

Informatik 10

Arbeitsheft mit 21 Lernseiten, Projekten und Lösungshinweisen

So arbeitest du mit dem Heft

1. Lies Kurz erklärt und markiere Fachbegriffe.
2. Löse zuerst ohne Lösungsteil.
3. Prüfe mit einem Testfall.
4. Verbessere deine Antwort sichtbar.

Dieses Heft ist vollständig eigenständig für SchlauStart erstellt. Es enthält keine kopierten Buchseiten oder fremden Arbeitsblätter. Für Aufgaben werden keine Namen, Kontodaten oder anderen personenbezogenen Angaben benötigt.

Inhalt

01	Python: Erste Schritte
02	Python strukturieren
03	Funktionen in Python
04	Listen verarbeiten
05	Dateien lesen und schreiben
06	Fehler und Ausnahmen behandeln
07	Python-Projekte
08	Ein textbasiertes Spiel entwickeln
09	Daten mit Python auswerten
10	Ein Python-Projekt dokumentieren
11	HTML & Webseiten
12	Webprojekt
13	Eine Website planen
14	HTML und CSS prüfen
15	Urheberrecht bei Webinhalten
16	Eine Website veröffentlichen
17	Künstliche Intelligenz
18	Trainingsdaten und Verzerrungen
19	Generative KI und Sprachmodelle
20	KI-Ergebnisse kritisch prüfen
21	Regeln für verantwortliche KI-Nutzung

01 · Python: Erste Schritte

Name: _____

Datum: _____

Kurz erklärt

Python: Erste Schritte Python ist eine textbasierte Programmiersprache mit gut lesbarer Struktur. Einrückung, Datentypen und genaue Fehlermeldungen gehören zur Sprache.

Beispiel: Ein kleines Programm liest einen Namen, berechnet aus Zahlen ein Ergebnis und gibt einen formatierten Satz aus.

Drei Fachbegriffe

Datentyp	Er bestimmt Wertebereich und erlaubte Operationen.
Einrückung	Sie markiert zusammengehörige Programmblöcke.
Ausnahme	Sie beschreibt einen Fehler während der Ausführung.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Schreibe Pseudocode für eine geprüfte Zahleneingabe und übertrage ihn anschließend in Python.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Entwickelt ein textbasiertes Spiel oder eine Datenauswertung mit mindestens drei Funktionen, Testfällen und kurzer README.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

02 · Python strukturieren

Name: _____

Datum: _____

Kurz erklärt

Python strukturieren Funktionen, Module und Datenstrukturen teilen Python-Programme in prüfbare Einheiten. Schnittstellen werden durch Parameter und Rückgaben sichtbar.

Beispiel: Eine Auswertung trennt Datei laden, Werte bereinigen, Kennzahl berechnen und Bericht ausgeben.

Drei Fachbegriffe

Datentyp	Er bestimmt Wertebereich und erlaubte Operationen.
Einrückung	Sie markiert zusammengehörige Programmblöcke.
Ausnahme	Sie beschreibt einen Fehler während der Ausführung.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Schreibe Pseudocode für eine geprüfte Zahleneingabe und übertrage ihn anschließend in Python.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Entwickelt ein textbasiertes Spiel oder eine Datenauswertung mit mindestens drei Funktionen, Testfällen und kurzer README.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

03 · Funktionen in Python

Name: _____

Datum: _____

Kurz erklärt

Funktionen in Python `def` legt eine Funktion fest. Parameter empfangen Werte, `return` liefert ein Ergebnis und ein Docstring kann Zweck und Erwartungen erklären.

Beispiel: `def quadrat(x): return x * x` berechnet für jede Zahl denselben Zusammenhang und lässt sich getrennt testen.

Drei Fachbegriffe

Datentyp	Er bestimmt Wertebereich und erlaubte Operationen.
Einrückung	Sie markiert zusammengehörige Programmblöcke.
Ausnahme	Sie beschreibt einen Fehler während der Ausführung.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Schreibe Pseudocode für eine geprüfte Zahleneingabe und übertrage ihn anschließend in Python.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Entwickelt ein textbasiertes Spiel oder eine Datenauswertung mit mindestens drei Funktionen, Testfällen und kurzer README.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

04 · Listen verarbeiten

Name: _____

Datum: _____

Kurz erklärt

Listen verarbeiten Listen speichern geordnete, veränderliche Folgen. Index, Schleife und Methoden erlauben Zugriff, Ergänzung, Entfernen und Auswertung.

Beispiel: `noten.append(2)` ergänzt eine Note; `sum(noten) / len(noten)` berechnet den Mittelwert einer nicht leeren Liste.

