



Leitfaden für Jugendarbeiter

CC BY-NC-SA 4.0



Diese Schulung von Code For Future steht unter der Creative Commons Lizenz Namensnennung – Nicht-kommerziell – Weitergabe unter gleichen Bedingungen 4.0 International



Kapitel 1.1 – Leitfaden zur Programmier-Terminologie: Wichtige Begriffe für Einsteiger:innen

Abstract

Diese Anleitung hilft Lehrkräften dabei, Schüler:innen grundlegende Programmierbegriffe näherzubringen. Sie enthält Definitionen und Kontexte für zentrale Begriffe, um das Verständnis und die effektive Anwendung in Programmieraufgaben zu erleichtern. Der Leitfaden sorgt dafür, dass das grundlegende Vokabular der Programmierung klar und zugänglich ist – eine wichtige Voraussetzung für weiteres Lernen im Bereich der Informatik.

Schlüsselwörter

Programmieren, Terminologie, Programmierung, Definitionen, Vokabular, Programmierbegriffe

Dauer

90 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- Grundlegende Programmierbegriffe wie Algorithmus, Bug und Schleife zu definieren
- Programmiervokabular im richtigen Kontext zu erkennen und anzuwenden
- die Bedeutung von Terminologie in der Programmierung zu verstehen
- Terminologiewissen in einfachen Programmierübungen anzuwenden
- effektiver über Programmierkonzepte zu kommunizieren

Benötigte Ausrüstung und Materialien

- Whiteboard und Marker
- Projektor oder Bildschirm zur Anzeige des Leitfadens
- Handouts des Leitfadens zur Programmier-Terminologie
- Computer oder Tablets mit Internetzugang



Aufgabe/Anweisungen

1. Beginne mit einer Präsentation und erkläre die Bedeutung des Verständnisses von Programmierbegriffen. Hebe hervor, wie es das Programmierlernen und die klare Kommunikation von Ideen unterstützt.
2. Zeige den Terminologie-Leitfaden über den Projektor. Gehe jeden Begriff durch, erkläre Definitionen und gib Beispiele. Ermutige die Schüler:innen, sich Notizen zu machen.
3. Teile die Klasse in Kleingruppen ein. Gib jeder Gruppe eine Auswahl von Programmierbegriffen aus dem Leitfaden. Bitte sie, für jeden Begriff eine Alltags-Analogie zu finden (z. B. einen Algorithmus mit einem Rezept vergleichen).
4. Jede Gruppe präsentiert ihre Analogien vor der Klasse, um das Verständnis zu vertiefen und verschiedene Sichtweisen kennenzulernen.
5. Gib eine einfache Programmieraufgabe (z. B. eine einfache Schleife schreiben). Bitte die Schüler:innen, die verwendeten Begriffe zu identifizieren und zu beschreiben.
6. Wiederhole die wichtigsten Begriffe und kläre offene Fragen. Erlaube den Schüler:innen, Fragen zu stellen, und gib bei Bedarf zusätzliche Erklärungen.

Lösung

Beende die Einheit mit einer Gruppendiskussion. Bitte die Schüler:innen, eigene Beispiele für die behandelten Begriffe zu teilen. Ermutige sie, die Begriffe in ihren eigenen Worten zu erklären, um das Verständnis zu festigen. Nutze die bereitgestellten einfachen Programmierbeispiele, um jeden Begriff in der Praxis zu zeigen, damit die Schüler:innen die Terminologie mit realen Codingsituationen verknüpfen können. Dieser Reflexionsmoment ermöglicht es den Schüler:innen, ihr Verständnis auszudrücken und Unklarheiten zu klären.



Kapitel 1.2 – Verschiedene Programmiersprachen und ihre Anwendungsbereiche

Abstract

Diese Anleitung bietet einen klaren und strukturierten Ansatz, um Schüler:innen die heute am häufigsten verwendeten Programmiersprachen näherzubringen. Es werden die Eigenschaften und typischen Anwendungen von fünf wichtigen Sprachen untersucht: Python, JavaScript, Java, C# und Ruby. Jede Sprache wird mit nachvollziehbaren Beispielen eingeführt, damit die Schüler:innen verstehen, wo und wie diese Sprachen eingesetzt werden. Ziel dieser Anleitung ist es, komplexe technische Konzepte zu vereinfachen, sodass auch Schüler:innen ohne Vorkenntnisse im Programmieren die Grundlagen erfassen können. Lehrkräfte werden durch praxisorientierte Unterrichtsaktivitäten und Diskussionsanregungen unterstützt.

Schlüsselwörter

Programmiersprachen; Python; JavaScript; Java; C#; Ruby

Dauer

Geschätzte Zeit: 90 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- fünf wichtige Programmiersprachen und deren Einsatzbereiche zu erkennen
 - jede Sprache dem Bereich zuzuordnen, in dem sie am häufigsten verwendet wird
 - die Rolle von Programmierung in verschiedenen Branchen (Web, Mobile, Games, Daten) zu verstehen
 - die Stärken und Anwendungsfälle von Python, JavaScript, Java, C# und Ruby zu vergleichen
 - persönliche Präferenzen basierend auf Spracheigenschaften und Einsatzbereichen zu teilen
-

Benötigte Ausrüstung und Materialien

- Computer mit Internetzugang
 - Projektor oder Smartboard
 - Ausgedruckte Übersichtsblätter zu den Programmiersprachen (optional)
 - Ein einfaches Arbeitsblatt für Zuordnungsübungen
 - Papier oder digitale Geräte für Quiz oder Reflexion
-



Aufgabe/Anweisungen

- **Einstieg:** Beginne mit der Frage, ob die Schüler:innen bereits von Programmiersprachen gehört haben und wo sie glauben, dass diese verwendet werden (Web, Spiele, Apps usw.). Notiere die Antworten an der Tafel als Diskussionsgrundlage.
- **Kurze Präsentation:** Stelle mithilfe des Projektors oder Whiteboards einen kurzen Überblick über die fünf Programmiersprachen vor (Python, JavaScript, Java, C#, Ruby). Nenne jeweils einige typische Einsatzgebiete oder Branchen.
- **Zuordnungsübung:** Verteile ein Arbeitsblatt mit 5 Sprachen und 10 Nutzungsszenarien. Die Schüler:innen sollen jede Sprache mit dem entsprechenden Anwendungsbereich verbinden (z. B. „Web-Apps → JavaScript“).
- **Gruppenarbeit:** Teile die Klasse in 5 Kleingruppen ein. Weise jeder Gruppe eine Programmiersprache zu. Die Gruppen sollen eine kurze „Präsentation“ erstellen, um zu erklären, warum ihre Sprache nützlich ist und in welchen Bereichen sie besonders gut eingesetzt werden kann (sie können zeichnen, schreiben oder nur sprechen).
- **Gruppenpräsentationen:** Jede Gruppe stellt ihre Ergebnisse der Klasse vor. Der Rest der Klasse darf einfache Rückfragen stellen.
- **Diskussionsimpuls:** Stelle die Frage: „Wenn du eine mobile App entwickeln würdest, welche Sprache würdest du wählen und warum?“ Einige Schüler:innen teilen ihre Gedanken mit der Klasse.
- **Mini-Quiz:** Führe ein kurzes Quiz mit 5 Fragen durch, um das Verständnis zu überprüfen. Die Fragen können Multiple-Choice oder einfache offene Fragen sein, z. B.: „Welche Sprache wird mit Unity verwendet, um Spiele zu entwickeln?“
- **Visuelle Zusammenfassung:** Erstelle gemeinsam mit der Klasse ein Plakat oder eine digitale Mindmap an der Tafel, die jede Sprache mit ihren wichtigsten Eigenschaften und typischen Anwendungsfeldern zusammenfasst.
- **Optionale Erweiterung (bei Zeit):** Zeige sehr kurze Videoclips oder Screenshots von Projekten, die mit jeder Sprache erstellt wurden (z. B. ein Python-Diagramm, ein Unity-Spiel, eine JavaScript-Webseite).
- **Abschlussfrage:** „Welche Sprache findest du am interessantesten und warum?“ Die Schüler:innen reflektieren still oder schreiben ihre Gedanken als „Exit-Ticket“ auf.



