

*TechLab*

# Einfacher Coden mit ChatGPT und Co.

```
18 <link rel="pingback" href="https://www.fruitful-get.com/trackback.php?url=" . $url . ">
19 <link rel="pingback" href="https://www.fruitful-get.com/trackback.php?url=" . $url . ">
20 <link rel="pingback" href="https://www.fruitful-get.com/trackback.php?url=" . $url . ">
21 <script src="https://www.fruitful-get.com/js/jquery.min.js"></script>
22 <script src="https://www.fruitful-get.com/js/jquery.cookie.js"></script>
23 <script src="https://www.fruitful-get.com/js/jquery.validate.js"></script>
24 <script src="https://www.fruitful-get.com/js/jquery.validate.unobtrusive.js"></script>
25 <script src="https://www.fruitful-get.com/js/jquery.validate.unobtrusive.js"></script>
26 <script src="https://www.fruitful-get.com/js/jquery.validate.unobtrusive.js"></script>
27 <script src="https://www.fruitful-get.com/js/jquery.validate.unobtrusive.js"></script>
28 <script src="https://www.fruitful-get.com/js/jquery.validate.unobtrusive.js"></script>
29 <script src="https://www.fruitful-get.com/js/jquery.validate.unobtrusive.js"></script>
30 <script src="https://www.fruitful-get.com/js/jquery.validate.unobtrusive.js"></script>
31 <script src="https://www.fruitful-get.com/js/jquery.validate.unobtrusive.js"></script>
```

# Agenda

- Kurze Einführung in KI-gestützte Programmierhilfen:
  - Aufzeigen grundlegender Einsatzmöglichkeiten
  - Generelle Vorteile & Grenzen
  - Übersicht relevanter Tools (z.B. ChatGPT, GitHub Copilot)
- Praktische Übungen:
  - Programmieraufgaben für ChatGPT und Co.
  - Gemeinsame Besprechung der Lösungen und Verbesserungsvorschläge
- Offene Fragerunde:
  - Erfahrungen, Herausforderungen und Erkenntnisse
  - Zukunft von KI in der Programmierung

# Einsatzmöglichkeiten

## Code-Generierung

Die KI-gestützte Code-Generierung kann natürlich-sprachige Anfragen, welche die gewünschte Funktionalität beschreiben, in ausführbaren Code umwandeln.

Das Modell versteht die bereitgestellten Vorschläge und bietet dann kontextuell relevanten Code an, wobei etablierte Programmiermuster berücksichtigt werden.

Das funktioniert je nach Modell und Komplexität der Anfragen teilweise sehr gut, teilweise erst nach vielen Iterationen, teilweise gar nicht.

# Einsatzmöglichkeiten

## **Code-Vervollständigung**

Die KI-gestützte Code-Vervollständigung kann mit der Auto-Completion in Word und anderen Schreibprogrammen verglichen werden.

Sie beinhaltet das Vorschlagen von Code-Snippets, während Programmcode eingegeben wird, indem sie den als nächstes wahrscheinlichsten Code basierend auf dem Kontext vorhersagt.

Dies kann den manuellen Eingabeaufwand reduzieren, die Konsistenz des Codes verbessern und Fehler minimieren, was besonders bei repetitiven Aufgaben wertvoll ist.

# Einsatzmöglichkeiten

## Code-Übersetzung

Die Code-Übersetzung bezieht sich auf das Umwandeln des Quellcodes in die gewünschte Programmiersprache.

Sie ermöglicht es, Code nahtlos aus verschiedenen Software-Umgebungen und/oder Development-Teams zu harmonisieren und integrieren.

Damit kann auch der Aufwand und die Kosten für das Aktualisieren von Legacy-Codes, die mittels veralteter Programmiersprachen oder Programmierpraktiken geschrieben wurden, reduziert werden.

# Einsatzmöglichkeiten

## **Code-Review/-Verständnis**

Das Code-Review ist ein Bewertungsprozess, der die Fähigkeit der KI nutzt, um die Codequalität durch Identifizierung und Korrektur von Fehlern zu erhöhen.

Zudem können KI-Programmertools auch Kommentare und Code-Erklärungen in natürlicher Sprache generieren.

Dies erleichtert das Verständnis des Überprüfungsprozesses erheblich, was besonders für Entwickelnde vorteilhaft ist, die ein detailliertes Feedback suchen, um ihre Programmierfähigkeiten zu verbessern.

