



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Chapter 01

MODERN DEVELOPMENT TOOLS

Today's Learning Objectives

By the end of this lecture, you will:

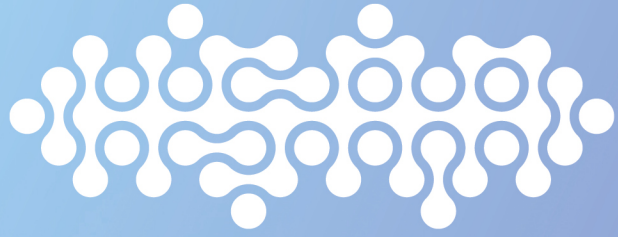
1.  **Understand** what repositories are and why they matter
2.  **Create** your GitHub account and first repository
3.  **Install** Git and configure it professionally
4.  **Clone** a repository to your local machine
5.  **Set up** VS Code with essential extensions
6.  **Configure** GitHub Copilot for AI-assisted coding
7.  **Install** Python properly on your system

Why These Tools Matter

In professional software development:

-  **Every** tech company uses version control (Git)
-  **GitHub** hosts 100+ million repositories
-  **VS Code** is the most popular IDE worldwide
-  **AI assistants** like Copilot are becoming standard

Your GitHub profile = Your professional identity



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Understanding Repositories

1

What is a Repository?

A **repository (repo)** is a central storage location containing:

-  **Source code files** - Your programs and scripts
-  **Documentation** - README, wikis, technical docs
-  **Configuration files** - Settings for your environment
-  **Test files** - Automated tests for quality
-  **History metadata** - WHO changed WHAT, WHEN, and WHY



Quick Question

Have you ever named files like this?

```
paper.docx  
paper_v2.docx  
paper_v2_final.docx  
paper_v2_final_reviewed.docx  
paper_v2_final_reviewed_actually_final.docx  
paper_v2_final_reviewed_actually_final_FOR_REAL.docx
```

Raise your hand if this looks familiar! 🙋

Problems with Manual Versioning

- ✗ **No clear history** - Which version had that brilliant paragraph you deleted?
- ✗ **Storage waste** - Each copy duplicates 99% of unchanged content
- ✗ **Collaboration nightmare** - How do you merge changes from multiple authors?
- ✗ **No accountability** - Who made that change that broke everything?
- ✗ **No comparison** - How do you see exactly what changed?

Repositories solve ALL of these problems!

Local vs. Remote Repositories

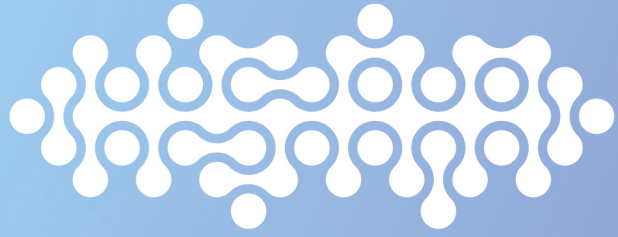
Local Repository

- Lives on YOUR computer
- Work offline
- Day-to-day development
- Full control

Remote Repository

- Hosted on servers (GitHub)
- Accessible via internet
- Team synchronization
- Backup & collaboration hub

Workflow: Make changes locally → Push to remote → Team pulls changes



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Why GitHub?

2

GitHub: The Industry Standard

GitHub Statistics:

-  **100+ million** repositories
-  **100+ million** developers worldwide
-  Used by **Google, Microsoft, Netflix, Amazon...**

Key Insight:

*A well-maintained GitHub profile is **more valuable** than a polished CV.*

Why? Because code doesn't lie.

What Your GitHub Profile Shows

Employers can see:

1.  **Code quality** - Is it clean, documented, following best practices?
2.  **Consistency** - Do you code regularly, or only before deadlines?
3.  **Collaboration** - How do you interact with other developers?
4.  **Problem-solving** - What problems do you tackle? How?
5.  **Growth** - How have your skills evolved over time?