Lösung

In dieser Sitzung gibt es keine absolut richtigen oder falschen Antworten, da es darum geht, Vertrautheit und Selbstvertrauen im Umgang mit Programmiersprachen aufzubauen. Die Zuordnungsübung und die Gruppenpräsentationen zeigen, wie gut die Schüler:innen die technischen Merkmale mit realen Anwendungen verknüpfen können. Diskussion und Quiz dienen dazu, herauszufinden, welche Sprachen und Anwendungsfälle den Schüler:innen am besten im Gedächtnis geblieben sind. Beende die Stunde mit offenem Austausch oder schriftlichen Reflexionen, um das Lernen zu festigen und Neugier auf zukünftige Themen zu wecken.



Kapitel 1.3 – Programmieranwendungen: Ein umfassender Leitfaden für Einsteiger:innen

Abstract

Diese Anleitung hilft Lehrkräften, Schüler:innen vorzustellen, wie Programmieren zur Erstellung von Anwendungen wie Websites, Spielen und Software-Tools eingesetzt wird. Sie betont die Bedeutung des Verständnisses von Nutzerbedürfnissen und eines klaren Entwicklungsprozesses, dem sogenannten Software-Lebenszyklus. Anhand einfacher Beispiele und strukturierter Schritte lernen die Schüler:innen, wie Anwendungen geplant, entwickelt, getestet und verbessert werden. Diese Einheit zeigt außerdem, wie Programmierung mit Kreativität und Problemlösung im digitalen Alltag verbunden ist. Ziel der Anleitung ist es, den Schüler:innen ein Gesamtverständnis darüber zu vermitteln, wie echte Anwendungen entstehen.

Schlüsselwörter

- Programmieranwendungen
 - Nutzerbedürfnisse
 - Software-Lebenszyklus
 - Webentwicklung
 - Spieleentwicklung
 - Software-Engineering
-

Dauer

Geschätzte Zeit: 75 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- zu verstehen, was eine Programmieranwendung ist und wo sie verwendet wird
 - die Bedeutung der Nutzerbedürfnisse bei der App-Entwicklung zu erkennen
 - die grundlegenden Phasen des Softwareentwicklungszyklus zu identifizieren
 - zu erklären, wie Websites, Spiele und andere Software mit Nutzerfeedback entwickelt werden
 - Programmierung mit realen Beispielen wie E-Commerce, mobilen Apps und Spielen zu verknüpfen
-



Benötigte Ausrüstung und Materialien

- Computer mit Internetzugang
 - Projektor oder Smartboard
 - Whiteboard und Marker
 - Ausgedrucktes Arbeitsblatt (Phasen des Software-Lebenszyklus + Zuordnungsübung)
 - Haftnotizen oder farbige Karten
 - Optional: Beispiel-Screenshots von Apps oder Spielen (ausgedruckt oder digital)
-

Aufgabe/Anweisungen

- **Einführung (5 Min):** Frage die Schüler:innen: „Welche Apps oder Websites nutzt ihr jeden Tag?“ Schreibe die Antworten an die Tafel und leite daraus ab, dass all diese Tools durch Programmieren entstanden sind.
 - **Kurze Konzeptvorstellung (10 Min):** Erkläre, was eine Programmieranwendung ist. Verwende Beispiele wie einen Online-Shop, ein Spiel oder eine Lern-App. Halte es einfach und visuell.
 - **Nutzerzentriertes Denken (10 Min):** Führe die Idee ein, dass Apps für Menschen gemacht sind und Entwickler:innen verstehen müssen, was Nutzer wollen. Gib zwei kurze Beispiele: eine Shopping-Website (klare Suche, einfacher Checkout) und eine Sprachlern-App (spielerische, kurze Lerneinheiten).
 - **Gruppenaktivität (15 Min):** Teile die Schüler:innen in Kleingruppen ein. Gib jeder Gruppe eine andere Farbe von Haftnotizen und weise ihnen eine App-Art zu (z. B. Website, Spiel, Lern-App). Frage: „Was würden Nutzer:innen von dieser App erwarten?“ Jede Gruppe schreibt 3 Ideen auf und klebt sie an die Tafel.
 - **Überblick über den Software-Lebenszyklus (10 Min):** Stelle die fünf Hauptphasen vor: Planung, Design, Entwicklung, Testen, Bereitstellung. Verwende visuelle Hilfsmittel oder gedruckte Folien und erkläre alles in einfachen Worten.
 - **Zuordnungsübung (10 Min):** Gib den Schüler:innen ein Arbeitsblatt mit den Lebenszyklusphasen und realen Aufgaben (z. B. „Nutzer:innen fragen, was sie brauchen“ → Planung). Die Schüler:innen arbeiten paarweise an der Übung.
 - **Mini-Diskussion (5 Min):** Frage die Schüler:innen, welche Phase sie für die wichtigste halten und warum. 2–3 Schüler:innen können ihre Gedanken teilen.
 - **Abschluss-Quiz oder Exit-Frage (5 Min):** Frage: „Warum ist Nutzerfeedback bei der Entwicklung einer App wichtig?“ Die Schüler:innen können schriftlich antworten oder es laut sagen.
 - **(Optionale Erweiterung):** Zeige eine App-Oberfläche (Screenshot) und frage: „Welche Funktionen wurden hier wahrscheinlich für die Nutzer:innen geplant?“
-



Lösung

Die Gruppenaktivität und das Arbeitsblatt helfen den Schüler:innen zu verstehen, dass das Erstellen von Apps mehr ist als nur das Schreiben von Code – es geht darum, echte Probleme für Nutzer:innen zu lösen. Es gibt keine einzig richtige Antwort für einige Aufgaben, aber Lehrkräfte können das Denken der Schüler:innen durch ihre Arbeitsblattantworten und Gruppenvorschläge überprüfen. Die Abschlussdiskussion und die Exit-Frage bieten eine gute Gelegenheit, zu prüfen, ob die Schüler:innen den Zusammenhang zwischen Nutzer:innen, Planung und Entwicklung erkannt haben. Lehrkräfte können die Schüler:innen dazu ermutigen, weiter darüber nachzudenken, wie digitale Tools entstehen und wie Programmierung dabei hilft, sie zu gestalten.



Kapitel 2.1 – Vorteile einer Programmierkarriere: Deinen beruflichen Weg stärken

Abstract

Dieses Kapitel behandelt die verschiedenen Vorteile einer Karriere im Bereich Programmierung. Es geht auf die hohe Nachfrage nach Programmierkenntnissen auf dem Arbeitsmarkt, das Potenzial für hohe Gehälter, die Flexibilität des Remote-Arbeitens sowie die Möglichkeiten zum kontinuierlichen Lernen und Wachsen ein. Außerdem wird betont, wie Programmierkenntnisse Türen zu verschiedenen Branchen und Tätigkeitsfeldern öffnen und somit ein vielseitiges und wertvolles Kompetenzprofil darstellen.

