणजे एका ब्रॅच किंवा फोर्क मधून दुसऱ्या ब्रॅचमध्ये बदल मर्ज करण्याची विनंती.

Answer:

A pull request is a request to merge changes from one branch or fork into another branch.

प्रश्न: Git मध्ये ब्रॅच कसा डिलीट करावा?

Question: How do you delete a branch in Git?

उत्तर:

- स्थानिक: `git branch -d <branch-name>`
- रिमोट: `git push origin --delete <branch-name>`

Answer:

- Locally: `git branch -d <branch-name>`
- Remotely: `git push origin --delete <branch-name>`

प्रश्न: Git मध्ये सबमॉड्यूल्स म्हणजे काय?

Question: What are submodules in Git?

उत्तर:

सबमॉड्यूल्स म्हणजे रिपॉझिटरीमधील रिपॉझिटरी, जी तुम्हाला बाह्य प्रकल्प समाविष्ट करण्याची परवानगी देते.

Answer:

Submodules are repositories within a repository, allowing you to include external projects.

प्रश्न: Git Bisect काय आहे?

Question: What is git bisect?

उत्तर:

Git Bisect हे एक डिबगिंग साधन आहे जे त्या कमिटला शोधण्यासाठी वापरले जाते ज्यामुळे बग निर्माण झाला.

Answer:

Git bisect is a debugging tool to find the commit that introduced a bug.

प्रश्न: Git Blame काय आहे?

Question: What is git blame?

उत्तर:

Git Blame ही एक कमांड आहे जी प्रत्येक ओळच्या बदलासाठी कोण, कधी आणि कुठल्या कमिटने बदल केला हे दाखवते.

Answer:

Git blame shows who made changes to each line of a file and when.

प्रश्न: Git मध्ये कमिट्स कशा स्क्वॅश कराव्यात?

Question: How do you squash commits in Git?

उत्तर:

git rebase -i वापरून अनेक कमिट्स एकाच कमिटमध्ये एकत्र करा.

Answer:

Use git rebase -i to combine multiple commits into one.

प्रश्न: Git Hooks काय आहे?

Question: What are Git hooks?

उत्तर:

Git Hooks म्हणजे स्क्रिप्ट्स ज्या Git इव्हेंट्सच्या प्रतिसादात आपोआप चालविल्या जातात, जसे की कमिट्स किंवा मर्जेस.

Answer:

Git hooks are scripts that run automatically in response to Git events, such as commits or merges.

प्रश्न: Git मध्ये Detached HEAD स्टेट काय आहे?

Question: What is a detached HEAD state in Git?

उत्तर:

Detached HEAD स्टेट म्हणजे त्या स्थितीत जिथे HEAD एखाद्या ब्रॅचऐवजी विशिष्ट कमिटकडे पॉइंट करत असतो.

Answer:

A detached HEAD state in Git is where HEAD points to a specific commit instead of a branch.

प्रश्न: Git मध्ये कमिट कसे रिव्हर्ट करावं?

Question: How do you roll back a commit?

उत्तर:

- स्थानिक कमिट्स हटवण्यासाठी git reset वापरा.
- इतिहास न बदलता कमिट उलटवण्यासाठी git revert वापरा.

Answer:

- Use git reset to undo local commits.
- Use git revert to undo a commit without modifying history.

प्रश्न: Git मध्ये git push कसे उलटवावे?

Question: How do you undo a git push?

उत्तर:

git revert वापरा किंवा git push origin +<branch-name> वापरून फोर्स पुश करा.

Answer:

Use git revert or force push using git push origin +<branch-name>.

प्रश्न: Git मध्ये ब्रॅच कसा नाव ठेवावा?

Question: How do you rename a branch in Git?

उत्तर:

git branch -m <new-branch-name> वापरून ब्रॅचचे नाव बदला.

Answer:

Use git branch -m <new-branch-name> to rename a branch.

प्रश्न: दोन ब्रॅचेसमधील फरक कसा तपासावा?

Question: How do you check the difference between two branches?

उत्तर:

git diff <branch1> <branch2> वापरून दोन ब्रॅचेसमधील फरक तपासा.

Answer:

Use git diff <branch1> <branch2> to check the difference between two branches.

प्रश्न: GitHub वर ओपन-सोर्स प्रकल्पात कसे योगदान करावे?

Question: How do you contribute to an open-source project on GitHub?