Drei Fachbegriffe

Datentyp	Er bestimmt Wertebereich und erlaubte Operationen.
Einrückung	Sie markiert zusammengehörige Programmblöcke.
Ausnahme	Sie beschreibt einen Fehler während der Ausführung.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Schreibe Pseudocode für eine geprüfte Zahleneingabe und übertrage ihn anschließend in Python.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Entwickelt ein textbasiertes Spiel oder eine Datenauswertung mit mindestens drei Funktionen, Testfällen und kurzer README.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

05 · Dateien lesen und schreiben

Name: _____

Datum: _____

Kurz erklärt

Dateien lesen und schreiben Dateien bewahren Daten über einen Programmlauf hinaus. Zeichencodierung, Dateipfad und ein kontrolliert geschlossener Zugriff sind Teil der Verarbeitung.

Beispiel: with open('werte.txt', encoding='utf-8') as datei liest Text und schließt die Datei auch bei einem Fehler zuverlässig.

Drei Fachbegriffe

Datentyp	Er bestimmt Wertebereich und erlaubte Operationen.
Einrückung	Sie markiert zusammengehörige Programmblöcke.
Ausnahme	Sie beschreibt einen Fehler während der Ausführung.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Schreibe Pseudocode für eine geprüfte Zahleneingabe und übertrage ihn anschließend in Python.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Entwickelt ein textbasiertes Spiel oder eine Datenauswertung mit mindestens drei Funktionen, Testfällen und kurzer README.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

06 · Fehler und Ausnahmen behandeln

Name: _____

Datum: _____

Kurz erklärt

Fehler und Ausnahmen behandeln Syntaxfehler verhindern den Start, Laufzeitfehler entstehen während der Ausführung und logische Fehler liefern falsche Ergebnisse. Ausnahmen werden nur gezielt behandelt.

Beispiel: try fängt ValueError bei `int('abc')` ab und erklärt die ungültige Zahl, statt alle Fehler unsichtbar zu verschlucken.

Drei Fachbegriffe

Datentyp	Er bestimmt Wertebereich und erlaubte Operationen.
Einrückung	Sie markiert zusammengehörige Programmblöcke.
Ausnahme	Sie beschreibt einen Fehler während der Ausführung.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Schreibe Pseudocode für eine geprüfte Zahleneingabe und übertrage ihn anschließend in Python.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Entwickelt ein textbasiertes Spiel oder eine Datenauswertung mit mindestens drei Funktionen, Testfällen und kurzer README.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

07 · Python-Projekte

Name: _____

Datum: _____

Kurz erklärt

Python-Projekte Ein Python-Projekt verbindet Anforderungen, Versionsstände, Tests, Dokumentation und eine nachvollziehbare Dateistruktur.

Beispiel: Ein Auswertungsprogramm besitzt Beispieldaten, Funktionen, Testfälle und eine README mit Startanleitung und bekannten Grenzen.

Drei Fachbegriffe

Datentyp	Er bestimmt Wertebereich und erlaubte Operationen.
Einrückung	Sie markiert zusammengehörige Programmblöcke.
Ausnahme	Sie beschreibt einen Fehler während der Ausführung.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Schreibe Pseudocode für eine geprüfte Zahleneingabe und übertrage ihn anschließend in Python.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Entwickelt ein textbasiertes Spiel oder eine Datenauswertung mit mindestens drei Funktionen, Testfällen und kurzer README.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

08 · Ein textbasiertes Spiel entwickeln

Name: _____

Datum: _____

Kurz erklärt

Ein textbasiertes Spiel entwickeln Ein Textspiel modelliert Orte, Zustände, Entscheidungen und Siegbedingungen. Funktionen verhindern wiederholten Code, Datenstrukturen speichern die Spielwelt.

Beispiel: Ein Wörterbuch verknüpft Räume mit Ausgängen; die Hauptschleife liest Befehle, verändert Position und prüft das Ziel.

Drei Fachbegriffe

Datentyp	Er bestimmt Wertebereich und erlaubte Operationen.
Einrückung	Sie markiert zusammengehörige Programmblöcke.
Ausnahme	Sie beschreibt einen Fehler während der Ausführung.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Schreibe Pseudocode für eine geprüfte Zahleneingabe und übertrage ihn anschließend in Python.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Entwickelt ein textbasiertes Spiel oder eine Datenauswertung mit mindestens drei Funktionen, Testfällen und kurzer README.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

09 · Daten mit Python auswerten

Name: _____

Datum: _____

Kurz erklärt

Daten mit Python auswerten Datenanalyse beginnt mit Herkunft, Datentypen und Bereinigung. Erst danach werden Kennzahlen berechnet und Ergebnisse im Kontext interpretiert.