Schlüsselwörter

- Programmieren
- Karriere
- Arbeitsmarkt
- Hohe Gehälter
- Remote-Arbeit
- Lebenslanges Lernen
- Kompetenzprofil
- Chancen

Dauer

Geschätzte Zeit: 60 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- die hohe Nachfrage nach Programmierkenntnissen auf dem heutigen Arbeitsmarkt zu verstehen
- das Potenzial für hohe Gehälter in Programmierberufen zu erkennen
- die Flexibilität und die Möglichkeiten der Remote-Arbeit zu beschreiben
- die Chancen für kontinuierliches Lernen und Wachstum im Bereich Programmierung zu identifizieren
- zu erklären, wie Programmierkenntnisse den Zugang zu verschiedenen Branchen und Rollen ermöglichen

Benötigte Ausrüstung und Materialien

- Computer mit Internetzugang
- Projektor oder Smartboard



1. **Einführung:** Beginne mit einer Diskussion über die Bedeutung von Programmieren auf dem modernen Arbeitsmarkt.
2. **Nachfrage auf dem Arbeitsmarkt:** Stelle Statistiken und Beispiele zur hohen Nachfrage nach Programmierkenntnissen vor.
3. **Verdienstpotezial:** Hebe das Potenzial für hohe Einkommen in Programmierberufen anhand realer Beispiele hervor.
4. **Flexibilität:** Besprich die Flexibilität der Remote-Arbeit und wie Programmierberufe diesen Vorteil häufig bieten.
5. **Lebenslanges Lernen:** Betone die Bedeutung von kontinuierlichem Lernen und persönlichem Wachstum in der Programmierwelt.
6. **Vielseitigkeit:** Erkläre, wie Programmierkenntnisse in verschiedenen Branchen und Funktionen angewendet werden können.
7. **Fragerunde:** Ermögliche den Schüler:innen, Fragen zu stellen und Gedanken über eine Karriere im Programmieren zu äußern.
8. **Praktische Aktivität:** Die Schüler:innen recherchieren und präsentieren verschiedene Karrierewege im Programmieren sowie deren Vorteile.
9. **Diskussion:** Führe eine Gruppendiskussion über die Erkenntnisse und Einblicke aus der Recherche durch.
10. **Abschluss:** Fasse die wichtigsten Punkte zusammen und ermutige die Schüler:innen, Programmieren als reale Berufsperspektive in Betracht zu ziehen.

Lösung

Für die praktische Aufgabe können die Schüler:innen eine Präsentation oder einen Bericht über verschiedene Karrierewege im Programmieren erstellen, einschließlich der jeweiligen Vorteile und Chancen. Wenn möglich, sollte ein Diskussionsmoment eingeplant werden, in dem die Schüler:innen ihre Ergebnisse vorstellen und Feedback von ihren Mitschüler:innen erhalten. Dies hilft, das Verständnis zu festigen und eine aktive Auseinandersetzung mit den Inhalten zu fördern.



Kapitel 2.2 – Karrierechancen im Bereich Programmierung: Vielfältige Wege entdecken

Abstract

Dieses Kapitel untersucht die vielfältigen Karrieremöglichkeiten, die Menschen mit Programmierkenntnissen offenstehen. Es beleuchtet verschiedene Rollen wie Softwareentwickler:in, Data Scientist, Cybersicherheits-Expert:in und mehr. Zudem wird erläutert, wie Programmierfähigkeiten in verschiedenen Branchen – darunter Technologie, Gesundheitswesen, Finanzen und Unterhaltung – angewendet werden können. Das Kapitel hebt das Potenzial für berufliches Wachstum und Spezialisierung in diesen Bereichen hervor.

Schlüsselwörter

- Programmierung
- Karrierechancen
- Softwareentwickler:in
- Data Scientist
- Cybersicherheit
- Branchen
- Berufliches Wachstum
- Spezialisierung

Dauer

Geschätzte Zeit: 60 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- verschiedene Karrierewege für Personen mit Programmierkenntnissen zu identifizieren
- die Aufgaben und Verantwortlichkeiten von Softwareentwickler:innen, Data Scientists und Cybersicherheits-Expert:innen zu verstehen
- die Anwendung von Programmierkenntnissen in verschiedenen Branchen wie Technologie, Gesundheitswesen, Finanzen und Unterhaltung zu erkennen
- das Potenzial für berufliches Wachstum und Spezialisierung in programmierbezogenen Bereichen zu analysieren
- die vielfältigen Möglichkeiten zu beschreiben, die Programmierkenntnisse in unterschiedlichen Sektoren bieten



Benötigte Ausrüstung und Materialien

- Computer mit Internetzugang
- Projektor oder Smartboard

Aufgabe/Anweisungen

1. **Einführung:** Beginne mit einem Überblick über die verschiedenen Karrierewege, die Programmierer:innen offenstehen.
2. **Karrierewege:** Besprich konkrete Rollen wie Softwareentwickler:in, Data Scientist und Cybersicherheits-Expert:in, einschließlich ihrer Aufgaben und benötigten Fähigkeiten.
3. **Branchenspezifische Anwendungen:** Erkläre, wie Programmierkenntnisse in verschiedenen Branchen wie Technologie, Gesundheitswesen, Finanzen und Unterhaltung eingesetzt werden.
4. **Berufliches Wachstum:** Hebe das Potenzial für Karriereentwicklung und Spezialisierung in programmierbezogenen Berufen hervor.
5. **Interaktive Diskussion:** Ermutige die Schüler:innen, ihre Gedanken darüber zu teilen, welche Karrierewege sie besonders interessieren und warum.
6. **Rechercheaufgabe:** Weisen Sie den Schüler:innen zu, einen bestimmten Karriereweg im Bereich Programmierung zu recherchieren und die damit verbundenen Chancen zu analysieren.
7. **Präsentation:** Die Schüler:innen präsentieren ihre Ergebnisse der Klasse – mit Fokus auf Rollen, Branchen und Entwicklungspotenzial.
8. **Q&A-Runde:** Moderiere eine Frage-Antwort-Runde, um offene Fragen zu klären.
9. **Gruppendiskussion:** Führe eine Diskussion über die Vielfalt der Karrierechancen im Bereich Programmierung und die Perspektiven der Schüler:innen.
10. **Abschluss:** Fasse die wichtigsten Punkte zusammen und motiviere die Schüler:innen, Programmierkarrieren weiter zu erkunden.

Lösung

Für die Rechercheaufgabe sollen die Schüler:innen einen detaillierten Bericht oder eine Präsentation zu einem gewählten Karriereweg im Bereich Programmierung erstellen – inklusive der damit verbundenen Chancen und des Wachstumspotenzials. Während der Q&A-Session können sie ihre Ergebnisse diskutieren und Einsichten austauschen. Dies unterstützt eine aktive Auseinandersetzung mit dem Thema und vertieft das Verständnis für die vielfältigen Möglichkeiten in Programmierberufen.



Kapitel: [2.3 Vielfältige Anwendungen des Programmierens: Deine Karrierehorizonte erweitern](#)

Abstract:

Programmieren ist längst nicht mehr auf traditionelle IT-Rollen beschränkt, sondern hat sich zu einer entscheidenden Fähigkeit in verschiedenen Branchen entwickelt. Dieses Kapitel untersucht die wachsende Rolle des Programmierens in nicht-technischen Berufen wie Marketing, Finanzen, Gesundheitswesen, Bildung und mehr. Programmierung befähigt Fachkräfte, die Effizienz zu steigern, Aufgaben zu automatisieren und bessere Entscheidungen zu treffen. Besonders wertvoll ist sie in Berufen wie Datenanalyse, digitalem Marketing, Projekt- und Produktmanagement sowie technischem Schreiben. Darüber hinaus zeigt die Anwendung von Programmierung in Bereichen wie FinTech, EdTech und HealthTech ihre Wirkung bei der Förderung von Innovation und der Optimierung von Prozessen. Das Verständnis dieser vielfältigen Anwendungen ermöglicht es Einzelpersonen, Programmierkenntnisse für berufliches Wachstum und verbesserte Berufsaussichten in einer digital geprägten Wirtschaft zu nutzen.

