Technische Hochschule
mannheim

Coden mit KI

Effektiv programmieren mit VS Code, Git und AI-Assistenten

Ablauf des Workshops

Technische Hochschule mannheim

1 · VS Code als Basis

Editor einrichten, Erweiterungen, Git im Editor (Rekap)

2 · AI-Coding – Grundlagen

Wie Assistenten funktionieren, Interaktionsmodi, Werkzeuge, Gratis-Angebote

3 · Effektiv mit KI coden

Prompting, Kontext, typische Aufgaben, Agent-Modus

4 · Qualität, Grenzen, Verantwortung

Risiken, Review-Pflicht, Datenschutz, Prüfungsregeln

5 · Workflow & Praxis

Von der Aufgabe bis zum Commit – mit Übungen

Lernziele

Technische Hochschule mannheim

VS Code produktiv einrichten und mit Git nutzen

Verstehen, wie KI-Assistenten arbeiten – und wo ihre Grenzen liegen

Die drei Interaktionsmodi (Vervollständigung, Chat, Agent) einsetzen

Wirksame Prompts schreiben und Kontext gezielt geben

KI-Ergebnisse kritisch prüfen, testen und verantwortungsvoll nutzen

Einen durchgängigen Workflow von der Aufgabe bis zum Commit beherrschen

Technische Hochschule
mannheim

1 VS Code als Basis

Der Editor, in dem alles zusammenläuft

Was ist Visual Studio Code?

Technische Hochschule mannheim

Leichtgewichtiger, erweiterbarer Code-Editor

Kostenlos, plattformübergreifend (Windows, macOS, Linux)

Erweiterungen (Extensions)

Sprachen, Linter, Debugger – und KI-Assistenten

Integriert

Terminal, Debugger und Git direkt im Editor

Wichtige Bedienelemente

Command Palette (Strg/Cmd+Shift+P) – der schnellste Weg zu allem

Seitenleiste: Explorer, Suche, Source Control, Extensions

Git in VS Code (Rekap)

Technische Hochschule mannheim

Source-Control-Ansicht

Änderungen sehen, stagen und committen – ohne Kommandozeile

Typischer Ablauf im Editor

Änderungen prüfen → Stage (+) → Commit-Nachricht → Commit

Branch unten links wechseln/erstellen; „Sync“ = pull + push

Diffs

Änderungen werden Seite an Seite hervorgehoben

Hinweis

Git-Grundlagen setzen wir voraus – hier nur die Einbettung in VS Code

VS Code für KI einrichten

Technische Hochschule mannheim

1 · Erweiterung installieren

KI-Assistent aus dem Extensions-Marketplace hinzufügen

2 · Anmelden

Mit dem Konto des Anbieters verbinden (z. B. GitHub-Konto)

3 · Bedienung kennenlernen

Inline-Vorschläge, Chat-Seitenleiste, Inline-Chat (Strg/Cmd+I)

4 · Datenschutz prüfen

Was wird gesendet? Telemetrie und Einstellungen kontrollieren

Technische Hochschule
mannheim

2 AI-Coding – Grundlagen

Wie die Assistenten arbeiten und was es gibt

Wie funktionieren KI-Code-Assistenten?

Technische Hochschule mannheim

Große Sprachmodelle (LLM)

Trainiert auf riesigen Mengen Text und Quellcode

Prinzip

Vorhersage des wahrscheinlich nächsten Tokens auf Basis des Kontexts

Wichtige Folgerung

Sie „verstehen“ nicht wie ein Mensch – sie erzeugen plausible Muster

Kontext ist entscheidend

Je besser der mitgegebene Kontext, desto besser das Ergebnis

Technische Hochschule mannheim

Drei Arten der Interaktion

1

Vervollständigung

Vorschläge beim Tippen

Tab zum Übernehmen

Schnell für Routinecode

2

Chat

Fragen & erklären lassen

Code besprechen

Inline-Chat im Editor

3

Agent

Mehrschrittige Aufgaben

Ändert mehrere Dateien

Führt Schritte aus

– unter Aufsicht

Werkzeuglandschaft (toolneutral)

Technische Hochschule mannheim

Im Editor integriert

GitHub Copilot (in VS Code), Continue, Cline u. a.

