

1. Was bedeutet "HEAD" in Git?

- **HEAD** zeigt auf den **aktuellen Stand** deines Projekts, also den **aktiven Commit** oder Branch, an dem du gerade arbeitest.
- Es ist wie ein **Marker**, der sagt: „Hier bist du gerade“.
- Wenn du zu einem anderen Branch wechselst oder zu einem früheren Commit zurückgehst, bewegt sich **HEAD** zu dieser neuen Position.
- Beispiel: Wenn du gerade auf dem Branch **master** arbeitest, zeigt **HEAD** auf den letzten Commit auf diesem Branch.

2. Was bedeutet "master" in Git?

- **master** (oder in neueren Versionen oft **main** genannt) ist der **Standard-Branch** in Git.
- Der **master**-Branch enthält in der Regel den **stabilen Code** deines Projekts, und oft werden neue Features in anderen Branches entwickelt und dann in den **master**-Branch zurückgeführt, wenn sie fertig sind.
- Du kannst andere Branches erstellen, um neue Funktionen oder Änderungen zu entwickeln, aber der **master**-Branch bleibt oft der Hauptzweig, in dem die stabile Version des Projekts liegt.

Einfache Analogie:

- Stell dir Git als einen Baum vor:
 - Der **master**-Branch ist der **Stamm** des Baumes, der das Hauptprojekt darstellt.
 - **HEAD** ist wie ein **Post-it-Zettel**, der auf den **Ast** zeigt, an dem du gerade arbeitest. Wenn du zu einem anderen Ast (Branch) wechselst, bewegt sich der Zettel dorthin.

Beispiel:

- Wenn du im **master**-Branch arbeitest, zeigt **HEAD** auf den letzten Commit im **master**.
- Wenn du in einen neuen Branch, z. B. **feature-login**, wechselst, wird **HEAD** auf den neuesten Commit in diesem Branch zeigen.

Zusammengefasst:

- **HEAD** zeigt dir, wo du dich gerade im Projekt befindest (welcher Commit oder Branch aktiv ist).
- **master** ist der **Hauptbranch** deines Projekts, in dem der stabile Code liegt.