Schlüsselwörter:

Programmieranwendungen, nicht-technische Rollen, Automatisierung, FinTech, EdTech, HealthTech, Karrieremöglichkeiten, Effizienz

Dauer:

60–90 Minuten

Lernziele:

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- die Bedeutung von Programmierkenntnissen in verschiedenen nicht-technischen Bereichen und deren Rolle bei der Effizienzsteigerung zu verstehen
 - zu erkennen, wie Programmieren Fachkräften in Rollen wie Datenanalytist:innen, Marketingfachleuten, Projektmanager:innen und technischen Redakteur:innen zugutekommt
 - realistische Anwendungsbeispiele von Programmieren in Forschung, Finanzen, Bildung und Gesundheitswesen zu erkunden
 - die Wege zu analysieren, wie Programmierung Produktivität durch Automatisierung und bessere Entscheidungsfindung verbessert
 - die Karrieremöglichkeiten zu erkennen, die Programmierkenntnisse in unterschiedlichen Branchen bieten
-

Benötigte Ausrüstung und Materialien:

- Computer mit Internetzugang



- Projektor oder Smartboard
- Vorbereitete Fallstudien oder Beispiele für Programmieranwendungen in verschiedenen Bereichen
- Online-Coding-Tools wie Python- oder JavaScript-Umgebungen (optional)

Aufgabe/Anweisungen:

1. Beginne mit einer einführenden Diskussion darüber, wie Programmieren über traditionelle IT-Rollen hinausgeht.
2. Präsentiere eine Slideshow oder ein Video, das verschiedene Branchen zeigt, in denen Programmierung bedeutende Auswirkungen hat.
3. Diskutiere Fallstudien über Programmieranwendungen in der Datenanalyse, im Marketing und im Gesundheitswesen.
4. Teile die Schüler:innen in Gruppen ein und weise jeder Gruppe eine andere Branche zu (Finanzen, Marketing, Gesundheitswesen, Bildung usw.).
5. Lass jede Gruppe recherchieren und präsentieren, wie Programmieren in ihrer zugewiesenen Branche angewendet wird.
6. Führe – wenn möglich – eine praktische Demonstration durch, z. B. eine einfache Automatisierungsaufgabe in Excel, Python oder einem anderen zugänglichen Tool.
7. Ermutige die Schüler:innen dazu, darüber nachzudenken, wie Programmieren in ihren eigenen beruflichen Zielen nützlich sein könnte.
8. Leite eine Gruppendiskussion, in der die Schüler:innen ihre Einsichten teilen, wie sie die Rolle des Programmierens in der Arbeitswelt der Zukunft sehen.
9. Beende die Sitzung mit einer Q&A-Runde, um offene Fragen zu klären und zur weiteren Erkundung anzuregen.

Lösung:

Für die praktischen Aktivitäten sollen die Gruppen ihre Ergebnisse zu den Programmieranwendungen in der ihnen zugewiesenen Branche präsentieren. Die Diskussionen sollen sich auf die Vorteile und Herausforderungen der Integration von Programmierung in nicht-technische Rollen konzentrieren. Falls eine praktische Programmieraufgabe enthalten ist, sollen die Schüler:innen ihr Automatisierungsskript oder ihre Datenvisualisierung demonstrieren. Eine abschließende Reflexion soll die Schüler:innen dazu anregen, über das Gelernte nachzudenken und weitere Bildungsmöglichkeiten im Bereich Programmieren zu erkunden.



Kapitel 3.1 – Erfolgsgeschichten aus der Programmierung: Inspirierende Wege von Tech-Innovator:innen

Abstract

Diese Anleitung stellt den Schüler:innen inspirierende Erfolgsgeschichten aus der Welt der Programmierung und Technologie vor. Sie beleuchtet die Werdegänge bekannter Start-ups wie Airbnb und Uber sowie einzelner Programmierer:innen, die durch Innovation, Ausdauer und Können einen Unterschied gemacht haben. Durch diese Geschichten lernen die Schüler:innen etwas über Unternehmertum, Problemlösung und die vielen Wege, die eine Karriere im Programmieren bieten kann. Die Einheit regt zur Reflexion darüber an, was einen erfolgreichen Entwickler oder eine erfolgreiche Entwicklerin ausmacht und wie jede:r seine eigene Reise beginnen kann. Lehrkräfte begleiten die Schüler:innen durch Storytelling, Diskussionen und kollaborative Aktivitäten, um Neugier und Motivation zu fördern.

Schlüsselwörter

- Programmieren
- Innovation
- Start-ups
- Erfolgsgeschichten
- Tech-Unternehmer:innen
- Entwickler:innen

Dauer

Geschätzte Zeit: 60 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- zentrale Erfolgsgeschichten aus der Programmier- und Tech-Branche zu erkennen
- zu verstehen, wie Programmieren zu Unternehmertum und Innovation führen kann
- die Eigenschaften und Gewohnheiten erfolgreicher Entwickler:innen zu benennen
- über die Rolle von Kreativität, Teamarbeit und Lernen in Programmierkarrieren zu reflektieren
- zu diskutieren, wie Programmieren zur Lösung realer Probleme eingesetzt wird

Benötigte Ausrüstung und Materialien

- Computer mit Internetzugang
- Projektor oder Smartboard



Aufgabe/Anweisungen

• **Aufwärmphase (5 Min):** Frage die Schüler:innen: „Weißt du, wer deine Lieblings-App oder dein Lieblingsspiel entwickelt hat?“ Notiere die Antworten und erkläre, dass hinter jeder App eine Person oder ein Team steht, das mit einer Idee begann und diese mit Code verwirklicht hat.

• **Geschichtenzeit (10 Min):** Teile kurze, vereinfachte Geschichten von Unternehmen wie Airbnb, Uber und Stripe sowie von Einzelpersonen wie Linus Torvalds oder Sarah Drasner. Hebe hervor, welches Problem sie gelöst haben, wie sie angefangen haben und was sie besonders gemacht hat.

• **Diskussionsimpuls (5 Min):** Frage: „Was haben all diese Geschichten gemeinsam?“ Führe die Schüler:innen dahin, gemeinsame Elemente wie Kreativität, Lernen aus Fehlern, Teamarbeit oder das Eingehen auf Nutzerbedürfnisse zu erkennen.

• **Arbeitsblatt-Aktivität (10–15 Min):** Verteile ein kurzes Arbeitsblatt mit Fragen wie:
• Welche Geschichte hat dich am meisten inspiriert und warum?
• Welche Fähigkeiten oder Eigenschaften haben diesen Menschen zum Erfolg verholfen?
• Welches Problem würdest du gerne mit Code lösen?

• **Gruppenaktivität (15 Min):** Teile die Schüler:innen in Kleingruppen ein. Bitte sie, sich eine kurze „zukünftige Erfolgsgeschichte“ zu einem Coding-Idee auszudenken und aufzuschreiben, die sie entwickeln würden – ein Spiel, eine App oder ein Tool, das ein Problem löst. Jede Gruppe gibt ihrem Projekt einen Namen und beschreibt, was es tut.

• **Präsentation (10 Min):** Die Gruppen stellen ihre Idee der Klasse vor. Die Lehrkraft oder andere Schüler:innen können einfache Fragen stellen wie „Wer würde es benutzen?“ oder „Wie würde es den Menschen helfen?“

• **Abschlussfrage (5 Min):** „Was hast du heute über den Weg zum erfolgreichen Coder gelernt?“ Die Schüler:innen können laut antworten oder ihre Gedanken auf einem Zettel notieren.

