



TEASER

Teacher as Avatar

Lehr- und Lernszenario

Python Basics with AI Avatars

Guidance

Inhalt

I. Stammdaten und Kontext.....	3
II. Pädagogisches Design	4
III. Technologische Umsetzung.....	5
IV. Detaillierter Unterrichtsablauf (Lesson Plan)	6
1. Einführung und Orientierung.....	6
2. Durchführung der Aufgabe.....	6
3. Bewertung / Überprüfung.....	7
4. Abschluss der Einheit	7
V. Ressourcen und Begleitmaterialien	7
1. Videos	7
2. Interaktive Komponenten.....	8
3. Medien-Portfolio	8

I. Stammdaten und Kontext

- **Szenario-Titel und Abstrakt:** Das Szenario trägt den Titel „**Python Basics with AI Avatars Guidance**“. Es bietet eine grundlegende Einführung in die Welt der Programmierung für Anfänger im IT- und Softwareentwicklungsbereich. Der Kerninhalt umfasst die Vermittlung fundamentaler **Python-Konzepte wie Variablen, Schleifen, Funktionen und Bedingungen**. Die Besonderheit dieses Szenarios liegt in der methodischen Kombination: Ein **KI-generierter Avatar (Synthesia)** führt durch die Lerninhalte, während die Lernenden den **Programiz Online Python Compiler** zum Testen ihrer Codes nutzen und **ChatGPT** als interaktiven „Buddy“ für die Fehlersuche (Debugging) und die Entwicklung von Lernstrategien einsetzen.
- **Berufsfeld und Zielgruppe:** Dieses Szenario ist im Berufsfeld der **IT, Softwareentwicklung und der allgemeinen digitalen Kompetenzen** angesiedelt. Die Zielgruppe gliedert sich in:
 - **Lernende:** Programmieranfänger, Auszubildende (VET learners) zu Beginn ihrer Python-Ausbildung sowie Personen aus fachfremden Bereichen wie Wirtschaft oder Finanzen, die grundlegende IT-Verständnisse erwerben müssen.
 - **Lehrende:** IT-Instruktoren und Lehrkräfte in der beruflichen Bildung, die innovative digitale Werkzeuge in ihren Unterricht integrieren möchten. Das Szenario ist so konzipiert, dass es auch Lernenden ohne tiefen technischen Hintergrund den Zugang zu abstrakten Coding-Konzepten ermöglicht.
- **Lernziele:** Die zu erwerbenden Kompetenzen sind in drei Kategorien unterteilt:
 - **Wissen (Knowledge):** Verständnis der grundlegenden Python-Syntax und -Logik (Variablen, Datentypen, Kontrollstrukturen wie Schleifen und Bedingungen) sowie der Rolle von KI-Unterstützung im modernen Programmierprozess.
 - **Fähigkeiten (Skills):** Die Fähigkeit, Programmcode eigenständig auszuführen, Fehler im Code zu identifizieren (Debugging), bestehenden Code an neue Anforderungen anzupassen und gezielt mit KI-Tools zur Unterstützung des Programmierens zu interagieren.
 - **Kompetenzen (Competencies):** Entwicklung von Strategien zur **unabhängigen Problemlösung** und zum iterativen Testen. Die Lernenden werden befähigt, Fehler mittels KI-gestützter Ansätze systematisch zu korrigieren und eine kritische Reflexion über die Möglichkeiten und Grenzen von KI beim Programmieren zu entwickeln.

