WDV341 Intro PHP

Misti Christianson

Git Key Terms

* VERSION CONTROL SOFTWARE --records changes to a file or set of files over time so that you can recall specific versions later. Allows you to:
  + Revert selected files back to a previous state
  + Revert entire project back to a previous state
  + Compare changes over time
  + See who last modified something that might be causing a problem or who introduced an issue and when, etc
  + Easily recover when encounter failures or lose files
  + Local version control systems are set up on a individual computers with a database that kept track of file changes
  + Centralized version control systems provided a central server/computer that contained all the versioned files and files could be checked out from the central location. Complications may arise if the central server ever went down.
  + Distributed version control systems in which clients don’t check out a file but fully mirror the whole repository and the full history. Each client computer then contains a backup if the main server goes down.
  + Source: <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>
* ADD --git command the updates the index using the current content found in the working tree; prepare the content staged for the next commit.
  + Can be performed multiple times before a commit
  + Only adds the content of the specified file(s) at the time the add command is run
  + Will not add ignored files; will fail if try to add files specified in the ignored file but can add ignored files with -f (force) option. Ex. git add -f <file\_name>
  + Opposite of git add is git rm to remove any files from the index and staging area.
  + Source: https://git-scm.com/docs/git-add
* COMMIT --creates a new commit containing the current contents of the index and the given log message describing the changes.
  + New commit is direct child of HEAD, the branch is updated to point to it.
  + git commit –dry-run: can be used to obtain a summary of what is included for the next commit
  + -m “”/--message=””: to add a message to the commit; otherwise will be prompted to add a message.
  + -a/--all: automatically stage files that have been modified/deleted; new files you have not told Git about are not affected
  + -p/--patch: choose which changes to commit
  + Lots of available options with the git commit command; too many to list and most of which we will not be using in this class.
  + Source: <https://git-scm.com/docs/git-commit>
* PUSH --Updates remote refs(references) using local refs, while sending objects necessary to complete the given refs. Pushes local changes/commit up to the repository.
  + Default behavior is to push the current branch to the corresponding upstream branch but will abort if the upstream branch does not have the same name as the local one. If no branch is specified the default is ‘origin’.
  + Source: <https://git-scm.com/docs/git-push>
* PULL --Incorporates changes from a remote repository into the current branch.
  + Runs git fetch with given parameters, followed by git rebase or git merge depending on config options or command line flags
  + Default is to bring the current branch up to date to match the remote
  + --rebase/--no-rebase: if the branch and remote have diverged, need to specify how to reconcile divergent branches
  + Git reset –merge: cancel a conflicting merge
  + If any remote changes overlap with local uncommitted changes, merge automatically canceled and work tree untouched.
  + Source: <https://git-scm.com/docs/git-pull/en>
* CLONE --clones a repository into a newly created directory, created remote-tracking branches for each branch in cloned repository, clones and checks out an initial branch forked from cloned repo’s currently active branch