Beispiel: Messwerte mit leerer Zeile und Tippfehler werden geprüft, bevor Minimum, Maximum und Mittelwert ausgegeben werden.

Drei Fachbegriffe

Datentyp	Er bestimmt Wertebereich und erlaubte Operationen.
Einrückung	Sie markiert zusammengehörige Programmblöcke.
Ausnahme	Sie beschreibt einen Fehler während der Ausführung.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Schreibe Pseudocode für eine geprüfte Zahleneingabe und übertrage ihn anschließend in Python.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Entwickelt ein textbasiertes Spiel oder eine Datenauswertung mit mindestens drei Funktionen, Testfällen und kurzer README.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

10 · Ein Python-Projekt dokumentieren

Name: _____

Datum: _____

Kurz erklärt

Ein Python-Projekt dokumentieren Dokumentation erklärt Ziel, Voraussetzungen, Bedienung, Datenformat, wichtige Entscheidungen, Tests und Grenzen. Kommentare erläutern Gründe statt offensichtliche Zeilen.

Beispiel: Eine README nennt Python-Version, Startbefehl, Beispielinput und erwartete Ausgabe; Funktionen tragen kurze Docstrings.

Drei Fachbegriffe

Datentyp	Er bestimmt Wertebereich und erlaubte Operationen.
Einrückung	Sie markiert zusammengehörige Programmblöcke.
Ausnahme	Sie beschreibt einen Fehler während der Ausführung.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Schreibe Pseudocode für eine geprüfte Zahleneingabe und übertrage ihn anschließend in Python.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Entwickelt ein textbasiertes Spiel oder eine Datenauswertung mit mindestens drei Funktionen, Testfällen und kurzer README.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

11 · HTML & Webseiten

Name: _____

Datum: _____

Kurz erklärt

HTML und Webseiten HTML beschreibt Bedeutung und Struktur von Webinhalten. CSS gestaltet die Darstellung; Browser erzeugen aus Dateien ein zugängliches Dokument.

Beispiel: Eine Überschrift wird als h1 ausgezeichnet und nicht nur optisch groß gemacht, damit Suchmaschine und Screenreader ihre Bedeutung erkennen.

Drei Fachbegriffe

Semantik	Elemente beschreiben Bedeutung statt nur Aussehen.
Kaskade	Mehrere CSS-Regeln werden nach festen Vorrangregeln kombiniert.
Barrierefreiheit	Inhalte bleiben mit verschiedenen Fähigkeiten und Geräten nutzbar.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Entwirf die semantische Struktur einer Projektseite mit Navigation, Hauptinhalt, zwei Abschnitten und Fußbereich.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Plant, baut und prüft eine kleine Website mit mindestens drei Seiten, einheitlicher Navigation, eigenem Inhalt und vollständigem Quellenverzeichnis.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

12 · Webprojekt

Name: _____

Datum: _____

Kurz erklärt

Webprojekt Ein Webprojekt verbindet Zielgruppe, Inhaltsstruktur, semantisches HTML, einheitliches CSS, Tests, Quellen und Veröffentlichung.

Beispiel: Eine Schul-AG-Seite wird zuerst als Seitenbaum geplant und anschließend auf Handy, Tablet, Tastaturbedienung und fehlerhafte Links geprüft.

Drei Fachbegriffe

Semantik	Elemente beschreiben Bedeutung statt nur Aussehen.
Kaskade	Mehrere CSS-Regeln werden nach festen Vorrangregeln kombiniert.
Barrierefreiheit	Inhalte bleiben mit verschiedenen Fähigkeiten und Geräten nutzbar.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Entwirf die semantische Struktur einer Projektseite mit Navigation, Hauptinhalt, zwei Abschnitten und Fußbereich.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Plant, baut und prüft eine kleine Website mit mindestens drei Seiten, einheitlicher Navigation, eigenem Inhalt und vollständigem Quellenverzeichnis.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

13 · Eine Website planen

Name: _____

Datum: _____

Kurz erklärt

Eine Website planen Planung klärt Zielgruppe, Aufgabe, Inhalte, Navigation und Erfolgskriterien. Wireframes ordnen Informationen, bevor Farben und Details ablenken.

Beispiel: Für eine Projektseite werden Start, Termine, Ergebnisse und Kontakt als klare Bereiche mit kurzen Nutzerwegen angeordnet.