Lösung

Das Hauptziel dieser Aktivität ist es nicht, sich Namen oder Daten zu merken, sondern den Schüler:innen zu vermitteln, dass Erfolg im Programmieren möglich und persönlich ist. Durch Storytelling und kreative Gruppenarbeit erkennen sie, dass großartige Ideen oft aus einfachen Anfängen entstehen. Die Arbeitsblätter und Präsentationen bieten eine Möglichkeit, das Engagement und das Verständnis zu bewerten. Lehrkräfte werden ermutigt, die Bedeutung von Ausprobieren, Lernen aus Fehlern und Neugierde hervorzuheben – die Schlüsselzutaten jeder Tech-Erfolgsgeschichte.



Kapitel: [3.2 Zukünftige Programmier Trends: Was die Softwareentwicklung erwartet](#)

Abstract

Dieses Kapitel untersucht die wichtigsten technologischen Trends, die die Zukunft der Softwareentwicklung prägen, darunter Künstliche Intelligenz (KI), Maschinelles Lernen (ML), Quantencomputing, Blockchain, das Internet der Dinge (IoT) sowie Augmented Reality (AR) und Virtual Reality (VR). Das Kapitel zeigt auf, wie KI und ML die Effizienz des Programmierens durch automatisierte Tools und prädiktive Benutzeroberflächen steigern. Es geht auch auf die Rolle des Quantencomputings bei der Lösung komplexer Rechenprobleme ein und beleuchtet den Einsatz von Blockchain jenseits von Kryptowährungen zur Verbesserung der Software-Sicherheit und -Transparenz. Darüber hinaus behandelt das Kapitel die wachsende Bedeutung von IoT, Edge Computing sowie immersiven AR/VR-Technologien bei der Gestaltung zukünftiger Softwarelösungen. Das Verständnis dieser Trends befähigt die Lernenden, sich an die sich wandelnden Anforderungen der Technologiebranche anzupassen.

Schlüsselwörter

Künstliche Intelligenz (KI), Maschinelles Lernen (ML), Quantencomputing, Blockchain, Internet der Dinge (IoT), Augmented Reality (AR), Virtual Reality (VR), Edge Computing

Dauer:

60–90 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- die Rolle von KI und ML in der Softwareentwicklung zu verstehen, einschließlich automatisierter Tools und prädiktiver Analytik
 - wichtige Programmiersprachen und Plattformen für KI und ML zu benennen, wie Python, R, TensorFlow und PyTorch
 - Karrieremöglichkeiten im Bereich KI und ML zu erkennen, einschließlich Rollen in Ethik und Governance von KI
 - die Bedeutung des Quantencomputings und dessen potenzieller Anwendungen zur Lösung komplexer Probleme zu erklären
 - die Anwendung der Blockchain-Technologie zur Verbesserung von Sicherheit, Transparenz und Effizienz in der Softwareentwicklung zu analysieren
 - die Auswirkungen von IoT, Edge Computing, AR und VR auf die Transformation von Softwarelösungen in verschiedenen Branchen zu bewerten
-



Benötigte Ausrüstung und Materialien

- Computer mit Internetzugang
- Projektor oder Smartboard
- Programmierumgebungen (Python, R, TensorFlow, PyTorch)
- Fallstudien zu KI-gestützten Programmier-Tools, Blockchain-Anwendungen und AR/VR-Erfahrungen

Aufgabe/Anweisungen

1. **Einführung in zukünftige Programmertrends** – Kurze Diskussion darüber, wie KI, ML, Quantencomputing, Blockchain, IoT und AR/VR die Softwareentwicklung beeinflussen
2. **Erkundung von KI und ML** – Vorführung von KI-gestützten Programmier-Tools wie GitHub Copilot, Tabnine oder KI-basierten Debugging-Plattformen
3. **Programmieren mit KI und ML** – Anleitung zu einer einfachen Übung im maschinellen Lernen mit Python und TensorFlow
4. **Grundlagen des Quantencomputings** – Erklärung der Grundprinzipien des Quantencomputings und seiner Anwendungen in Kryptographie und Problemlösung
5. **Blockchain in der Softwareentwicklung** – Beispiele für dezentrale Anwendungen (dApps) und Diskussion über die Rolle der Blockchain bei der Software-Sicherheit
6. **IoT und Edge Computing** – Vorstellung realer Anwendungen von IoT und Edge Computing in Branchen wie Gesundheitswesen und Smart Cities
7. **AR/VR in der Softwareentwicklung** – Möglichkeit für Schüler:innen, AR/VR-Anwendungen zu erkunden und deren potenzielle Auswirkungen auf Gaming, Schulungen und Simulationen zu verstehen
8. **Diskussion und Reflexion** – Schüler:innen diskutieren, welche Technologie sie für am vielversprechendsten halten und wie sie ihre zukünftige Karriere beeinflussen könnte
9. **Abschließende Projektaufgabe** – Die Schüler:innen schlagen eine Softwarelösung vor, die eine oder mehrere der behandelten Technologien beinhaltet
10. **Abschluss und Auswertung** – Zusammenfassung der wichtigsten Erkenntnisse und Diskussion über Branchentrends und mögliche zukünftige Lernpfade

Lösung

- Bei der KI- und ML-Übung trainieren die Schüler:innen ein einfaches ML-Modell zur Mustererkennung in einem Datensatz.
- Im Blockchain-Teil erkunden sie ein einfaches Beispiel für einen Smart Contract mit Ethereum.
- Bei der abschließenden Projektaufgabe stellen die Schüler:innen ihre vorgeschlagene Zukunfts-Softwarelösung vor, gefolgt von einer Klassendiskussion über Machbarkeit und Auswirkungen.
- Abschließend findet eine offene Diskussion statt, in der die Schüler:innen ihre Gedanken darüber teilen, wie diese Trends ihre Karriere in der Softwareentwicklung beeinflussen werden.



Kapitel [3.3 – IoT-Technologietrends: Die Zukunft der Vernetzung](#)

Abstract

Diese Anleitung hilft Lehrkräften, Schüler:innen das Konzept des Internets der Dinge (IoT) näherzubringen und zu erklären, wie es alltägliche Gegenstände miteinander verbindet, um unser Leben zu verbessern. Anhand realer Beispiele wie Smart Homes, tragbarer Technologie und intelligenter Landwirtschaft verstehen die Schüler:innen, wie Geräte Daten sammeln und nutzen. Die Anleitung behandelt auch zentrale Herausforderungen wie Datenschutz, Energieverbrauch und Sicherheit. Die Schüler:innen erkunden, wie das IoT heute funktioniert und wie es sich in Zukunft entwickeln könnte. Die Einheit fördert Neugier und kritisches Denken über die Technologie, die uns umgibt.