II. Pädagogisches Design

- **Die „Educational Question“:** Die zentrale pädagogische Herausforderung in der Programmierausbildung besteht darin, dass Anfänger oft Schwierigkeiten haben, **abstrakte Coding-Konzepte** zu greifen und bei der Fehlersuche (**Debugging**) schnell frustriert sind. Das Szenario adressiert die spezifische Frage: **„Wie können KI-gestützte Avatare im LMS als Echtzeit-Begleiter fungieren, um Lernende bei der Fehlersuche in Python-Programmierungen zu unterstützen?“**. Der Einsatz von KI und Avataren löst dieses Problem, indem er klare Schritt-für-Schritt-Erklärungen bietet, die kognitive Überlastung reduziert und den Lernenden das Gefühl gibt, einen **„24/7 Coding-Buddy“** an ihrer Seite zu haben.
- **Didaktisches Setting:** Das Szenario ist in den europäischen Kompetenzrahmen **DigComp 2.2** und **DigCompEdu** eingebettet, wobei insbesondere die Bereiche Problemlösung und digitale Inhaltserstellung gefördert werden. Im **SAMR-Modell** erreicht die Einheit die Stufe der **„Modification“ (Umgestaltung)**, da die KI-gestützte, iterative Fehlerbehebung und Code-Optimierung den Lernprozess funktional verändert und vertieft. Die Lehrmethode folgt dem im Projekt etablierten **4-Phasen-Modell**:
 1. **Orientierung (5 Min.):** Ein Avatar führt in die Python-Grundlagen (Variablen, Schleifen, Funktionen) ein.
 2. **Durchführung (8 Min.):** Die Lernenden erstellen eigenständig ein „Hello World“-Programm in einem Online-Compiler.
 3. **Bewertung/Check (10 Min.):** Aktives Troubleshooting fehlerhafter Code-Snippets mit Unterstützung von ChatGPT.
 4. **Abschluss (8-10 Min.):** Gemeinsame Reflexion über die Möglichkeiten und Grenzen von KI beim Programmieren.
- **Rolle des Ausbilders/Lehrers:** Die Lehrkraft fungiert in diesem Szenario primär als **Facilitator und Coach**. Sie tritt als Wissensvermittler in den Hintergrund, da die theoretischen Inhalte durch den Avatar und die KI abgedeckt werden. Zu den Aufgaben gehören:
 - **Einführung und Anleitung:** Den Rahmen der Sitzung setzen und die Lernenden bei der Nutzung des Online-Compilers anleiten.
 - **Lernprozess-Monitoring:** Den Fortschritt beobachten, Notizen zum Interaktionsverhalten mit der KI machen und bei technischen Hürden unterstützen.
 - **Moderation und Klärung:** Die abschließende Diskussionsrunde leiten, um Missverständnisse auszuräumen und sicherzustellen, dass die pädagogischen Ziele erreicht wurden.

III. Technologische Umsetzung

- **KI- und Avatar-Lösung:** In diesem Szenario kommt ein **KI-generierter Avatar** zum Einsatz, der primär **lineare Instruktionseinhalte** und Walkthroughs von Programmierkonzepten liefert. Der Avatar übernimmt im Lernprozess die Rolle eines **strukturierten visuellen Instructors**, der in die Themen einführt, Code-Beispiele demonstriert und die Lernenden schrittweise durch die Übungen leitet. Diese visuelle Führung ergänzt das textbasierte Feedback der KI und fungiert als motivierendes Element, um die kognitive Belastung beim Erlernen abstrakter Syntaxregeln zu reduzieren. Es werden dabei vorwiegend **2D-Avatare** genutzt, die mittels spezialisierter Plattformen erstellt wurden, um eine konsistente Wissensvermittlung ohne menschliches Eingreifen bei jeder Sitzung zu gewährleisten.
- **Technische Werkzeuge:** Die technische Infrastruktur stützt sich auf eine Kombination aus webbasierten Plattformen und Standard-Hardware:
 - **KI-Avatar-Tool: Synthesia** wird zur schnellen Generierung der Video-Tutorials verwendet, wobei Texte in lippensynchrone Avatar-Ansprachen umgewandelt werden.
 - **Interaktive KI (Assistent): ChatGPT** dient als Echtzeit-Begleiter für Fragen und Antworten (Q&A) sowie als entscheidendes Werkzeug für das **Debugging** (Fehlersuche) im Programmcode.
 - **Programmier-Tool:** Der **Programiz Online Python Compiler** wird als integrierte Entwicklungsumgebung im Browser genutzt, damit Lernende Code direkt ausführen und testen können, ohne Software lokal installieren zu müssen.
 - **Lernplattform (LMS):** Die Inhalte werden in ein gängiges **LMS wie Learnpress** integriert, das als zentrale Anlaufstelle für Videos, Quizze und interaktive Pfade fungiert.
 - **Hardware:** Die Durchführung erfolgt an standardmäßigen **Laptops oder PCs** mit stabilem Internetzugang.
- **Software-Hopping-Ansatz:** Die Erstellung der Lerninhalte folgt dem im Projekt TEASER etablierten **niederschweligen „Software-Hopping-Ansatz“**, der verschiedene Werkzeuge ohne Programmieraufwand miteinander verzahnt.
 1. **Inhaltsoptimierung:** Zunächst werden fachliche Skripte erstellt und durch **ChatGPT** sprachlich verfeinert, um eine didaktisch wertvolle Ansprache zu gewährleisten.
 2. **Videoproduktion:** Diese optimierten Texte werden in **Synthesia** importiert, um den Avatar zu generieren, der die theoretischen Grundlagen (wie Variablen und Schleifen) erklärt.
 3. **Interaktive Anwendung:** Die fertigen Videos werden im LMS mit dem **Programiz Compiler** und spezifischen **ChatGPT-Prompts** kombiniert. Dieser Prozess ermöglicht es den Auszubildenden, auch ohne tiefgreifende Multimedia-Kenntnisse professionelle und interaktive digitale Lernerfahrungen zu gestalten.