KI-zentrierte Editoren

Cursor, Windsurf – VS-Code-ähnlich, KI tief eingebaut

Chat-/Agenten-Assistenten

Claude, ChatGPT/Codex, Gemini – auch als CLI/Agent nutzbar

Grundsatz

Die Prinzipien sind übertragbar – Werkzeuge wechseln schnell

Kostenlos für Studierende? (Stand Juli 2026)

Technische Hochschule mannheim

GitHub Copilot Free

Für alle kostenlos in VS Code – begrenzt (u. a. ~2.000 Vervollständigungen/Monat)

GitHub Copilot über den Student Developer Pack

Kostenlos mit Hochschul-Verifizierung; Neuanmeldungen zeitweise pausiert

Claude (Anthropic)

Kostenlose Stufe auf claude.ai für Code-Hilfe; „Claude for Education“, falls die Hochschule eine Lizenz hat

OpenAI Codex for Students

100 $ Guthaben – aber nur USA/Kanada, in Deutschland nicht verfügbar

Cursor / Google

Cursor: kostenlose „Hobby“-Stufe; Gemini Code Assist Free in VS Code eingestellt (→ Antigravity)

Angebote ändern sich häufig – vor Nutzung aktuell prüfen

Technische Hochschule
mannheim

3 Effektiv mit KI coden

Vom vagen Wunsch zum brauchbaren Code

Prompting für Code – die Grundregeln

Technische Hochschule mannheim

Kontext geben

Sprache, Framework, Ziel und Randbedingungen nennen

Präzise sein

Erwartete Ein-/Ausgaben, Beispiele und Einschränkungen angeben

Iterativ arbeiten

Ergebnis prüfen, Feedback geben, verfeinern – nicht alles auf einmal

Zerlegen

Große Aufgaben in kleine, prüfbare Schritte aufteilen

Beispiel: schwacher vs. guter Prompt

Technische Hochschule mannheim

Schwach

„Schreib eine Funktion, die Daten verarbeitet."

Gut

„Schreibe eine Python-Funktion parse_csv(pfad: str) -> list[dict], die eine CSV mit Header liest, Zeilen als dicts zurückgibt, leere Zeilen überspringt und bei fehlender Datei einen aussagekräftigen Fehler wirft. Mit Docstring und Typannotationen."

Warum besser?

Klare Signatur, Verhalten, Fehlerfälle und Qualitätsanforderungen

Kontext ist alles

Technische Hochschule mannheim

Der Assistent sieht nur, was er als Kontext bekommt

Geöffnete Dateien und aktuelle Auswahl fließen oft mit ein

Gezielt referenzieren

Dateien/Symbole erwähnen (z. B. #Datei, @-Referenzen – je nach Tool)

Projektregeln

Konventionen als Regeldatei hinterlegen (z. B. Stil, Sprache, Verbote)

Fokus schaffen

Unnötige Dateien schließen; nur relevanten Code zeigen

Technische Hochschule mannheim

Typische Aufgaben für KI

1

Erstellen

Boilerplate

Funktionen & Klassen

Tests generieren

Datenmodelle

2

Verstehen

Code erklären

Fehler suchen (Debug)

Doku erzeugen

Fremdcode einordnen

3

Verbessern

Refactoring

Optimierung

Sprache übersetzen

Review-Hinweise

Übung · Funktion + Tests generieren

Technische Hochschule mannheim

Aufgabe

Lass dir per Chat eine kleine Funktion inkl. Unit-Tests erzeugen und führe die Tests aus.

Vorgehen

1 · Präzisen Prompt mit Signatur und Verhalten formulieren

2 · Vorschlag lesen und verstehen (nicht blind übernehmen)

3 · Tests generieren lassen und ausführen

4 · Bei Fehlern gezielt nachfassen (Fehlermeldung mitgeben)

Ergebnis

Eine getestete, verstandene Funktion – von dir verantwortet

Agent-Modus: mehrschrittige Aufgaben

Technische Hochschule mannheim

Was ist das?

