



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Kapitel 01

MODERNE ENTWICKLUNGSWERKZEUGE

Heutige Lernziele

Am Ende dieser Vorlesung werden Sie:

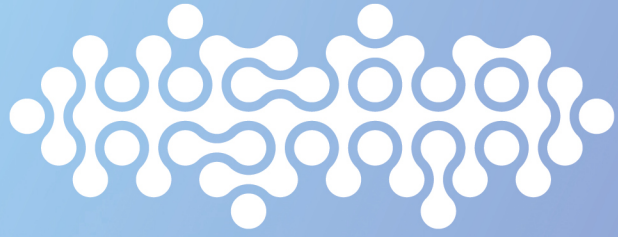
1.  **Verstehen**, was Repositories sind und warum sie wichtig sind
2.  **Erstellen** Ihres GitHub-Kontos und ersten Repositories
3.  **Installieren** von Git und professionelle Konfiguration
4.  **Klonen** eines Repositories auf Ihren lokalen Rechner
5.  **Einrichten** von VS Code mit wichtigen Erweiterungen
6.  **Konfigurieren** von GitHub Copilot für KI-unterstütztes Programmieren
7.  **Installieren** von Python auf Ihrem System

Warum diese Werkzeuge wichtig sind

In der professionellen Softwareentwicklung:

-  **Jedes** Tech-Unternehmen nutzt Versionskontrolle (Git)
-  **GitHub** hostet über 100 Millionen Repositories
-  **VS Code** ist die weltweit beliebteste IDE
-  **KI-Assistenten** wie Copilot werden zum Standard

Ihr GitHub-Profil = Ihre professionelle Identität



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Repositories verstehen

1

Was ist ein Repository?

Ein **Repository (Repo)** ist ein zentraler Speicherort, der enthält:

-  **Quellcode-Dateien** - Ihre Programme und Skripte
-  **Dokumentation** - README, Wikis, technische Docs
-  **Konfigurationsdateien** - Einstellungen für Ihre Umgebung
-  **Testdateien** - Automatisierte Tests für Qualität
-  **Verlaufsmetadaten** - WER hat WAS, WANN und WARUM geändert

Kurze Frage

Haben Sie schon mal Dateien so benannt?

```
hausarbeit.docx  
hausarbeit_v2.docx  
hausarbeit_v2_final.docx  
hausarbeit_v2_final_korrigiert.docx  
hausarbeit_v2_final_korrigiert_wirklich_final.docx  
hausarbeit_v2_final_korrigiert_wirklich_final_ECHT_JETZT.docx
```

Heben Sie die Hand, wenn Ihnen das bekannt vorkommt! 

Probleme mit manueller Versionierung

- ✗ **Keine klare Historie** - Welche Version hatte den brillanten Absatz, den Sie gelöscht haben?
- ✗ **Speicherverschwendung** - Jede Kopie dupliziert 99% des unveränderten Inhalts
- ✗ **Kollaborations-Albtraum** - Wie führen Sie Änderungen mehrerer Autoren zusammen?
- ✗ **Keine Verantwortlichkeit** - Wer hat die Änderung gemacht, die alles kaputt gemacht hat?
- ✗ **Kein Vergleich** - Wie sehen Sie genau, was sich geändert hat?

Repositories lösen ALL diese Probleme!

Lokale vs. Remote Repositories

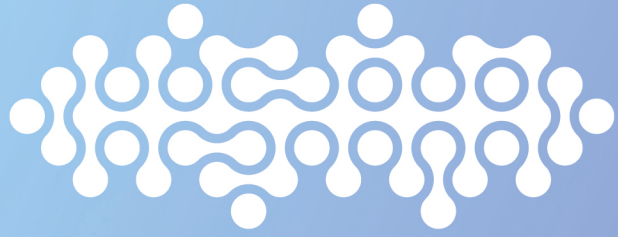
Lokales Repository

- Liegt auf IHREM Computer
- Offline arbeiten
- Tägliche Entwicklung
- Volle Kontrolle

Remote Repository

- Gehostet auf Servern (GitHub)
- Über Internet erreichbar
- Team-Synchronisation
- Backup & Kollaborations-Hub

Workflow: Lokal ändern → Zum Remote pushen → Team pullt Änderungen



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Warum GitHub?

2

GitHub: Der Industriestandard

GitHub Statistiken:

-  **100+ Millionen** Repositories
-  **100+ Millionen** Entwickler weltweit
-  Genutzt von **Google, Microsoft, Netflix, Amazon...**

Wichtige Erkenntnis:

*Ein gut gepflegtes GitHub-Profil ist **wertvoller** als ein polierter Lebenslauf.*

Warum? Weil Code nicht lügt.