IV. Detaillierter Unterrichtsablauf (Lesson Plan)

Dieser Unterrichtsplan ist darauf ausgelegt, Programmieranfängern grundlegende Python-Konzepte in einer interaktiven, KI-gestützten Umgebung zu vermitteln.

1. Einführung und Orientierung

- **Dauer:** 5 Minuten.
- **Inhalte:** Die Lernenden erhalten eine Einführung in die Bedeutung der Programmierung mit **Python**, einer Sprache, die von Unternehmen wie Google, Netflix und NASA genutzt wird. Es werden Kernkonzepte wie **Variablen** (als „beschriftete Boxen“), **Schleifen** zur Automatisierung, **Bedingungen** für logische Entscheidungen und **Funktionen** als wiederverwendbare Codestücke vorgestellt. Zudem wird die Rolle von KI als „24/7 Coding-Buddy“ erläutert.
- **Aktivitäten:**
 - **Lernende:** Betrachten der Avatar-basierten Video-Tutorials und erste Interaktionen mit ChatGPT.
 - **Lehrkraft:** Gibt einen Überblick über die Sitzung, führt die Lernenden durch das Setting und hebt die wichtigsten Punkte der Video-Tutorials hervor.
- **Medien:** Synthesia-Avatar-Videos, ChatGPT.

2. Durchführung der Aufgabe

- **Dauer:** 8 Minuten.
- **Inhalte:** Praktische Anwendung grundlegender Python-Funktionen. Ziel ist das Erstellen und Ausführen eines ersten Programms, das eine Nachricht auf dem Bildschirm ausgibt.
- **Aktivitäten:**
 - **Lernende:** Nutzen den **Programiz Online Python Compiler**, um eine Nachricht wie „Hello, Geo“ auszugeben. Sie löschen vorhandenen Beispielcode und schreiben ihre eigenen Befehle basierend auf den Avatar-Instruktionen.
 - **Lehrkraft:** Leitet die Lernenden bei der Interaktion mit dem Online-Compiler an, überwacht den Fortschritt durch Notizen und moderiert den Austausch in der Gruppe.
- **Medien:** KI-Avatare, **Programiz Online Python Compiler**.

3. Bewertung / Überprüfung

- **Dauer:** 10 Minuten.
- **Inhalte: Fehlersuche (Troubleshooting)** und Korrektur gängiger Coding-Fehler (Bugs).
- **Aktivitäten:**
 - **Lernende:** Bearbeiten die Herausforderung „**Fix the Code**“. Sie analysieren fehlerhafte Code-Schnipsel (z. B. Rechtschreibfehler wie prnt statt print oder fehlende Anführungszeichen). Sie nutzen ChatGPT gezielt für Tipps („Warum funktioniert mein Code nicht?“) und präsentieren ihre Lösungen.
 - **Lehrkraft:** Gibt Feedback, um Wissenslücken zu schließen, verstärkt die gelernten Konzepte und führt eine Bewertung der Ergebnisse durch.
- **Medien:** ChatGPT, Programiz Compiler, KI-Avatar.