Der Assistent plant und führt mehrere Schritte aus – über Dateien hinweg

Gut geeignet für

Gerüst anlegen, Umbenennungen, kleine Features, wiederkehrende Muster

Deine Rolle

Auftrag klar formulieren, Zwischenschritte prüfen, Änderungen freigeben

Sicherheit

In kleinen Schritten arbeiten, Versionskontrolle als Sicherheitsnetz nutzen

Technische Hochschule
mannheim

4 Qualität, Grenzen & Verantwortung

KI ist ein Werkzeug – nicht der Verantwortliche

Risiken & Grenzen

Technische Hochschule mannheim

Halluzinationen

Plausibel aussehender, aber falscher Code oder erfundene APIs

Sicherheitslücken

Generierter Code kann verwundbar oder unsicher sein

Lizenz & Urheberrecht

Unklare Herkunft von Vorschlägen

Veraltetes Wissen & Bias

Überholte Praktiken aus den Trainingsdaten

Automation Bias

Die Gefahr, Vorschlägen unkritisch zu vertrauen

Review-Pflicht: nie blind übernehmen

Technische Hochschule mannheim

Verstehen vor Übernehmen

Nur Code einchecken, den du selbst erklären kannst

Testen

Automatisierte Tests bleiben unverzichtbar – gerade bei KI-Code

Kritisch lesen

Auf Sicherheit, Randfälle und unnötige Komplexität achten

Verantwortung bleibt bei dir

Nicht das Werkzeug haftet für das Ergebnis, sondern du

Datenschutz, IP & Prüfungsregeln

Technische Hochschule mannheim

Keine Geheimnisse teilen

Keine Passwörter, Schlüssel oder personenbezogenen Daten in Prompts

Firmen-/Hochschulrichtlinien beachten

Freigegebene Tools und Datenschutz-Einstellungen verwenden

Prüfungs- & Abgaberegeln

Klären, ob und wie KI bei Übungen, Hausarbeiten und Klausuren erlaubt ist

Transparenz

KI-Nutzung kennzeichnen, wenn es gefordert ist – Plagiat vermeiden

KI und Git zusammen nutzen

Technische Hochschule mannheim

Kleine, nachvollziehbare Commits

KI-Änderungen in kleinen Schritten übernehmen und committen

Diffs reviewen

Jede KI-Änderung im Diff prüfen, bevor sie eingecheckt wird

Commit-Nachrichten

KI kann Entwürfe liefern – Inhalt bleibt deine Verantwortung

Sicherheitsnetz

Branch pro Experiment; jederzeit zurückrollbar

Technische Hochschule
mannheim

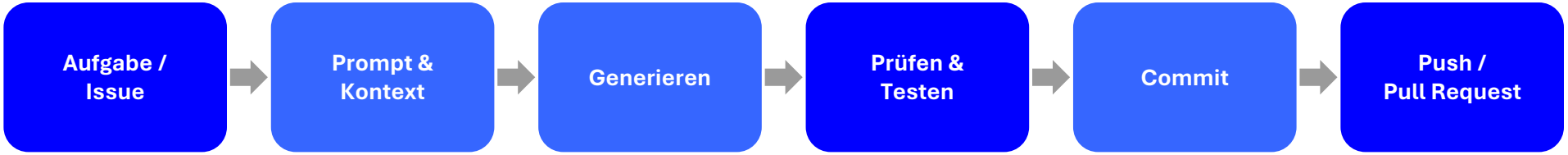
5 Workflow & Praxis

Von der Aufgabe bis zum Commit

Ein durchgängiger KI-Coding-Workflow

Technische Hochschule mannheim

„Prüfen & Testen“ ist der wichtigste Schritt – hier bleibt der Mensch verantwortlich.



Danach beginnt der Zyklus für die nächste Teilaufgabe von vorn.

Übung · Kleine Aufgabe von A bis Z

Technische Hochschule mannheim

Aufgabe

Setze ein kleines Feature komplett um: von der KI-Hilfe bis zum Commit.

Schritte

1 · Feature-Branch in VS Code anlegen

2 · Mit präzisem Prompt Code + Tests generieren lassen

3 · Verstehen, anpassen, Tests ausführen

4 · Diff prüfen, sinnvoll committen

5 · Pushen und (optional) Pull Request öffnen

Ergebnis

Ein getestetes, verstandenes und sauber versioniertes Feature

Technische Hochschule mannheim

Best Practices

1

Klug prompten

Kontext & Ziel geben

Klein iterieren

Beispiele nennen

2

Immer prüfen

Verstehen & testen

Diffs lesen

Sicherheit beachten

Nichts blind vertrauen

3

Sauber arbeiten

Kleine Commits

Keine Geheimnisse

Regeln beachten

Du bleibst verantwortlich

Zusammenfassung

Technische Hochschule mannheim

VS Code bündelt Editor, Terminal, Git und KI an einem Ort

KI-Assistenten sagen plausible Muster vorher – Kontext entscheidet

Drei Modi: Vervollständigung, Chat und Agent gezielt einsetzen

Gute Prompts: Kontext, Präzision, Iteration, Zerlegung

Immer prüfen und testen – die Verantwortung bleibt beim Menschen

Datenschutz und Prüfungsregeln beachten

Mit Git kleine, nachvollziehbare Schritte – KI-Diffs reviewen

Ressourcen (Stand Juli 2026)

Technische Hochschule mannheim

VS Code – Dokumentation: code.visualstudio.com/docs

GitHub Copilot – Pläne & Doku: github.com/features/copilot

GitHub Student Developer Pack: education.github.com/pack

Claude: claude.com · OpenAI Codex: openai.com/codex

Cursor: cursor.com · VS-Code-Marktplatz für Erweiterungen

Hinweis: Preise und Gratis-Angebote ändern sich häufig – bitte aktuell prüfen

Technische Hochschule
mannheim

Viel Erfolg beim Coden mit KI!

Fragen, Ausprobieren und Diskussion willkommen