# Vorteile & Grenzen

## Vorteile:

- Beschleunigung des Entwicklungsprozesses
- Effiziente Fehlersuche und –bereinigung
- Niederschwelliger Einstieg in die Welt der Programmierung für Programmier-Laien

## Nachteile:

- Besonders bei komplexen Codes kann die KI einem nicht immer weiterhelfen
- KI hat (noch) kein umfassendes Verständnis des Kontextes und der Spezifitäten eines Projektes
- Gefahr der «Überabhängigkeit» von den KI-Tools -> Kreativität und eigenes Denken könnte verloren gehen

-> “Overconfident junior developers with deep historical knowledge and a shallow understanding of modern standards”

# Relevante KI-Programmierhilfen

- ChatGPT
- GitHub Copilot
- Cody AI
- Tabnine
- Amazon CodeWhisperer
- Codium



# Relevante KI-Programmierhilfen

## ChatGPT



### Ressourcen:

- Python lernen mit ChatGPT: <https://hub.hslu.ch/informatik/python-lernen-mit-chatgpt-ein-selbstversuch/>
- Programmieren mit ChatGPT (1): <https://ankhlab.de/glossar/programmieren-mit-chatgpt-eine-anleitung-fuer-entwickler/>
- Programmieren mit ChatGPT (2): <https://www.computerbild.de/artikel/cb-Tipps-Software-Programmieren-mit-ChatGPT-Ein-Erfahrungsbericht-was-geht-nicht-35164425.html>
- Wie man ChatGPT fürs Coding verwendet: <https://neuroflash.com/de/blog/wie-man-chatgpt-fuer-coding-verwendet-mit-beispielen/>

# Relevante KI-Programmierhilfen

## GitHub Copilot



### Ressourcen:

- Schnellstart: <https://docs.github.com/de/copilot/quickstart>
- Erste Schritte: <https://docs.github.com/de/copilot/using-github-copilot/getting-started-with-github-copilot>
- Einführung: <https://learn.microsoft.com/de-de/training/modules/introduction-to-github-copilot/>
- GitHub Copilot Erfahrungsbericht nach einem Jahr: <https://hellocoding.de/blog/coding-language/allgemein/github-copilot>

# Relevante KI-Programmierhilfen

## Cody AI



### **Ressourcen:**

- Getting Started: <https://intercom.help/cody/en/collections/3833053-getting-started>
- Cody Terminologien erklärt: <https://meetcody.ai/de/blog/cody-terminologien-101/>
- Wie man einen AI-Website-Chatbot erstellt Widgets: <https://meetcody.ai/de/blog/erstellung-eines-ki-reisefuehrers-mit-cody-ft-benutzerdefinierte-widgets/>

# Relevante KI-Programmierhilfen

## Tabnine



### Ressourcen:

- Wie funktioniert Tabnine: <https://www.dev-insider.de/wie-funktoniert-tabnine-a-2b5fe5862c32952b3bf89a501b889a31/>
- Tabnine Docs: <https://docs.tabnine.com/saas/info>
- Become a Tabnine expert in 40 minutes: <https://www.youtube.com/live/XXERCwezdsQ?si=OVobAOr3AbenQOei>

# Relevante KI-Programmierhilfen

## Amazon CodeWhisperer



### Ressourcen:

- Handbuch: [https://docs.aws.amazon.com/de\\_de/toolkit-for-vscode/latest/userguide/codewhisperer.html](https://docs.aws.amazon.com/de_de/toolkit-for-vscode/latest/userguide/codewhisperer.html)
- Getting Started: <https://cmani.medium.com/getting-started-with-amazon-codewhisperer-b5d82d5075d3>
- FAQ: <https://aws.amazon.com/de/codewhisperer/faqs/>
- AWS CodeWhisperer. Besser als ChatGPT?: [https://youtu.be/bMj6H7jZ098?si=f9VlfYzjTu7bnA\\_2](https://youtu.be/bMj6H7jZ098?si=f9VlfYzjTu7bnA_2)

# Relevante KI-Programmierhilfen

## Codium



### **Ressourcen:**

- Nützliche Ressourcen: <https://www.codium.ai/resources/>
- Codium Visual Studio Installation: [https://codeium.com/vscode\\_tutorial](https://codeium.com/vscode_tutorial)
- FAQ: <https://www.codium.ai/#faq>

# Praktische Übungen

- Online Code Editor für ganz viele Programmiersprachen fürs Üben:
  - <https://onecompiler.com/languages>

# Praktische Übungen

- Problem als Programmier-AnfängerIn...