Schlüsselwörter

- IoT
 - smarte Geräte
 - Vernetzung
 - Daten; Sensoren
 - Datenschutz
 - smarte Städte
-

Dauer

Geschätzte Zeit: 75 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- zu erklären, was das Internet der Dinge (IoT) ist und wie es funktioniert
 - Beispiele für IoT im Alltag und in verschiedenen Branchen zu erkennen
 - grundlegende Herausforderungen im Zusammenhang mit IoT (Sicherheit, Energieverbrauch, Datenschutz) zu verstehen
 - zukünftige Trends im IoT wie KI, Edge Computing und Blockchain zu erkennen
 - darüber zu reflektieren, wie IoT ihr eigenes Leben und die Gesellschaft beeinflusst
-

Benötigte Ausrüstung und Materialien

- Computer mit Internetzugang
 - Projektor oder Smartboard
-



Aufgabe/Anweisungen

- **Einführung (5 Min):** Beginne mit der Frage: „Hast du schon einmal ein smartes Gerät benutzt?“ Beispiele können Smartwatches, Sprachassistenten oder Smart-TVs sein. Nutze die Antworten, um das IoT-Konzept einzuführen.
- **Einfache Erklärung (10 Min):** Erkläre, wie IoT funktioniert: Geräte mit Sensoren sammeln Daten und kommunizieren über das Internet miteinander. Verwende einfache, reale Beispiele: Ein Fitness-Tracker zählt Schritte und sendet die Daten an dein Handy.
- **Interaktive Zuordnungsaktivität (15 Min):** Verteile gedruckte Karten mit Namen von IoT-Geräten und ein separates Set mit Beschreibungen (z. B. „Smartwatch“ – „Überwacht deine Herzfrequenz und sendet Daten an eine App“). In Kleingruppen ordnen die Schüler:innen Geräte den Funktionen zu.
- **Mini-Präsentation (10 Min):** Präsentiere kurze Beschreibungen verschiedener IoT-Anwendungsbereiche: smarte Häuser, Städte, Landwirtschaft, Gesundheit. Nutze Bilder und einprägsame Ein-Satz-Zusammenfassungen, um es visuell und einfach zu halten.
- **Gruppendiskussion (10 Min):** Jede Gruppe wählt einen Bereich (z. B. smarte Landwirtschaft) und beantwortet: „Wie hilft hier das IoT?“ Notiere ihre Gedanken an der Tafel. Diskutiere mögliche Herausforderungen (z. B. was passiert, wenn das System ausfällt?).
- **Zukunfts-Technologie-Brainstorming (10 Min):** Sprich kurz über neue Trends: KI in Geräten, Blockchain zur Sicherheit oder Edge Computing. Frage: „Was würdest du mit dem IoT erfinden?“ Gib den Schüler:innen 3–5 Minuten Zeit, eine Idee zu schreiben oder zu zeichnen.
- **Galerie-Rundgang (10 Min):** Die Schüler:innen hängen ihre Zeichnungen oder Ideen an die Wand. Alle gehen herum, lesen die Ideen der anderen und wählen die interessanteste.
- **Abschließende Reflexion (5 Min):** Frage: „Welcher Teil des IoT wird deiner Meinung nach in der Zukunft am wichtigsten sein?“ Die Schüler:innen können laut antworten oder einen Satz als Exit-Ticket aufschreiben.

Lösung

Diese Aktivität hilft den Schüler:innen, das IoT auf eine Weise zu verstehen, die mit ihrem Alltag verknüpft ist. Das Zuordnungsspiel und das Gruppen-Brainstorming regen kreatives Denken an und zeigen gleichzeitig reale Anwendungen. Ihre Antworten in der Abschlussreflexion und den Diskussionen zeigen, ob sie sowohl die Vorteile als auch die Herausforderungen dieser Technologie verstanden haben. Der Fokus liegt nicht auf technischer Tiefe, sondern auf dem Aufbau von Bewusstsein, Neugier und durchdachten Fragen zur Rolle smarter Technologien in der Zukunft.



Kapitel [3.4 – Innovationsorientierte Erfolgsgeschichten: Inspirierende Unternehmen](#)

Abstract

Diese Anleitung stellt den Schüler:innen reale Beispiele vor, wie Programmieren zur Förderung von Innovation in verschiedenen Sektoren eingesetzt wird. Sie enthält inspirierende Geschichten aus den Bereichen Fahrdienstvermittlung, Telemedizin, Finanzen, Landwirtschaft und Umweltinitiativen. Die Schüler:innen erkunden, wie aus Ideen durch Code Lösungen werden – und wie Zusammenarbeit, Problemlösung und Kreativität technischen Erfolg ermöglichen. Ziel der Aktivität ist es, das Verständnis für den Zusammenhang zwischen Technologie und realen Auswirkungen zu fördern und die Schüler:innen zu ermutigen, sich selbst als zukünftige Veränderungsträger:innen zu sehen.

Schlüsselwörter

- Innovation
 - Programmieren
 - Start-ups
 - Problemlösung
 - FinTech
 - Soziale Wirkung
 - Technologie
-

Dauer

Geschätzte Zeit: 75 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- zu verstehen, wie Programmieren zur Lösung realer Probleme eingesetzt wird
 - Unternehmen und Projekte zu benennen, die Code für Innovation und sozialen Nutzen einsetzen
 - die wichtigsten Eigenschaften zu beschreiben, die Innovation über technische Fähigkeiten hinaus unterstützen
 - über die sozialen, ökologischen und wirtschaftlichen Auswirkungen technischer Lösungen zu reflektieren
 - eine eigene, auf Programmierung basierende Lösung für ein alltägliches Problem zu entwerfen
-



Benötigte Ausrüstung und Materialien

- Computer mit Internetzugang
 - Projektor oder Smartboard
-

Aufgabe/Anweisungen

• **Einstiegsfrage (5 Min):** Frage die Schüler:innen: „Kann Code die Welt verändern? Wie?“ Erlaube ein paar kurze Antworten, um den Rahmen zu setzen.

• **Kurze Präsentation (10 Min):** Teile vereinfachte Fallbeispiele aus den Handouts – z. B. Über (Transport), M-Pesa (Finanzen), Ocean Cleanup (Umwelt). Hebe hervor, wie jedes Projekt ein reales Bedürfnis erkannt und mit Code gelöst hat.

• **Gruppen-Brainstorming (10 Min):** Teile die Klasse in Kleingruppen ein. Frage: „Welches Problem in eurer Schule oder Gemeinde würdet ihr gerne lösen?“ Die Gruppen brainstormen und notieren ihre Ideen.

• **Projekt-Design-Aktivität (15 Min):** Mithilfe eines Arbeitsblatts oder Vorlagenblatts beschreiben die Gruppen ihre Idee, als wäre sie ein Start-up:

- Was ist das Problem?
- Wer würde es nutzen?
- Was macht es?
- Wie könnte Code helfen?

• **Pitch-Zeit (15 Min):** Jede Gruppe hält eine kurze, informelle 1–2-minütige Präsentation ihrer Idee. Kreativität und Storytelling werden ermutigt.

• **Klassenfeedback (10 Min):** Nach jedem Pitch stellt die Klasse einfache Fragen wie: „Wer würde das benutzen?“ oder „Was wäre eine coole Funktion?“ Das stärkt Selbstvertrauen und Beteiligung.

• **Abschluss (10 Min):** Schreibe an die Tafel: „Innovation = Code + ?“ Die Schüler:innen füllen die Lücke mit Antworten (Teamarbeit, Kreativität, Empathie usw.) – basierend auf dem, was sie gelernt haben.

Lösung

Bei dieser Aktivität gibt es keine richtigen oder falschen Antworten, da es darum geht, kreatives Denken zu fördern und den Schüler:innen zu zeigen, dass Technologie reale Probleme lösen kann. Durch die Auseinandersetzung mit inspirierenden Geschichten und die Entwicklung eigener Ideen sehen die Schüler:innen Programmieren nicht nur als technische Fähigkeit, sondern als Werkzeug für Veränderung. Lehrkräfte können das Verständnis durch die Gruppenbeteiligung, geteilte Ideen und die Verbindung von Innovation zum Alltag einschätzen. Die abschließende Reflexion hilft, die Botschaft zu festigen: Jeder mit einer Idee und grundlegenden Programmierkenntnissen kann etwas verändern.