Your GitHub = Your living portfolio

GitHub vs. GitLab: Quick Comparison

Aspect	GitHub	GitLab
Market Share	~90% of open source	Growing but smaller
Community	Massive, highly active	Smaller
Job Relevance	Near-universal requirement	Less common
Student Benefits	Comprehensive package	Available
Learning Curve	Better for beginners	More complex

Our choice: GitHub - Industry standard, better career preparation



Hands-On: Create Your GitHub Account

Visit github.com and sign up - choose your username wisely!

✓ Good:

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

✗ Avoid:

- c00ld3v420
- xxxprogrammer
- temporary_account

This becomes your professional identity!

GitHub Student Benefits

Apply at education.github.com

You'll receive **FREE**:

-  **GitHub Pro** - Unlimited private repositories
-  **GitHub Copilot** - AI coding assistant (worth \$100+/year!)
-  **Cloud credits** - Heroku, DigitalOcean hosting
-  **JetBrains IDEs** - Professional development tools
-  **Much more** - Developer tools worth thousands!

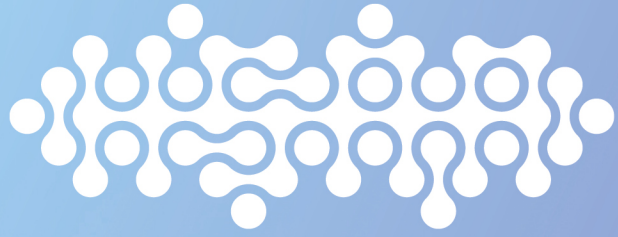
Action: Apply today with your university email!



Hands-On: Create Your First Repository

1. Click the "+" icon → **"New repository"**
2. Configure your repository:
 - Repository name: `MyFirstRepo`
 - Description: `My first repository for learning Git`
 - ☒ Public ☒ Add README ☒ Add .gitignore ☒ MIT License
3. Click **"Create repository"**

Congratulations! You're officially on GitHub! 🎉



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Version Control with Git

3

Why Version Control is Mandatory

Version control provides:

1. 📋 **Historical record** - Every change tracked with who, when, why
2. 👤 **Parallel development** - Multiple developers, no interference
3. 🧪 **Experimentation safety** - Try things, discard if needed
4. ⏮ **Rollback capability** - Undo mistakes instantly
5. 👁 **Code review** - Quality control before integration
6. 📖 **Living documentation** - History explains decisions

In modern development, version control is NOT optional.

Git: The Universal Standard

Used by:

-  Google, Microsoft, Facebook, Netflix
-  Linux kernel, Python, React
-  Every major CS program

Why Git?

-  Distributed architecture
-  Powerful branching and merging
-  Extremely fast operations
-  Cryptographic integrity

Core Git Concepts

Concept	Description
Repository	Complete project with all history
Commit	Snapshot of your project at a point in time
Branch	Parallel version for experimentation
Merge	Combining changes from different branches
Clone	Creating local copy of remote repository
Push	Uploading local changes to remote
Pull	Downloading remote changes to local

Hands-On: Installing Git

Windows:

1. Download from git-scm.com
2. Run installer with recommended settings
3. Verify: `git --version`

Linux (Ubuntu):

```
# Often pre-installed! Check first:  
git --version
```

```
# If not installed:  
sudo apt update  
sudo apt install git-all
```



Configure Git with Your Identity

Essential configuration:

```
git config --global user.name "Your Full Name"  
git config --global user.email "your.email@university.edu"  
git config --global init.defaultBranch main
```

Important:

- Use your **real name**
- Use the **same email** as your GitHub account
- This appears in ALL your commits!



Clone Your Repository

Step 1: Go to your `MyFirstRepo` on GitHub

Step 2: Click green **"Code"** button → Copy HTTPS URL

Step 3: In terminal, navigate to your projects folder:

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

Step 4: Clone!

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



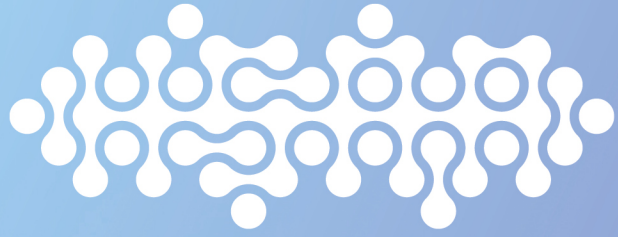
Quick Check

What just happened?