...wo fange ich an?

- Input ChatGPT:

*«Ich möchte ganz pragmatisch und praktisch Programmieren lernen und weiss nicht, wo ich anfangen soll. Ich habe keine Programmiererfahrung, kenne die Konzepte und Theorien dahinter nicht, habe keine Idee, was ich mit Programmieren umsetzen kann und weiss auch nicht, welche Tools ich brauche, um mit dem Programmieren zu beginnen. Kannst du mir einen strukturierten, von den Basics bis zu fortgeschrittenen Konzepten umfassenden Kurs für einen Programmierlaien wie mich geben, der mich Schritt für Schritt mittels Rückfragen, Beispielen und praxisnahen, im Alltag anwendbaren Übungen durch den Programmierschungel lotst? Stelle mir nach jedem deiner Outputs eine oder mehrere Folgefragen, welche die jeweils behandelten Themen vertiefen, plus eine Option, direkt zum nächsten Kapitel des Kurses zu springen. Ok?»*



# Praktische Übungen

- Übungen für AnfängerInnen:
  - Basics:
    - <https://onecompiler.com/challenges/3w7dby3mt/beginners-coding-challenge>
  - Strings:
    - <https://onecompiler.com/challenges/3w8xvfbtb/strings-challenge>
  - Rechenoperationen:
    - <https://onecompiler.com/challenges/3xv4whax2/basic-arithmetic-operations>
  - If-Else:
    - <https://onecompiler.com/challenges/3xveaz8es/if-else-set-1>
  - Schleifen:
    - <https://onecompiler.com/challenges/3xvgabhq3/loops-1>

# Praktische Übungen

- Übungen für Fortgeschrittene:
  - <https://onecompiler.com/challenges/3w9us3eby/intermediate-coding-challenge>
- Viele weitere Übungen:
  - <https://onecompiler.com/challenges>

# Praktische Übungen Webdesign

- Hilfe beim Debugging & Codeinterpretation
  - Nutze ChatGPT, um Fehler im Code zu finden und um dir den Code Zeile für Zeile erklären zu lassen.

# Praktische Übungen Webdesign

<https://jsfiddle.net/>

- **Übungsvorschlag 1**
  - Erstelle eine Webseite, die einen Timer zeigt.
  - Die Webseite sollte einen «Start»-Button (Starten des Timers), einen «Stop»-Button (Stoppen der Timers), und einen «Reset»-Button (Timer auf Null stellen) beinhalten.
  - Die Webseite soll mittels HTML, CSS und JavaScript programmiert werden.

# Praktische Übungen Webdesign

<https://jsfiddle.net/>

- **Übungsvorschlag 2**
  - Erstelle eine Website mit HTML, CSS und JavaScript, die einen Schreibgeschwindigkeitstester anbietet.
  - Die Website muss Text für den Test, einen Timer, ein Eingabefeld, einen «Pause»-Button, einen «Zurücksetzen»-Button und eine Zeichenzählung enthalten.
  - Der Timer sollte starten, wenn du mit dem Tippen beginnst, und anhalten, entweder wenn du auf den Pause-Button klickst oder den Zurücksetzen-Button klickst oder wenn du den vorgegebenen Text vollständig getippt hast.

# Praktische Übungen Webdesign

<https://jsfiddle.net/>

- **Übungsvorschlag 3**
  - Erstelle eine Portfolio-Website mit HTML, CSS und JavaScript, die mit Tailwind gestaltet ist.
  - Die Website sollte ein großes Header-Bild oben zusammen mit dem Namen "Mein Portfolio" zeigen, ein Drei-Spalten-Layout mit Inhaltskarten darunter und dann ein Einspalten-Layout mit einem kurzen Lebenslauf, einschließlich Dummy-Text und Bildern für die Seite.

# Offene Fragerunde

- Erfahrungen, Herausforderungen und Erkenntnisse
- Zukunft von KI in der Programmierung

*TechLab*

# Einfacher Coden mit ChatGPT und Co.

*Danke  
für Eure Teilnahme!*