Section 13: Github Grab Bag: Odds & Ends

115. What Really Matters In This Section

116. Github Repo Visibility: Public Vs. Private

* Public Repos: Public repos are accessible to

everyone on the internet. Anyone can see the repo on Github

* Private Repos: Private repos are only accessible

to the owner and people who have been granted access.

* We can change the repos visibility in settings tab.

117. Adding Github Collaborators

* We can add collaborators in github by going to settings and manage access tab.

118. Github Collaboration Demo

* To collaborate on a project clone the file, then push it to the repo.
* Pull the file from repo to start working over the recent commit.

119. What are READMEs?

* README file is used to communicate important information about a repository including
  + What the project does
  + How to run the project
  + Why it's noteworthy
  + Who maintains the project

120. A Markdown Crash Course

* READMEs are Markdown files, ending with the .md

extension. Markdown is a convenient syntax to

generate formatted text. It's easy to pick up!

* This tool generates markup "<https://daringfireball.net/>"
* This tool also generates markup "<https://markdown-it.github.io/>"

121. Adding a README To A Project

* Add README.md file inside your directory.
* Vscode has markdown preview tool.

122. Creating Github Gists

* Github Gists are a simple way to share code snippets and useful fragments with others. Gists are much easier to create, but offer far fewer features than a typical Github repository.
* To go to gists page: Go to profile -> click gists.

123. Introducing Github Pages

* Gitub Pages are public webpages that are hosted and published via Github. They allow you to create a website simply by pushing your code to Github.
* Gitub pages only support static pages. Just HTML/CSS/JS
* Two types of github page
  + User site : Every github account has one personal github site.
  + Project Sites : We get unlimited project sites. Each github repo can have a corresponding hosted website.

124. Github Pages Demo

* To create a Github page make a different branch for the page for convenience.

Section 14: Git Collaboration Workflows

125. What Really Matters In This Section

126. The Pitfalls Of A Centralized Workflow

* Centralized workflow : The simplest collaborative workflow is to have everyone work on the master branch.
* Drawback of centralized workflow is that if person 1 commits and pushes to git repo and person 2 commits and pushes to repo. A error pops saying updates rejected because tip of current branch is behind its remote counterpart.

127. Centralized Workflow Demonstration

* Demonstration of centralized workflow and conflicts which may occur during this type of workflow.

128. The All-Important Feature Branch Workflow

* Pro's of working on a feature branch
  + Master branch is treated as official project history
  + Multiple teammates can collaborate on a single feature branch without polluting master branch.
  + Master branch won't contain broken code.
* Con's of working on a feature branch
  + Lots of time spent on resolving conflicts and merging code.

129. Feature Branch Workflow Demo

* Demonstration of feature branch workflow and its useful feature is shown in the demo.

130. Merging Feature Branches

* Use merge command to merge a feature branch onto main branch.

131. Introducing Pull Requests

* Pull requests are a feature built in to products like github and Bitbucket. They are not native to git itself.
* To merge the feature branch to main branch , we have to create a pull request and if approved feature will get merged to main.

132. Making Our First Pull Request

* Create a pull request from github.
* Owner will get a notification about merging the pull request.

133. Merging Pull Requests With Conflicts

* Pull requests can be merged manually with conflicts by changing or approving the changes made in the file and merging it to the main branch.

134. Configuring Branch Protection Rules

* To configure branch protection rules : Go to settings -> Branch tab -> click add rule under branch protection rules.
* In a big organization protection rules are necessary for a proper functioning.

135. Introducing Forking

* Fork and clone is another type of a workflow.
* Which is used in a large project such as open source projects.
* Forking : Github (and similar tools) allow us to create personal

copies Of Other peoples' repositories. We call those

copies a "fork" of the original.

136. Forking Demonstration

* Click on fork button to fork that project to your github repo.
* Make changes to the file and push it back to the remote repo.

137. The Fork & Clone Workflow

* To share the work make a pull request from your fork to the original repo.
* Having two remote repos, One remote origin and one upstream pointing to the main repo.

138. Fork & Clone Workflow Demonstration

* Push the changes to remote origin repository.
* And from the remote repository make a pull request to the main repository.
* If accepted the files will be added to the main repository.