उत्तर:

1. रिपॉझिटरी फोर्क करा.
2. फोर्क क्लोन करा.
3. एक नवीन ब्रॅच तयार करा.
4. बदल करा आणि कमिट करा.
5. ब्रॅच पुश करा आणि पुल रिक्वेस्ट तयार करा.

Answer:

1. Fork the repository.
2. Clone the fork.
3. Create a branch.
4. Make changes and commit.
5. Push the branch and create a pull request.

प्रश्न: रिपॉझिटरीची स्थिती कशी तपासावी?

Question: How do you check the status of a repository?

उत्तर:

git status कमांड वापरा.

Answer:

Use the command git status.

प्रश्न: फोर्क केलेली रिपॉझिटरी कशी अपडेट करावी?

Question: How do you update a forked repository?

उत्तर:

अपस्ट्रीम रिपॉझिटरी जोडा आणि बदल पुल करा.

git remote add upstream <repo-url>

git pull upstream <branch-name>

Answer:

Add the upstream repository and pull changes.

git remote add upstream <repo-url>

git pull upstream <branch-name>

प्रश्न: Git ची कार्यक्षमता कशी सुधारावी?

Question: How do you improve Git performance?

उत्तर:

1. शॅलो क्लोन (--depth) वापरा.
2. स्पॅर्स चेकआउट वापरा.
3. अनावश्यक ब्रॅचेस आणि ऑब्जेक्ट्स साफ करा.

Answer:

1. Use shallow clones (--depth).
 2. Use sparse checkout.
 3. Clean up unnecessary branches and objects.
-

प्रश्न: git fsck काय आहे?

Question: What is git fsck?

उत्तर:

git fsck कमांड Git रिपॉझिटरीतील ऑब्जेक्ट्सची अखंडता तपासते.

Answer:

git fsck verifies the integrity of objects in a Git repository.

प्रश्न: git archive आणि git bundle यामधील फरक काय आहे?

Question: What is the difference between git archive and git bundle?

उत्तर:

- git archive: रिपॉझिटरीचे टार किंवा झिप आर्काइव्ह तयार करते.
- git bundle: एक पोर्टेबल Git रिपॉझिटरी तयार करते.

Answer:

- git archive: Creates a tar or zip archive of a repository.
 - git bundle: Creates a portable Git repository.
-

प्रश्न: git clean काय करते?

Question: What does git clean do?

उत्तर:

git clean अनट्रॅक्ड फायली आणि डिरेक्टरीज काम करणाऱ्या डिरेक्टरीतून काढून टाकते.

Answer:

git clean removes untracked files and directories from the working directory.

प्रश्न: git gc काय आहे?

Question: What is git gc?

उत्तर:

git gc अनावश्यक फायली साफ करून रिपॉझिटरीला ऑप्टिमाइझ करण्यासाठी गॅरेज कलेक्शन करतो.

Answer:

git gc performs garbage collection to clean up unnecessary files and optimize the repository.

प्रश्न: एक अपयशी मर्ज कसा डिबग करावा?

Question: How do you debug a failing merge?

उत्तर:

git log, git diff वापरा आणि संघर्षांना manually सोडवा.

Answer:

Use git log, git diff, and resolve conflicts manually.

प्रश्न: सर्व रिमोट ब्रॅचेस कशा सूचीबद्ध कराव्यात?

Question: How do you list all remote branches?

उत्तर:

git branch -r कमांड वापरा.

Answer:

Use the command git branch -r.

प्रश्न: Git कसे जागतिक पातळीवर कॉन्फिगर करावे?

Question: How do you configure Git globally?

उत्तर:

git config --global <key> <value> वापरा.

Answer:

Use git config --global <key> <value>.

प्रश्न: git reset --soft काय करते?

Question: What does git reset --soft do?

उत्तर:

git reset --soft HEAD पॉइंटर हलवते पण स्टेजिंग क्षेत्रातील बदल कायम ठेवते.

Answer:

git reset --soft moves the HEAD pointer but retains the changes in the staging area.

प्रश्न: Git मध्ये मोठ्या फाइल्स कशा हाताळाव्यात?

Question: How do you handle large files in Git?

उत्तर:

Git LFS (लार्ज फाइल स्टोरेज) वापरा मोठ्या फाइल्सचे ट्रॅकिंग आणि स्टोरेज करण्यासाठी.

Answer:

Use Git LFS (Large File Storage) to track and store large files.