Drei Fachbegriffe

Semantik	Elemente beschreiben Bedeutung statt nur Aussehen.
Kaskade	Mehrere CSS-Regeln werden nach festen Vorrangregeln kombiniert.
Barrierefreiheit	Inhalte bleiben mit verschiedenen Fähigkeiten und Geräten nutzbar.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Entwirf die semantische Struktur einer Projektseite mit Navigation, Hauptinhalt, zwei Abschnitten und Fußbereich.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Plant, baut und prüft eine kleine Website mit mindestens drei Seiten, einheitlicher Navigation, eigenem Inhalt und vollständigem Quellenverzeichnis.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

14 · HTML und CSS prüfen

Name: _____

Datum: _____

Kurz erklärt

HTML und CSS prüfen Validierung findet formale Fehler; manuelle Tests prüfen Bedienung, Inhalt, verschiedene Größen und assistive Nutzung. Beides ergänzt sich.

Beispiel: Ein Validator meldet ein nicht geschlossenes Element, während ein Tastaturtest einen unsichtbaren Fokus entdeckt.

Drei Fachbegriffe

Semantik	Elemente beschreiben Bedeutung statt nur Aussehen.
Kaskade	Mehrere CSS-Regeln werden nach festen Vorrangregeln kombiniert.
Barrierefreiheit	Inhalte bleiben mit verschiedenen Fähigkeiten und Geräten nutzbar.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Entwirf die semantische Struktur einer Projektseite mit Navigation, Hauptinhalt, zwei Abschnitten und Fußbereich.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Plant, baut und prüft eine kleine Website mit mindestens drei Seiten, einheitlicher Navigation, eigenem Inhalt und vollständigem Quellenverzeichnis.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

15 · Urheberrecht bei Webinhalten

Name: _____

Datum: _____

Kurz erklärt

Urheberrecht bei Webinhalten Eigene Webseiten dürfen fremde Texte, Bilder oder Musik nur mit Erlaubnis oder passender Lizenz nutzen. Quelle, Urheber, Lizenz und Änderungen werden dokumentiert.

Beispiel: Ein CC-BY-Bild wird mit Name, Werktitel, Quelle und Lizenz vermerkt; ein Bild ohne Lizenzangabe wird nicht übernommen.

Drei Fachbegriffe

Semantik	Elemente beschreiben Bedeutung statt nur Aussehen.
Kaskade	Mehrere CSS-Regeln werden nach festen Vorrangregeln kombiniert.
Barrierefreiheit	Inhalte bleiben mit verschiedenen Fähigkeiten und Geräten nutzbar.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Entwirf die semantische Struktur einer Projektseite mit Navigation, Hauptinhalt, zwei Abschnitten und Fußbereich.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Plant, baut und prüft eine kleine Website mit mindestens drei Seiten, einheitlicher Navigation, eigenem Inhalt und vollständigem Quellenverzeichnis.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

16 · Eine Website veröffentlichen

Name: _____

Datum: _____

Kurz erklärt

Eine Website veröffentlichen Vor Veröffentlichung werden personenbezogene Daten, Rechte, Dateipfade, Sicherheit, Impressumsbedarf und Aktualisierung geklärt. Öffentlich bedeutet weltweit abrufbar.

Beispiel: Ein Projekt nutzt nur freigegebene Bilder, relative Links, HTTPS und eine verantwortliche Person für spätere Korrekturen.

Drei Fachbegriffe

Semantik	Elemente beschreiben Bedeutung statt nur Aussehen.
Kaskade	Mehrere CSS-Regeln werden nach festen Vorrangregeln kombiniert.
Barrierefreiheit	Inhalte bleiben mit verschiedenen Fähigkeiten und Geräten nutzbar.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Entwirf die semantische Struktur einer Projektseite mit Navigation, Hauptinhalt, zwei Abschnitten und Fußbereich.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Plant, baut und prüft eine kleine Website mit mindestens drei Seiten, einheitlicher Navigation, eigenem Inhalt und vollständigem Quellenverzeichnis.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

17 · Künstliche Intelligenz

Name: _____

Datum: _____

Kurz erklärt

Künstliche Intelligenz KI-Systeme bearbeiten Aufgaben, für die Menschen Wahrnehmen, Entscheiden oder Sprache verwenden. Ihre Ergebnisse entstehen aus Daten, Modellen und programmierten Zielen.

Beispiel: Eine Bilderkennung ordnet ein neues Foto anhand gelernter Merkmale einer Kategorie zu, ohne das Motiv wie ein Mensch zu verstehen.