Was Ihr GitHub-Profil zeigt

Arbeitgeber können sehen:

1.  **Code-Qualität** - Ist er sauber, dokumentiert, folgt Best Practices?
2.  **Konsistenz** - Programmieren Sie regelmäßig oder nur vor Deadlines?
3.  **Zusammenarbeit** - Wie interagieren Sie mit anderen Entwicklern?
4.  **Problemlösung** - Welche Probleme lösen Sie? Wie?
5.  **Wachstum** - Wie haben sich Ihre Fähigkeiten entwickelt?

Ihr GitHub = Ihr lebendiges Portfolio

GitHub vs. GitLab: Kurzvergleich

Aspekt	GitHub	GitLab
Marktanteil	~90% bei Open Source	Wachsend aber kleiner
Community	Riesig, sehr aktiv	Kleiner
Job-Relevanz	Fast universelle Anforderung	Weniger verbreitet
Studenten-Vorteile	Umfassendes Paket	Verfügbar
Lernkurve	Besser für Anfänger	Komplexer

Unsere Wahl: GitHub - Industriestandard, bessere Karrierevorbereitung



Hands-On: GitHub-Konto erstellen

Besuchen Sie github.com und registrieren Sie sich - wählen Sie Ihren Benutzernamen weise!

✓ Gut:

- maria-schmidt
- dev-alex-chen
- sarah-k-developer

✗ Vermeiden:

- c00ld3v420
- xxxprogrammer
- temporary_account

Dies wird Ihre professionelle Identität!

GitHub Studenten-Vorteile

Beantragen Sie bei education.github.com

Sie erhalten **KOSTENLOS**:

-  **GitHub Pro** - Unbegrenzte private Repositories
-  **GitHub Copilot** - KI-Coding-Assistent (Wert \$100+/Jahr!)
-  **Cloud-Credits** - Heroku, DigitalOcean Hosting
-  **JetBrains IDEs** - Professionelle Entwicklungswerkzeuge
-  **Und vieles mehr** - Entwicklertools im Wert von Tausenden!

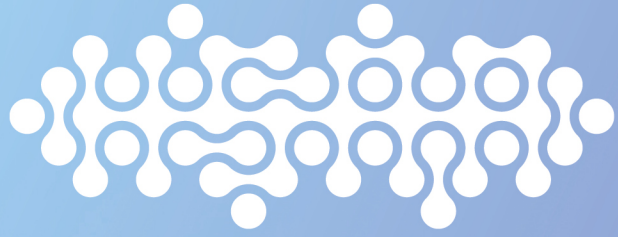
Aktion: Noch heute mit Ihrer Hochschul-E-Mail beantragen!



Hands-On: Erstes Repository erstellen

1. Klicken Sie auf das "+" Symbol → **"New repository"**
2. Konfigurieren Sie Ihr Repository:
 - Repository name: `MyFirstRepo`
 - Description: `My first repository for learning Git`
 - ☒ Public ☒ Add README ☒ Add .gitignore ☒ MIT License
3. Klicken Sie auf **"Create repository"**

Herzlichen Glückwunsch! Sie sind jetzt offiziell auf GitHub! 🎉



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Versionskontrolle mit Git

3

Warum Versionskontrolle Pflicht ist

Versionskontrolle bietet:

1. 📄 **Historische Aufzeichnung** - Jede Änderung mit wer, wann, warum
2. 👤 **Parallele Entwicklung** - Mehrere Entwickler, keine Störungen
3. 🧪 **Experimentiersicherheit** - Dinge ausprobieren, bei Bedarf verwerfen
4. ⏮ **Rollback-Fähigkeit** - Fehler sofort rückgängig machen
5. 👁 **Code Review** - Qualitätskontrolle vor Integration
6. 📖 **Lebende Dokumentation** - Historie erklärt Entscheidungen

In der modernen Entwicklung ist Versionskontrolle NICHT optional.

Git: Der universelle Standard

Genutzt von:

-  Google, Microsoft, Facebook, Netflix
-  Linux Kernel, Python, React
-  Jedes große Informatik-Programm

Warum Git?