1. You created a remote repository on GitHub
2. You cloned it to your local machine
3. Now you have TWO copies that stay synchronized!

Try:

```
ls           # See your files
git status   # See repository status
git log      # See commit history
```



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Visual Studio Code

4

Why an IDE Matters

IDE = Integrated Development Environment

Without IDE (Notepad):

- No syntax highlighting
- No code completion
- No debugging
- No Git integration

With IDE (VS Code):

-  All of the above + more!
-  IntelliSense suggestions
-  Built-in debugger
-  Thousands of extensions

Why VS Code?

Stack Overflow Developer Survey: Most popular IDE!

- ✓ **Free & Open Source** - No licensing costs
- ✓ **Cross-platform** - Windows, macOS, Linux
- ✓ **Lightweight** - Fast startup, low memory
- ✓ **Extensible** - 30,000+ extensions
- ✓ **GitHub integration** - Native Copilot support
- ✓ **Industry standard** - Used by major companies



Installing VS Code

Windows:

1. Download from code.visualstudio.com
2. Run installer with these options:
 - ☒ Add "Open with Code" to context menu
 - ☒ Register as editor for supported files
 - ☒ **Add to PATH** (important!)

Linux:

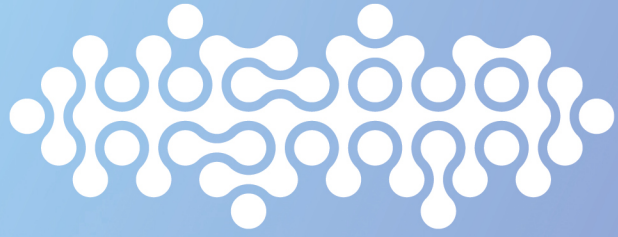
```
sudo snap install --classic code
```

Essential Extensions

Install these extensions (Ctrl+Shift+X):

Extension	Purpose
Python (Microsoft)	Python language support
Ruff	Fast Python linter & formatter
GitLens	Enhanced Git integration
Live Server	Local web server

Search → Install → You're ready!



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

GitHub Copilot

5

What is GitHub Copilot?

AI-powered code completion trained on billions of lines of code

Copilot can:

-  Suggest entire functions from comments
-  Complete lines as you type
-  Generate repetitive patterns
-  Translate natural language to code
-  Write tests and documentation

Think: An experienced programmer pair-programming with you 24/7

Why Copilot Matters for Learning

Accelerated Learning

- See idiomatic implementations instantly
- Learn correct syntax by example
- Focus on problem-solving, not syntax

Real-World Skill

- AI-assisted coding is industry standard
- Microsoft, Google, Amazon encourage it
- Essential professional skill

Copilot Example

You write:

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

Copilot suggests:

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

Magic? No - pattern recognition on billions of examples!

Important: You Remain the Programmer!

Copilot is a TOOL, not a replacement.

You must:

-  **Understand** the code it suggests
-  **Review** for correctness
-  **Test** thoroughly
-  **Maintain** architectural vision
-  **Consider** edge cases

Never accept code you don't understand!

What Copilot Excels At vs. Struggles With

✓ Excels At:

- Boilerplate code
- Well-defined problems
- Popular languages/frameworks
- Unit tests
- Documentation

✗ Struggles With:

- Novel algorithms
- Your specific business logic
- Complex architecture decisions
- Security-critical code
- Understanding full context



Installing Copilot in VS Code

1. Open **Extensions** (Ctrl+Shift+X)
2. Search "**GitHub Copilot**"
3. Click **Install**
4. Sign in with your GitHub account
5. Authorize when prompted

Optional but recommended: Also install "**GitHub Copilot Chat**"

Free for students! (GitHub Education benefits)