Drei Fachbegriffe

Trainingsdaten	Beispiele prägen, welche Muster ein Modell lernt.
Modell	Gewichtete Regeln erzeugen eine Ausgabe aus Eingaben.
Evaluation	Unbekannte Testdaten zeigen Leistung und Fehler.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Entwirf einen Entscheidungsbaum, der Gegenstände nach Form, Farbe und Größe in drei Kisten sortiert.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Trainiert Menschen als Klassifikationsmodell: Eine Gruppe legt Beispiele und Merkmale fest, eine zweite formuliert Regeln, eine dritte testet unbekannte Fälle und dokumentiert Fehlentscheidungen.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

18 · Trainingsdaten und Verzerrungen

Name: _____

Datum: _____

Kurz erklärt

Trainingsdaten und Verzerrungen Ein Modell übernimmt Muster und Lücken seiner Trainingsdaten. Unausgewogene Auswahl, falsche Beschriftung oder historische Benachteiligung können Ergebnisse verzerren.

Beispiel: Ein Bildermodell, das Fahrräder nur bei Sonnenschein kennt, kann Fahrräder im Schnee schlechter erkennen.

Drei Fachbegriffe

Trainingsdaten	Beispiele prägen, welche Muster ein Modell lernt.
Modell	Gewichtete Regeln erzeugen eine Ausgabe aus Eingaben.
Evaluation	Unbekannte Testdaten zeigen Leistung und Fehler.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Entwirf einen Entscheidungsbaum, der Gegenstände nach Form, Farbe und Größe in drei Kisten sortiert.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Trainiert Menschen als Klassifikationsmodell: Eine Gruppe legt Beispiele und Merkmale fest, eine zweite formuliert Regeln, eine dritte testet unbekannte Fälle und dokumentiert Fehlentscheidungen.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

19 · Generative KI und Sprachmodelle

Name: _____

Datum: _____

Kurz erklärt

Generative KI und Sprachmodelle Sprachmodelle berechnen wahrscheinliche Fortsetzungen aus großen Textmengen. Sie erzeugen flüssige Texte, besitzen aber keine eingebaute Wahrheitsprüfung.

Beispiel: Ein Modell kann eine glaubwürdig klingende, aber erfundene Quellenangabe erzeugen, weil Form und Inhalt statistisch passen.

Drei Fachbegriffe

Trainingsdaten	Beispiele prägen, welche Muster ein Modell lernt.
Modell	Gewichtete Regeln erzeugen eine Ausgabe aus Eingaben.
Evaluation	Unbekannte Testdaten zeigen Leistung und Fehler.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Entwirf einen Entscheidungsbaum, der Gegenstände nach Form, Farbe und Größe in drei Kisten sortiert.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Trainiert Menschen als Klassifikationsmodell: Eine Gruppe legt Beispiele und Merkmale fest, eine zweite formuliert Regeln, eine dritte testet unbekannte Fälle und dokumentiert Fehlentscheidungen.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

20 · KI-Ergebnisse kritisch prüfen

Name: _____

Datum: _____

Kurz erklärt

KI-Ergebnisse kritisch prüfen KI-Ausgaben werden mit Auftrag, Quellen, Fachwissen, Aktualität und möglichen Betroffenen geprüft. Unsicherheit wird sichtbar gemacht.

Beispiel: Eine historische Aussage wird in einer Primärquelle und einer unabhängigen Fachquelle kontrolliert, bevor sie übernommen wird.

Drei Fachbegriffe

Trainingsdaten	Beispiele prägen, welche Muster ein Modell lernt.
Modell	Gewichtete Regeln erzeugen eine Ausgabe aus Eingaben.
Evaluation	Unbekannte Testdaten zeigen Leistung und Fehler.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Entwirf einen Entscheidungsbaum, der Gegenstände nach Form, Farbe und Größe in drei Kisten sortiert.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Trainiert Menschen als Klassifikationsmodell: Eine Gruppe legt Beispiele und Merkmale fest, eine zweite formuliert Regeln, eine dritte testet unbekannte Fälle und dokumentiert Fehlentscheidungen.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

21 · Regeln für verantwortliche KI-Nutzung

Name: _____

Datum: _____

Kurz erklärt

Regeln für verantwortliche KI-Nutzung Verantwortliche Nutzung schützt persönliche Daten, kennzeichnet Unterstützung, prüft Ergebnisse und lässt wichtige Entscheidungen bei Menschen.

Beispiel: In einen KI-Dienst werden keine Namen oder vertraulichen Schülerdaten eingegeben; verwendete Hilfe wird im Projekt transparent angegeben.

Drei Fachbegriffe

Trainingsdaten	Beispiele prägen, welche Muster ein Modell lernt.
Modell	Gewichtete Regeln erzeugen eine Ausgabe aus Eingaben.
Evaluation	Unbekannte Testdaten zeigen Leistung und Fehler.