Kapitel [4.1 – Programme zur Weiterbildung: Entfalte dein Potenzial](#)

Abstract

Die Technologie entwickelt sich rasant weiter, wodurch kontinuierliche Weiterbildung und Kompetenzentwicklung für Fachkräfte im IKT-Bereich unerlässlich werden. Dieses Kapitel beleuchtet die Bedeutung des lebenslangen Lernens und betont spezialisierte Schulungen, Zertifizierungen und die Beherrschung neuer Technologien, um wettbewerbsfähig zu bleiben. Es hebt die Rolle von Networking für das berufliche Wachstum hervor, die Wichtigkeit von Anpassungsfähigkeit, um mit technologischen Entwicklungen Schritt zu halten, sowie die Vorteile der Zusammenarbeit mit Start-ups für praxisnahe Erfahrungen. Fachkräfte, die aktiv kontinuierliches Lernen betreiben, verbessern ihre Karrierechancen, bleiben relevant und machen ihre Fähigkeiten zukunftssicher.

Schlüsselwörter

Weiterbildung, Kompetenzentwicklung, berufliches Wachstum, Networking, Anpassungsfähigkeit, Zertifizierungen, Vermarktbarkeit, Innovation

Dauer

60–90 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- die Bedeutung von kontinuierlicher Weiterbildung und beruflicher Entwicklung im IKT-Sektor zu erkennen
 - verschiedene Wege zur Kompetenzentwicklung zu benennen, einschließlich spezialisierter Schulungen und anerkannter Zertifizierungen
 - die Rolle von Networking für den beruflichen Aufstieg zu analysieren und zu lernen, wie man berufliche Kontakte effektiv nutzt
 - die Bedeutung von Anpassungsfähigkeit für die Relevanz in der Technologiebranche zu verstehen
 - die Vorteile der Mitarbeit in Start-ups und innovativen Projekten für praktische Erfahrungen und berufliches Wachstum zu bewerten
-



Benötigte Ausrüstung und Materialien

- Computer mit Internetzugang
- Projektor oder Smartboard
- Gedruckte oder digitale Handouts zu spezialisierten Schulungen und Zertifizierungen
- Zugang zu professionellen Netzwerkplattformen (LinkedIn, branchenspezifische Foren)

Aufgabe/Anweisungen

- 1.1 **Einleitende Diskussion:** Beginne mit einer Klassendiskussion darüber, wie sich Technologie im letzten Jahrzehnt entwickelt hat und welche Auswirkungen dies auf Berufsbilder im IKT-Bereich hatte.
2. **Fallstudienanalyse:** Präsentiere reale Beispiele von Fachkräften, die ihre Karriere durch kontinuierliches Lernen und Zertifizierungen vorangebracht haben.
3. **Recherche-Aktivität:** Die Schüler:innen recherchieren verschiedene Online-Kurse und Zertifizierungsprogramme, die für ihr Fachgebiet relevant sind.
4. **Networking-Simulation:** Organisiere eine simulierte Networking-Veranstaltung, bei der die Schüler:innen das Gespräch mit Kolleg:innen, Mentor:innen und Branchenexpert:innen üben.
- 5.1 **Workshop zur Anpassungsfähigkeit:** Diskutiere neue Technologien wie KI, Blockchain und Cloud-Computing. Die Schüler:innen sollen deren Einfluss auf den Arbeitsmarkt recherchieren und präsentieren.
- 6.1 **Interaktives Quiz:** Führe ein Quiz zu den zentralen Konzepten von Weiterbildung, Kompetenzentwicklung und beruflichem Wachstum durch.
7. **Praxisprojekt:** Die Schüler:innen erstellen einen persönlichen Karriereentwicklungsplan, der Zertifizierungen, Networking-Strategien und Verbesserungsbereiche umfasst.
8. **Abschließende Reflexion:** Gruppen-Diskussion über die Lernerfahrungen und wie sie das kontinuierliche Lernen in ihre Karriere integrieren möchten.

Lösung

- **Ergebnis der Recherche-Aktivität:** Die Schüler:innen erstellen eine Liste von mindestens drei Online-Trainingsprogrammen oder Zertifizierungen, die für ihr Fachgebiet relevant sind.
- **Networking-Simulation:** Die Schüler:innen üben und zeigen effektive Networking-Techniken.
- **Praxisprojekt:** Jede:r Schüler:in präsentiert seinen/ihren Karriereentwicklungsplan, in dem Lern- und Netzwerkstrategien hervorgehoben werden.
- **Diskussion und Reflexion:** Die Schüler:innen teilen ihre Gedanken über lebenslanges Lernen und Wettbewerbsfähigkeit in der Branche.



Kapitel [4.2 – Tipps zum Personal Branding für Programmierer:innen: Dein professionelles Image stärken](#)

Abstract

Dieses Kapitel gibt wertvolle Tipps für Programmierer:innen, wie sie ihre persönliche Marke aufbauen und verbessern können. Es behandelt Strategien wie das Erstellen eines Online-Portfolios, die Nutzung von Social Media, Networking innerhalb der Tech-Community sowie kontinuierliches Lernen und Wissensaustausch. Das Kapitel betont die Bedeutung einer starken persönlichen Marke zur Gewinnung von Jobchancen, zur Etablierung von Glaubwürdigkeit und zur Weiterentwicklung der eigenen Karriere.

Schlüsselwörter

- Personal Branding
- Programmierer:innen
- Online-Portfolio
- Social Media
- Networking
- Tech-Community
- Lebenslanges Lernen
- Beruflicher Aufstieg

Dauer

Geschätzte Zeit: 60 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- die Bedeutung von Personal Branding für Programmierer:innen zu verstehen
- ein Online-Portfolio zu erstellen und zu pflegen, um ihre Fähigkeiten und Projekte zu präsentieren
- Social Media effektiv zu nutzen, um ihre persönliche Marke aufzubauen
- sich innerhalb der Tech-Community zu vernetzen, um Kontakte und Chancen zu schaffen
- kontinuierlich zu lernen und Wissen zu teilen, um ihre Glaubwürdigkeit und Karrierechancen zu verbessern

Benötigte Ausrüstung und Materialien

- Computer mit Internetzugang
- Projektor oder Smartboard



Aufgabe/Anweisungen

1. **Einführung:** Beginne mit einem Überblick über Personal Branding und dessen Bedeutung für Programmierer:innen.
2. **Online-Portfolio:** Bespreche die Elemente eines starken Online-Portfolios und wie man eines erstellt.
3. **Social Media:** Erkläre die Rolle von Social Media im Personal Branding und Strategien für die effektive Nutzung.
4. **Networking:** Hebe die Wichtigkeit des Networkings in der Tech-Community hervor und wie man sinnvolle Verbindungen aufbaut.
5. **Lebenslanges Lernen:** Betone die Notwendigkeit des kontinuierlichen Lernens und Wissensaustauschs zur Pflege und Weiterentwicklung der eigenen Marke.
6. **Interaktive Aktivität:** Die Schüler:innen erstellen einen Entwurf ihres Online-Portfolios, einschließlich ihrer Fähigkeiten, Projekte und Kontaktdaten.
7. **Social-Media-Strategie:** Die Schüler:innen entwickeln eine Social-Media-Strategie mit Plattformen, Inhaltsarten und Interaktionstaktiken.
8. **Networking-Plan:** Die Schüler:innen erstellen einen Plan zum Networking in der Tech-Community, einschließlich Veranstaltungen, Online-Gruppen und Berufsverbänden.
9. **Q&A-Session:** Gelegenheit für Fragen und Diskussion der Branding-Strategien der Schüler:innen.
10. **Abschluss:** Fasse die wichtigsten Punkte zusammen und motiviere die Schüler:innen, aktiv an ihrem Personal Branding zu arbeiten.