4. Abschluss der Einheit

- **Dauer:** 8–10 Minuten.
- **Inhalte:** Zusammenfassung des Gelernten und Reflexion über den Einsatz von KI in der Softwareentwicklung.
- **Aktivitäten:**
 - **Lernende:** Fassen die Lernergebnisse zusammen und diskutieren die größten Herausforderungen sowie die Hilfestellung durch die KI.
 - **Lehrkraft:** Moderiert die Abschlussdiskussion, geht auf die Grenzen der KI und die Bedeutung menschlichen Eingreifens ein und räumt letzte Missverständnisse aus.
- **Medien:** Gruppendiskussion, moderiert durch die Lehrkraft (Facilitator).

V. Ressourcen und Begleitmaterialien

1. Videos

Die theoretische Vermittlung basiert auf **Synthesia-Avatar-Videos**:

- **Phase 1: Unlock the Power of Python – Your First Steps into Coding**
 - **Inhalt:** Definition von Programmierung als Schritt-für-Schritt-Anweisung für Computer. Einführung in Python als einsteigerfreundliche Sprache, die von Google, Netflix und der NASA genutzt wird.
 - **Konzepte:** Erläuterung von **Variablen** (Daten, die sich ändern können), **Schleifen** (Wiederholung von Aktionen), **Bedingungen** (Steuerung der Logik) und **Funktionen** (wiederverwendbare Codestücke).
 - **KI-Rolle:** Vorstellung der KI als „**24/7 Coding-Buddy**“, der beim Debuggen und Lernen hilft.

- **Phase 2: Python Functions and Syntax Basics**
 - *Inhalt:* Fokus auf die **Syntax** als „Grammatik des Codes“.
 - *Praxis:* Erklärung der `print()`-Funktion zur Anzeige von Nachrichten, Nutzung von Variablen als „Container“ für Informationen und Einsatz von **Kommentaren (#)** für die menschliche Lesbarkeit.
 - *Anweisung:* Aufforderung, den **Programiz Online Compiler** zu öffnen und ein erstes Programm („Hello, Geo“) zu schreiben.
- **Phase 3: Fix the Code**
 - *Inhalt:* Eine Coding-Challenge zur Fehlerbehebung.
 - *Beispiele:* Analyse von Snippets mit Tippfehlern (z. B. `prnt` statt `print`) oder fehlenden Satzzeichen (Anführungszeichen, Klammern).
 - *Motivation:* Debugging wird als Weg beschrieben, um „**Programmier-Superkräfte**“ aufzubauen.

2. Interaktive Komponenten

Das Szenario integriert verschiedene Werkzeuge für aktives, exploratives Lernen:

- **ChatGPT als Debugging-Assistent:** Lernende sind ausdrücklich dazu aufgefordert, ChatGPT bei Fehlermeldungen direkt zu konsultieren (z. B. „Warum funktioniert mein Code nicht?“). Die KI liefert schrittweise Anleitungen zur Fehlerbehebung.
- **KI-gestütztes Grundlagen-Quiz:** Nach Abschluss der Aufgaben kann ein **4-Fragen-Quiz** zu den Python-Basics gestartet werden.
 - *Feedbackschleife:* Die KI gibt nicht nur einen Score aus, sondern liefert **persönliches Feedback** und individuelle Tipps zur Verbesserung.
- **Programiz Online Python Compiler:** Ein browserbasiertes Tool, das es ermöglicht, Code ohne lokale Installation sofort auszuführen und zu testen.

3. Medien-Portfolio

- **Avatar-Videos:** Eine Suite von linearen Lehrvideos, erstellt mit **Synthesia**.
- **YouTube-Portfolio:** Direkte Links zu den Videos für jede Phase:
 - Phase 1: *Unlock the Power of Python*.
 - Phase 2: *Python Functions and Syntax Basics*.
 - Phase 3: *Fix the Code*.
- **Visuelle Materialien:** Screenshots der korrekten Syntax und grafische Darstellungen von Variablen als „beschriftete Boxen“ zur Veranschaulichung abstrakter Konzepte.