-  Verteilte Architektur
-  Mächtiges Branching und Merging
-  Extrem schnelle Operationen
-  Kryptographische Integrität

Grundlegende Git-Konzepte

Konzept	Beschreibung
Repository	Komplettes Projekt mit gesamter Historie
Commit	Schnappschuss Ihres Projekts zu einem Zeitpunkt
Branch	Parallele Version zum Experimentieren
Merge	Zusammenführen von Änderungen verschiedener Branches
Clone	Lokale Kopie eines Remote Repositories erstellen
Push	Lokale Änderungen zum Remote hochladen
Pull	Remote-Änderungen lokal herunterladen



Hands-On: Git installieren

Windows:

1. Download von git-scm.com
2. Installer mit empfohlenen Einstellungen ausführen
3. Überprüfen: `git --version`

Linux (Ubuntu):

```
# Oft vorinstalliert! Erst prüfen:  
git --version
```

```
# Falls nicht installiert:  
sudo apt update  
sudo apt install git-all
```



Git mit Ihrer Identität konfigurieren

Wichtige Konfiguration:

```
git config --global user.name "Ihr Vollständiger Name"  
git config --global user.email "ihre.email@hochschule.de"  
git config --global init.defaultBranch main
```

Wichtig:

- Verwenden Sie Ihren **echten Namen**
- Verwenden Sie die **gleiche E-Mail** wie bei GitHub
- Dies erscheint in ALLEN Ihren Commits!



Ihr Repository klonen

Schritt 1: Gehen Sie zu Ihrem `MyFirstRepo` auf GitHub

Schritt 2: Klicken Sie auf den grünen **"Code"** Button → HTTPS URL kopieren

Schritt 3: Im Terminal zum Projektordner navigieren:

```
cd ~/Documents/Projects      # Linux  
cd C:\Users\IhrName\Documents\Projects  # Windows
```

Schritt 4: Klonen!

```
git clone https://github.com/IHR_USERNAME/MyFirstRepo.git  
cd MyFirstRepo
```



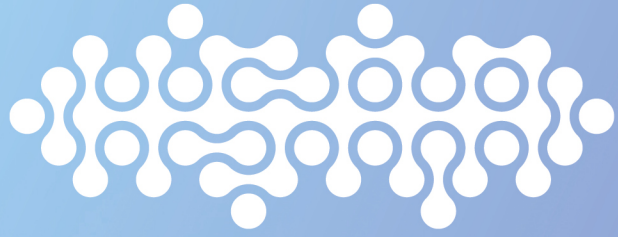
Kurze Überprüfung

Was ist gerade passiert?

1. Sie haben ein Remote Repository auf GitHub erstellt
2. Sie haben es auf Ihren lokalen Rechner geklont
3. Jetzt haben Sie ZWEI Kopien, die synchronisiert bleiben!

Probieren Sie:

```
ls          # Dateien anzeigen  
git status  # Repository-Status anzeigen  
git log     # Commit-Historie anzeigen
```



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Visual Studio Code

4

Warum eine IDE wichtig ist

IDE = Integrierte Entwicklungsumgebung

Ohne IDE (Notepad):

- Kein Syntax-Highlighting
- Keine Code-Vervollständigung
- Kein Debugging
- Keine Git-Integration

Mit IDE (VS Code):

-  All das + mehr!
-  IntelliSense-Vorschläge
-  Eingebauter Debugger
-  Tausende Erweiterungen

Warum VS Code?

Stack Overflow Developer Survey: Beliebteste IDE!

- ✓ **Kostenlos & Open Source** - Keine Lizenzkosten
- ✓ **Plattformübergreifend** - Windows, macOS, Linux
- ✓ **Leichtgewichtig** - Schneller Start, wenig Speicher
- ✓ **Erweiterbar** - 30.000+ Erweiterungen
- ✓ **GitHub-Integration** - Native Copilot-Unterstützung
- ✓ **Industriestandard** - Von großen Unternehmen genutzt



VS Code installieren

Windows:

1. Download von code.visualstudio.com
2. Installer mit diesen Optionen ausführen:
 - ☒ "Mit Code öffnen" zum Kontextmenü hinzufügen
 - ☒ Als Editor für unterstützte Dateien registrieren
 - ☒ **Zu PATH hinzufügen** (wichtig!)

Linux:

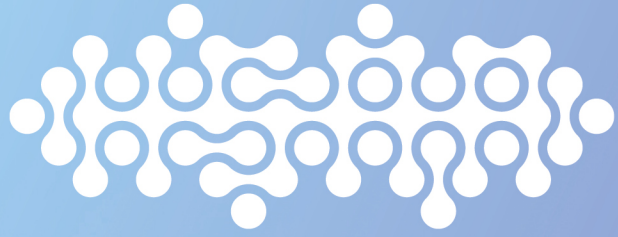
```
sudo snap install --classic code
```

Wichtige Erweiterungen

Installieren Sie diese Erweiterungen (Strg+Umschalt+X):

Erweiterung	Zweck
Python (Microsoft)	Python-Sprachunterstützung
Ruff	Schneller Python-Linter & Formatter
GitLens	Erweiterte Git-Integration
Live Server	Lokaler Webserver

Suchen → Installieren → Fertig!