1. Erkläre das Thema in eigenen Worten. Nutze mindestens zwei Fachbegriffe aus der Tabelle.

2. Zerlege das Beispiel in Ausgangslage, Verarbeitung und überprüfbares Ergebnis.

3. Entwirf einen Entscheidungsbaum, der Gegenstände nach Form, Farbe und Größe in drei Kisten sortiert.

4. Notiere einen Rand- oder Fehlerfall. Sage zuerst voraus, was passieren soll, und begründe danach deine Erwartung.

Mini-Projekt

Trainiert Menschen als Klassifikationsmodell: Eine Gruppe legt Beispiele und Merkmale fest, eine zweite formuliert Regeln, eine dritte testet unbekannte Fälle und dokumentiert Fehlentscheidungen.

Dokumentiere Ziel, Plan, drei Tests, einen gefundenen Fehler und eine Verbesserung.

Lösungen und Bewertungshinweise

Die Hinweise zeigen mögliche fachliche Kerne. Offene Aufgaben dürfen anders gelöst sein, wenn Begriffe, Begründung und Test nachvollziehbar stimmen.

Lösungshinweise

01 · Python: Erste Schritte

1: Python ist eine textbasierte Programmiersprache mit gut lesbarer Struktur. Einrückung, Datentypen und genaue Fehlermeldungen gehören zur Sprache.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Ein kleines Programm liest einen Namen, berechnet aus Zahlen ein Ergebnis und gibt einen formatierten Satz aus. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Die Eingabe wird als Text gelesen, gezielt in eine Zahl umgewandelt, bei ungültigem Format erklärt und erst danach verarbeitet.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

02 · Python strukturieren

1: Funktionen, Module und Datenstrukturen teilen Python-Programme in prüfbare Einheiten. Schnittstellen werden durch Parameter und Rückgaben sichtbar.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Eine Auswertung trennt Datei laden, Werte bereinigen, Kennzahl berechnen und Bericht ausgeben. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Die Eingabe wird als Text gelesen, gezielt in eine Zahl umgewandelt, bei ungültigem Format erklärt und erst danach verarbeitet.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

Lösungshinweise

03 · Funktionen in Python

1: def legt eine Funktion fest. Parameter empfangen Werte, return liefert ein Ergebnis und ein Docstring kann Zweck und Erwartungen erklären.

2: Ausgangslage und Daten stammen aus diesem Beispiel: `def quadrat(x): return x * x` berechnet für jede Zahl denselben Zusammenhang und lässt sich getrennt testen. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Die Eingabe wird als Text gelesen, gezielt in eine Zahl umgewandelt, bei ungültigem Format erklärt und erst danach verarbeitet.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

04 · Listen verarbeiten

1: Listen speichern geordnete, veränderliche Folgen. Index, Schleife und Methoden erlauben Zugriff, Ergänzung, Entfernen und Auswertung.

2: Ausgangslage und Daten stammen aus diesem Beispiel: `noten.append(2)` ergänzt eine Note; `sum(noten) / len(noten)` berechnet den Mittelwert einer nicht leeren Liste. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Die Eingabe wird als Text gelesen, gezielt in eine Zahl umgewandelt, bei ungültigem Format erklärt und erst danach verarbeitet.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

Lösungshinweise

05 · Dateien lesen und schreiben

1: Dateien bewahren Daten über einen Programmlauf hinaus. Zeichencodierung, Dateipfad und ein kontrolliert geschlossener Zugriff sind Teil der Verarbeitung.

2: Ausgangslage und Daten stammen aus diesem Beispiel: `with open('werte.txt', encoding='utf-8')` als Datei liest Text und schließt die Datei auch bei einem Fehler zuverlässig. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Die Eingabe wird als Text gelesen, gezielt in eine Zahl umgewandelt, bei ungültigem Format erklärt und erst danach verarbeitet.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

06 · Fehler und Ausnahmen behandeln

1: Syntaxfehler verhindern den Start, Laufzeitfehler entstehen während der Ausführung und logische Fehler liefern falsche Ergebnisse. Ausnahmen werden nur gezielt behandelt.

