allgemeine Definitionen

Branch

Ein Branch (deutsch: Zweig) in GitHub ist ein eigenständiger Entwicklungszweig innerhalb eines Repositories. Er ermöglicht es, neue Features zu entwickeln, Fehler zu beheben oder Ideen auszuprobieren, ohne den Hauptzweig (meist „main“ oder „master“) zu beeinflussen. Ein Branch ist dabei wie ein „abgetrennter Arbeitsbereich“ oder eine Art „Paralleluniversum“ für deinen Code

Typischerweise wird ein Branch vom aktuellen Stand eines bestehenden Branches erstellt. Änderungen, die du auf einem Branch vornimmst, sind isoliert von anderen Branches. Erst wenn du zufrieden bist, kannst du die Änderungen über einen sogenannten Merge (Zusammenführung) wieder in den Hauptzweig übernehmen

Repüository

Ein Repository auf GitHub ist das grundlegendste Element der Plattform. Es ist ein Speicherort, an dem du deinen gesamten Code, alle zugehörigen Dateien sowie deren vollständigen Revisionsverlauf ablegen und verwalten kannst. In einem Repository können mehrere Personen gemeinsam arbeiten, Änderungen verfolgen und diskutieren. Repositories können entweder öffentlich (für alle sichtbar) oder privat (nur für ausgewählte Personen zugänglich) sein

Pull

Auf GitHub bedeutet „pull“, dass du Änderungen aus einem entfernten (Remote-)Repository herunterlädst und diese direkt in dein lokales Repository integrierst. Technisch kombiniert der Befehl `git pull` zwei Schritte: Zuerst werden mit `git fetch` die neuesten Änderungen vom Remote-Repository geholt, anschließend werden diese Änderungen mit `git merge` in deinen aktuellen lokalen Branch eingepflegt. So bleibt dein lokaler Stand immer aktuell mit dem, was auf GitHub veröffentlicht wurde.

Mit „pull“ synchronisierst du dein lokales Repository mit den neuesten Änderungen aus dem Remote-Repository auf GitHub

Fetch

Fetch auf GitHub bedeutet, dass du die neuesten Änderungen (wie neue Commits, Branches oder Tags) aus einem entfernten Repository herunterlädst und in dein lokales Repository überträgst – allerdings ohne diese Änderungen direkt in deinen aktuellen Arbeitsstand (Branch) zu übernehmen.

Mit `fetch` kannst du dir also zuerst anschauen, was sich im Remote-Repository geändert hat, bevor du entscheidest, ob und wie du diese Änderungen in deinen eigenen Code integrierst. Erst durch einen anschließenden Merge oder Rebase werden die heruntergeladenen Änderungen mit deinem aktuellen Stand zusammengeführt

From:
http://wiki.waldhofer.at/ - **Wiki von Franz**

Permanent link:
http://wiki.waldhofer.at/doku.php?id=github:allgemein

Last update: **2025/05/30 14:40**