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

GitHub Copilot

5

Was ist GitHub Copilot?

KI-gestützte Code-Vervollständigung, trainiert mit Milliarden Codezeilen

Copilot kann:

-  Ganze Funktionen aus Kommentaren vorschlagen
-  Zeilen während des Tippens vervollständigen
-  Wiederholende Muster generieren
-  Natürliche Sprache in Code übersetzen
-  Tests und Dokumentation schreiben

Denken Sie: Ein erfahrener Programmierer, der 24/7 mit Ihnen programmiert

Warum Copilot wichtig für das Lernen ist

Beschleunigtes Lernen

- Idiomatische Implementierungen sofort sehen
- Korrekte Syntax durch Beispiele lernen
- Auf Problemlösung fokussieren, nicht Syntax

Praxisrelevante Fähigkeit

- KI-unterstütztes Coding ist Industriestandard
- Microsoft, Google, Amazon fördern es
- Essenzielle professionelle Fähigkeit

Copilot-Beispiel

Sie schreiben:

```
# Create a function that reads a CSV file and returns a pandas DataFrame
```

Copilot schlägt vor:

```
def read_csv_file(filepath: str) -> pd.DataFrame:  
    """Read a CSV file and return as DataFrame."""  
    return pd.read_csv(filepath)
```

Magie? Nein - Mustererkennung aus Milliarden Beispielen!

Wichtig: Sie bleiben der Programmierer!

Copilot ist ein WERKZEUG, kein Ersatz.

Sie müssen:

-  **Verstehen**, was der Code vorschlägt
-  **Überprüfen** auf Korrektheit
-  **Gründlich testen**
-  **Architektonische Vision** beibehalten
-  **Grenzfälle** berücksichtigen

Akzeptieren Sie niemals Code, den Sie nicht verstehen!

Wobei Copilot glänzt vs. Schwierigkeiten hat

✓ Glänzt bei:

- Boilerplate-Code
- Klar definierten Problemen
- Populären Sprachen/Frameworks
- Unit-Tests
- Dokumentation

✗ Schwierigkeiten bei:

- Neuartigen Algorithmen
- Ihrer spezifischen Geschäftslogik
- Komplexen Architekturentscheidungen
- Sicherheitskritischem Code
- Vollständigem Kontextverständnis



Copilot in VS Code installieren

1. Öffnen Sie **Erweiterungen** (Strg+Umschalt+X)
2. Suchen Sie **"GitHub Copilot"**
3. Klicken Sie auf **Installieren**
4. Melden Sie sich mit Ihrem GitHub-Konto an
5. Autorisieren Sie bei Aufforderung

Optional aber empfohlen: Installieren Sie auch **"GitHub Copilot Chat"**

Kostenlos für Studenten! (GitHub Education Vorteile)

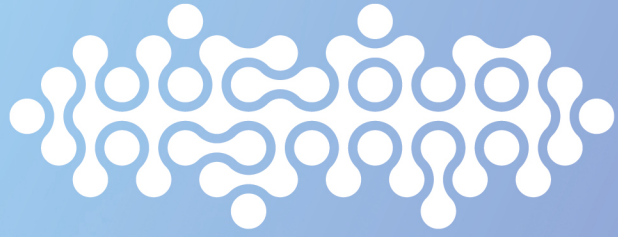
Demnächst: Cursor IDE

Cursor = VS Code Fork, gebaut um KI

Was es besonders macht:

-  KI versteht Ihr gesamtes Projekt
-  Natürlichsprachliche Code-Bearbeitung
-  Multi-Datei KI-Vorschläge
-  Fortgeschrittenes KI-gestütztes Refactoring

Wir werden Cursor später im Kurs erkunden nach den Grundlagen.