2: Ausgangslage und Daten stammen aus diesem Beispiel: `try` fängt `ValueError` bei `int('abc')` ab und erklärt die ungültige Zahl, statt alle Fehler unsichtbar zu verschlucken. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Die Eingabe wird als Text gelesen, gezielt in eine Zahl umgewandelt, bei ungültigem Format erklärt und erst danach verarbeitet.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

Lösungshinweise

07 · Python-Projekte

1: Ein Python-Projekt verbindet Anforderungen, Versionsstände, Tests, Dokumentation und eine nachvollziehbare Dateistruktur.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Ein Auswertungsprogramm besitzt Beispieldaten, Funktionen, Testfälle und eine README mit Startanleitung und bekannten Grenzen. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Die Eingabe wird als Text gelesen, gezielt in eine Zahl umgewandelt, bei ungültigem Format erklärt und erst danach verarbeitet.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

08 · Ein textbasiertes Spiel entwickeln

1: Ein Textspiel modelliert Orte, Zustände, Entscheidungen und Siegbedingungen. Funktionen verhindern wiederholten Code, Datenstrukturen speichern die Spielwelt.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Ein Wörterbuch verknüpft Räume mit Ausgängen; die Hauptschleife liest Befehle, verändert Position und prüft das Ziel. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Die Eingabe wird als Text gelesen, gezielt in eine Zahl umgewandelt, bei ungültigem Format erklärt und erst danach verarbeitet.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

Lösungshinweise

09 · Daten mit Python auswerten

1: Datenanalyse beginnt mit Herkunft, Datentypen und Bereinigung. Erst danach werden Kennzahlen berechnet und Ergebnisse im Kontext interpretiert.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Messwerte mit leerer Zeile und Tippfehler werden geprüft, bevor Minimum, Maximum und Mittelwert ausgegeben werden. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Die Eingabe wird als Text gelesen, gezielt in eine Zahl umgewandelt, bei ungültigem Format erklärt und erst danach verarbeitet.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

10 · Ein Python-Projekt dokumentieren

1: Dokumentation erklärt Ziel, Voraussetzungen, Bedienung, Datenformat, wichtige Entscheidungen, Tests und Grenzen. Kommentare erläutern Gründe statt offensichtliche Zeilen.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Eine README nennt Python-Version, Startbefehl, Beispielinput und erwartete Ausgabe; Funktionen tragen kurze Docstrings. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Die Eingabe wird als Text gelesen, gezielt in eine Zahl umgewandelt, bei ungültigem Format erklärt und erst danach verarbeitet.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

Lösungshinweise

11 · HTML & Webseiten

1: HTML beschreibt Bedeutung und Struktur von Webinhalten. CSS gestaltet die Darstellung; Browser erzeugen aus Dateien ein zugängliches Dokument.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Eine Überschrift wird als h1 ausgezeichnet und nicht nur optisch groß gemacht, damit Suchmaschine und Screenreader ihre Bedeutung erkennen. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Eine mögliche Struktur nutzt header und nav, genau ein main, darin section oder article und am Ende footer; Überschriften bilden eine nachvollziehbare Hierarchie.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

12 · Webprojekt

1: Ein Webprojekt verbindet Zielgruppe, Inhaltsstruktur, semantisches HTML, einheitliches CSS, Tests, Quellen und Veröffentlichung.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Eine Schul-AG-Seite wird zuerst als Seitenbaum geplant und anschließend auf Handy, Tablet, Tastaturbedienung und fehlerhafte Links geprüft. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Eine mögliche Struktur nutzt header und nav, genau ein main, darin section oder article und am Ende footer; Überschriften bilden eine nachvollziehbare Hierarchie.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

Lösungshinweise

13 · Eine Website planen

1: Planung klärt Zielgruppe, Aufgabe, Inhalte, Navigation und Erfolgskriterien. Wireframes ordnen Informationen, bevor Farben und Details ablenken.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Für eine Projektseite werden Start, Termine, Ergebnisse und Kontakt als klare Bereiche mit kurzen Nutzerwegen angeordnet. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Eine mögliche Struktur nutzt header und nav, genau ein main, darin section oder article und am Ende footer; Überschriften bilden eine nachvollziehbare Hierarchie.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

14 · HTML und CSS prüfen

1: Validierung findet formale Fehler; manuelle Tests prüfen Bedienung, Inhalt, verschiedene Größen und assistive Nutzung. Beides ergänzt sich.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Ein Validator meldet ein nicht geschlossenes Element, während ein Tastaturtest einen unsichtbaren Fokus entdeckt. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Eine mögliche Struktur nutzt header und nav, genau ein main, darin section oder article und am Ende footer; Überschriften bilden eine nachvollziehbare Hierarchie.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

Lösungshinweise

15 · Urheberrecht bei Webinhalten

1: Eigene Webseiten dürfen fremde Texte, Bilder oder Musik nur mit Erlaubnis oder passender Lizenz nutzen. Quelle, Urheber, Lizenz und Änderungen werden dokumentiert.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Ein CC-BY-Bild wird mit Name, Werktitel, Quelle und Lizenz vermerkt; ein Bild ohne Lizenzangabe wird nicht übernommen. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Eine mögliche Struktur nutzt header und nav, genau ein main, darin section oder article und am Ende footer; Überschriften bilden eine nachvollziehbare Hierarchie.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