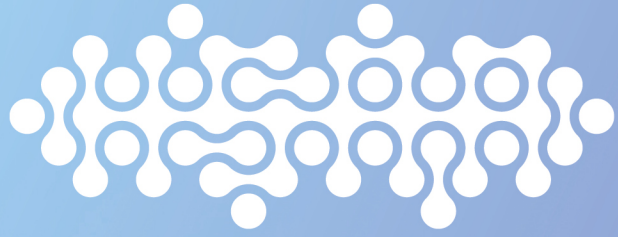
Coming Soon: Cursor IDE

Cursor = VS Code fork built around AI

What makes it different:

-  AI understands your entire project
-  Natural language code editing
-  Multi-file AI suggestions
-  Advanced AI-powered refactoring

We'll explore Cursor later in the course after mastering fundamentals.



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Installing Python

6

Python Installation: Windows

Step 1: Download from python.org

Step 2: Run installer - **CRITICAL SETTINGS:**

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

Step 3: Verify:

```
python --version    # Should show Python 3.12.x
pip --version       # Package installer
```

Python Installation: Linux

Check if already installed:

```
python3 --version
```

If not installed or outdated:

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

Verify:

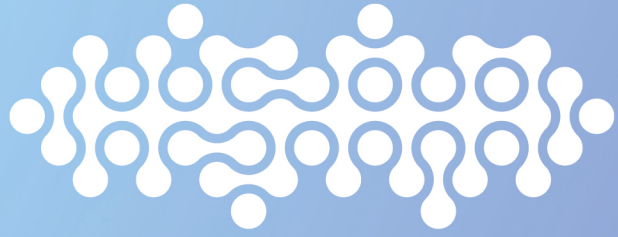
```
python3 --version  
pip3 --version
```

Python Command Reference

Course Materials	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 Tip: Create an alias:

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



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Your First Development Workflow

7



Let's Write Code!

Step 1: Open your repository in VS Code: `code ~/Documents/Projects/MyFirstRepo`

Step 2: Create `hello_world.py`:

```
def greet_user(name: str) -> str:
    return f"Hello, {name}! Welcome to software engineering."

if __name__ == "__main__":
    user_name = input("Enter your name: ")
    print(greet_user(user_name))
```

Test and Commit

Step 3: Test your program

```
python hello_world.py
```

Step 4: Commit and push

```
git status                # See what changed
git add hello_world.py    # Stage the file
git commit -m "Add hello world program"
git push origin main      # Push to GitHub
```

Step 5: Check GitHub - your code is now online! 🎉

The Modern Development Workflow

The cycle you'll use every day:

Write → **Test** → **Commit** → **Push** → **Collaborate** → **Deploy**

 *Iterate and repeat*

Key insight: Each step builds on the tools we learned today:

- VS Code + Copilot for writing
- Git for commits
- GitHub for push, collaborate, deploy

Summary: Tools We Installed Today

Tool	Purpose
GitHub Account	Professional identity & code hosting
Git	Version control system
VS Code	Development environment
Copilot	AI coding assistant
Python	Programming language

You now have a professional development setup!

Key Takeaways

1.  **Repositories** are living ecosystems for your code
2.  **GitHub** is your professional portfolio
3.  **Git** enables collaboration and history tracking
4.  **VS Code** is your programming command center
5.  **Copilot** accelerates learning (but YOU remain the programmer)
6.  **Python** is properly installed and configured

You're equipped with the same tools used at Google, Microsoft, Netflix!



Next Steps

Before next lecture:

1. ☒ Complete GitHub account setup
2. ☒ Apply for GitHub Education benefits
3. ☒ Clone your repository
4. ☒ Verify all tools are working
5. ☒ Take the lecture quiz!

Quiz: Test your understanding of today's concepts

Additional Resources

Essential Reading:

-  [Pro Git Book](#) - Comprehensive Git reference
-  [GitHub Docs](#) - Official documentation
-  [VS Code Tips](#)

Practice:

-  [Learn Git Branching](#) - Interactive learning
-  [GitHub Learning Lab](#)



Hochschule Aalen
UNIVERSITY OF APPLIED SCIENCES

Questions?

THANK YOU