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Python installieren

6

Python-Installation: Windows

Schritt 1: Download von python.org

Schritt 2: Installer ausführen - **KRITISCHE EINSTELLUNGEN:**

-  **"Add python.exe to PATH"** ← Wichtig!
-  **"Install launcher for all users"**

Schritt 3: Überprüfen:

```
python --version    # Sollte Python 3.12.x zeigen  
pip --version       # Paket-Installer
```

Python-Installation: Linux

Prüfen ob bereits installiert:

```
python3 --version
```

Falls nicht installiert oder veraltet:

```
sudo add-apt-repository ppa:deadsnakes/ppa  
sudo apt update  
sudo apt install python3.12 python3.12-venv python3.12-dev
```

Überprüfen:

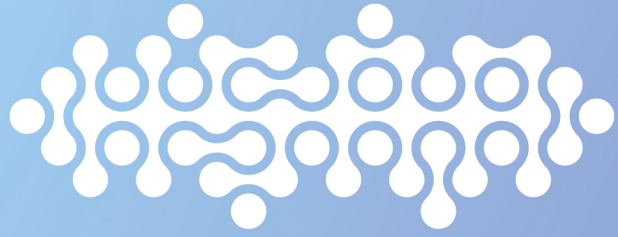
```
python3 --version  
pip3 --version
```

Python-Befehlsreferenz

Kursmaterialien	Windows	Linux
<code>python</code>	<code>python</code>	<code>python3</code>
<code>pip</code>	<code>pip</code>	<code>pip3</code>
<code>python --version</code>	<code>python --version</code>	<code>python3 --version</code>

Linux-Tipp: Alias erstellen:

```
echo "alias python='python3'" >> ~/.bashrc  
source ~/.bashrc
```



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Ihr erster Entwicklungs- Workflow

7



Lass uns Code schreiben!

Schritt 1: Öffnen Sie Ihr Repository in VS Code: `code`
`~/Documents/Projects/MyFirstRepo`

Schritt 2: Erstellen Sie `hello_world.py` :

```
def greet_user(name: str) -> str:
    return f"Hallo, {name}! Willkommen bei Software Engineering."

if __name__ == "__main__":
    user_name = input("Geben Sie Ihren Namen ein: ")
    print(greet_user(user_name))
```



Testen und Committen

Schritt 3: Ihr Programm testen

```
python hello_world.py
```

Schritt 4: Committen und pushen

```
git status          # Änderungen sehen  
git add hello_world.py  # Datei stagen  
git commit -m "Add hello world program"  
git push origin main  # Zu GitHub pushen
```

Schritt 5: Überprüfen Sie GitHub - Ihr Code ist jetzt online! 🎉

Der moderne Entwicklungs-Workflow

Der Zyklus, den Sie täglich nutzen werden:

Schreiben → **Testen** → **Committen** → **Pushen** → **Kollaborieren** → **Deployen**

 *Iterieren und wiederholen*

Wichtige Erkenntnis: Jeder Schritt baut auf den heute gelernten Tools auf:

- VS Code + Copilot zum Schreiben
- Git für Commits
- GitHub für Push, Kollaboration, Deploy

Zusammenfassung: Heute installierte Tools

Tool	Zweck
GitHub-Konto	Professionelle Identität & Code-Hosting
Git	Versionskontrollsystem
VS Code	Entwicklungsumgebung
Copilot	KI-Coding-Assistent
Python	Programmiersprache

Sie haben jetzt ein professionelles Entwicklungs-Setup!

Wichtigste Erkenntnisse

1.  **Repositories** sind lebende Ökosysteme für Ihren Code
2.  **GitHub** ist Ihr professionelles Portfolio
3.  **Git** ermöglicht Kollaboration und Historienverfolgung
4.  **VS Code** ist Ihre Programmier-Kommandozone
5.  **Copilot** beschleunigt das Lernen (aber SIE bleiben der Programmierer)
6.  **Python** ist korrekt installiert und konfiguriert

Sie sind mit den gleichen Tools ausgestattet wie bei Google, Microsoft, Netflix!



Nächste Schritte

Vor der nächsten Vorlesung:

1. ☒ GitHub-Konto-Einrichtung abschließen
2. ☒ GitHub Education Vorteile beantragen
3. ☒ Ihr Repository klonen
4. ☒ Alle Tools funktionieren verifizieren
5. ☒ Das Vorlesungs-Quiz machen!

Quiz: Testen Sie Ihr Verständnis der heutigen Konzepte

Zusätzliche Ressourcen

Wichtige Lektüre:

-  [Pro Git Book](#) - Umfassende Git-Referenz
-  [GitHub Docs](#) - Offizielle Dokumentation
-  [VS Code Tipps](#)

Übung:

-  [Learn Git Branching](#) - Interaktives Lernen
-  [GitHub Learning Lab](#)



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Fragen?

VIELEN DANK