16 · Eine Website veröffentlichen

1: Vor Veröffentlichung werden personenbezogene Daten, Rechte, Dateipfade, Sicherheit, Impressumsbedarf und Aktualisierung geklärt. Öffentlich bedeutet weltweit abrufbar.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Ein Projekt nutzt nur freigegebene Bilder, relative Links, HTTPS und eine verantwortliche Person für spätere Korrekturen. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Eine mögliche Struktur nutzt header und nav, genau ein main, darin section oder article und am Ende footer; Überschriften bilden eine nachvollziehbare Hierarchie.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

Lösungshinweise

17 · Künstliche Intelligenz

1: KI-Systeme bearbeiten Aufgaben, für die Menschen Wahrnehmen, Entscheiden oder Sprache verwenden. Ihre Ergebnisse entstehen aus Daten, Modellen und programmierten Zielen.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Eine Bilderkennung ordnet ein neues Foto anhand gelernter Merkmale einer Kategorie zu, ohne das Motiv wie ein Mensch zu verstehen. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Jede innere Stelle stellt eine eindeutige Frage; jede Antwort führt zu einem weiteren Knoten oder genau einer Zielkiste.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

18 · Trainingsdaten und Verzerrungen

1: Ein Modell übernimmt Muster und Lücken seiner Trainingsdaten. Unausgewogene Auswahl, falsche Beschriftung oder historische Benachteiligung können Ergebnisse verzerren.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Ein Bildermodell, das Fahrräder nur bei Sonnenschein kennt, kann Fahrräder im Schnee schlechter erkennen. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Jede innere Stelle stellt eine eindeutige Frage; jede Antwort führt zu einem weiteren Knoten oder genau einer Zielkiste.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

Lösungshinweise

19 · Generative KI und Sprachmodelle

1: Sprachmodelle berechnen wahrscheinliche Fortsetzungen aus großen Textmengen. Sie erzeugen flüssige Texte, besitzen aber keine eingebaute Wahrheitsprüfung.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Ein Modell kann eine glaubwürdig klingende, aber erfundene Quellenangabe erzeugen, weil Form und Inhalt statistisch passen. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Jede innere Stelle stellt eine eindeutige Frage; jede Antwort führt zu einem weiteren Knoten oder genau einer Zielkiste.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

20 · KI-Ergebnisse kritisch prüfen

1: KI-Ausgaben werden mit Auftrag, Quellen, Fachwissen, Aktualität und möglichen Betroffenen geprüft. Unsicherheit wird sichtbar gemacht.

2: Ausgangslage und Daten stammen aus diesem Beispiel: Eine historische Aussage wird in einer Primärquelle und einer unabhängigen Fachquelle kontrolliert, bevor sie übernommen wird. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Jede innere Stelle stellt eine eindeutige Frage; jede Antwort führt zu einem weiteren Knoten oder genau einer Zielkiste.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

Lösungshinweise

21 · Regeln für verantwortliche KI-Nutzung

1: Verantwortliche Nutzung schützt persönliche Daten, kennzeichnet Unterstützung, prüft Ergebnisse und lässt wichtige Entscheidungen bei Menschen.

2: Ausgangslage und Daten stammen aus diesem Beispiel: In einen KI-Dienst werden keine Namen oder vertraulichen Schülerdaten eingegeben; verwendete Hilfe wird im Projekt transparent angegeben. Die Verarbeitung wird mit dem Kerngedanken der Seite begründet; das Ergebnis muss beobachtbar sein.

3: Jede innere Stelle stellt eine eindeutige Frage; jede Antwort führt zu einem weiteren Knoten oder genau einer Zielkiste.

4: Individuelle Lösung. Bewertbar sind eine einzige gezielt veränderte Voraussetzung, eine begründete Vorhersage und ein reproduzierbarer Test.

Quellen und Rechtliches

Alle Erklärtexpte, Aufgaben, Projekte und Lösungshinweise dieses Hefts: SchlauStart Eigenmaterial, 2026.

Weiterführende offene Materialien werden auf den zugehörigen SchlauStart-Themenseiten mit Urheber, Quelle und Lizenz einzeln ausgewiesen. Verlinkte Angebote sind nicht Bestandteil dieses Hefts.

Scratch ist eine Marke der Scratch Foundation. Calliope mini, LEGO, Ozobot, Python, Open Roberta und MDN werden ausschließlich zur sachlichen Benennung der jeweiligen Lernumgebung verwendet.