Lösung

Für die interaktive Aktivität sollen die Schüler:innen einen Entwurf ihres Online-Portfolios erstellen und eine Social-Media-Strategie entwickeln. Während der Q&A-Session können sie ihre Strategien vorstellen und Feedback von Mitschüler:innen und der Lehrkraft erhalten. Dies hilft ihnen, die praktischen Aspekte des Personal Brandings zu verstehen und die Tipps effektiv anzuwenden



Kapitel [4.3 – Networking und Community Engagement: Sinnvolle Verbindungen aufbauen](#)

Abstract

Networking und Community Engagement sind grundlegende Elemente für berufliches Wachstum und Innovation im Technologiesektor. Dieses Kapitel untersucht, wie die aktive Teilnahme an beruflichen Communities Zusammenarbeit, Kompetenzentwicklung und Karrierechancen fördert. Es stellt Strategien vor, um Sichtbarkeit zu erhöhen, digitale Plattformen effektiv zu nutzen und sich auf sinnvolle Weise zu vernetzen. Das Verständnis des Wertes beruflicher Netzwerke hilft Einzelpersonen, relevant zu bleiben, zur Weiterentwicklung der Branche beizutragen und starke Unterstützungsnetzwerke innerhalb des Tech-Ökosystems aufzubauen.

Schlüsselwörter

Networking, Community Engagement, berufliche Entwicklung, Zusammenarbeit, Online-Plattformen, Innovation, Karrierewachstum, Sichtbarkeit

Dauer

60–90 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- die strategische Bedeutung von Networking und Community Engagement in der IKT-Branche zu verstehen
 - zentrale Plattformen und Foren für berufliche Interaktion und Karriereentwicklung zu identifizieren
 - praktische Networking-Strategien zu entwickeln, um die berufliche Sichtbarkeit und Zusammenarbeit zu fördern
 - die Vorteile der Beteiligung an Open-Source-Projekten, Mentoring-Programmen und Branchendiskussionen zu erkennen
 - best practices kennenzulernen, um ein inklusives und innovatives berufliches Umfeld zu schaffen
-

Benötigte Ausrüstung und Materialien

- Computer mit Internetzugang
- Projektor oder Smartboard
- Digitale oder gedruckte Materialien zu Networking-Strategien und beruflichem Engagement
- Zugang zu professionellen Plattformen (GitHub, LinkedIn, Stack Overflow, Branchenforen)



Aufgabe/Anweisungen

1. **Einleitende Diskussion:** Führe eine Diskussion über die Rolle des Networkings beim beruflichen Aufstieg und Einfluss innerhalb der Branche.
2. **Fallstudienanalyse:** Präsentiere Beispiele von Fachleuten, die Networking für ihre Karriereentwicklung und Projektkooperationen genutzt haben.
3. **Online-Recherche-Aktivität:** Die Schüler:innen erkunden berufliche Networking-Plattformen und identifizieren relevante Communities für ihr Fachgebiet.
4. **Simuliertes Networking-Event:** Organisiere eine Rollenspielübung, in der die Schüler:innen den Austausch mit Kolleg:innen und Branchenexpert:innen üben.
5. **Workshop zum Community Engagement:** Diskutiere die Vorteile der aktiven Teilnahme an Open-Source-Projekten, Mentoring-Programmen und Branchenevents.
6. **Strategie-Session für Social Media:** Unterstütze die Schüler:innen beim Optimieren ihrer beruflichen Profile und beim gezielten Austausch mit branchenspezifischen Inhalten.
7. **Praxisprojekt:** Die Schüler:innen erstellen einen Networking-Aktionsplan mit wichtigen Kontakten, Plattformen und Engagement-Strategien.
8. **Abschlussdiskussion:** Gruppenreflexion über die Bedeutung von Networking und Community-Engagement für das berufliche Wachstum.

Lösung

- **Ergebnis der Recherche:** Die Schüler:innen erstellen eine Liste mit mindestens drei Online-Communities oder Networking-Plattformen, die zu ihren Karriereinteressen passen.
- **Networking-Rollenspiel:** Die Schüler:innen üben und demonstrieren effektive Networking-Fähigkeiten.
- **Social-Media-Optimierung:** Jede:r Schüler:in verbessert sein/ihr LinkedIn- oder GitHub-Profil, um technische Fähigkeiten und berufliche Erfolge zu präsentieren.
- **Networking-Aktionsplan:** Die Schüler:innen stellen ihre Strategie zum Ausbau und zur Pflege ihres professionellen Netzwerks vor – inklusive Best Practices für sinnvolles Engagement.
- **Diskussion und Reflexion:** Die Schüler:innen teilen ihre Erkenntnisse darüber, wie Networking ihre Sicht auf Karriereentwicklung beeinflusst hat.



Kapitel [4.4 – Kontinuierliches Wachstum und Anpassung: Relevant bleiben in der Tech-Branche](#)

Abstract

Dieses Kapitel behandelt die Bedeutung von kontinuierlichem Wachstum und Anpassung für Fachkräfte in der Technologiebranche. Es betont die Notwendigkeit des lebenslangen Lernens, der regelmäßigen Auseinandersetzung mit neuen Technologien und der Fähigkeit, sich an Veränderungen anzupassen. Das Kapitel bietet außerdem Strategien zur kontinuierlichen Verbesserung – wie die Teilnahme an Workshops, Online-Kursen und die Einbindung in Tech-Communities. Ziel ist es, Fachkräfte darin zu unterstützen, in einer sich schnell entwickelnden Branche relevant und wettbewerbsfähig zu bleiben.

Schlüsselwörter

- Kontinuierliches Wachstum
 - Anpassung
 - Lebenslanges Lernen
 - Neue Technologien
 - Technologiebranche
 - Kontinuierliche Verbesserung
 - Workshops
 - Online-Kurse
 - Tech-Communities
-

Dauer

Geschätzte Zeit: 60 Minuten

Lernziele

Am Ende dieser Aktivität werden die Schüler:innen in der Lage sein:

- die Bedeutung von kontinuierlichem Wachstum und Anpassung in der Tech-Branche zu verstehen
 - Strategien für lebenslanges Lernen und das Auf-dem-Laufenden-Bleiben mit neuen Technologien zu benennen
 - die Vorteile der Teilnahme an Workshops und Online-Kursen zu erkennen
 - sich in Tech-Communities zu engagieren, um Wissen zu erweitern und Networking-Möglichkeiten zu nutzen
 - einen persönlichen Plan für kontinuierliche Verbesserung und Anpassung im Berufsleben zu entwickeln
-



Benötigte Ausrüstung und Materialien

- Computer mit Internetzugang
- Projektor oder Smartboard

Aufgabe/Anweisungen

1. **Einführung:** Beginne mit einem Überblick über die rasante Entwicklung der Technologiebranche und die Notwendigkeit kontinuierlichen Wachstums.
2. **Lebenslanges Lernen:** Diskutiere das Konzept des lebenslangen Lernens und warum es für Tech-Fachkräfte essenziell ist.
3. **Auf dem Laufenden bleiben:** Erkläre, wie man sich über neue Technologien mithilfe verschiedener Ressourcen informiert.
4. **Workshops und Online-Kurse:** Hebe hervor, wie wichtig Workshops und Online-Kurse für die Weiterentwicklung von Fähigkeiten sind.
5. **Tech-Communities:** Betone die Rolle von Tech-Communities beim Wissensaustausch und Networking.
6. **Interaktive Diskussion:** Ermutige die Schüler:innen, ihre Erfahrungen mit kontinuierlichem Lernen und Anpassung zu teilen.
7. **Recherche-Aktivität:** Gib den Schüler:innen die Aufgabe, anstehende Workshops, Online-Kurse und Tech-Communities zu recherchieren, denen sie beitreten können.
8. **Planentwicklung:** Die Schüler:innen entwickeln einen persönlichen Plan für kontinuierliches Wachstum und Anpassung mit konkreten Maßnahmen und Zeitrahmen.
9. **Q&A-Session:** Möglichkeit für Fragen und Diskussion der persönlichen Verbesserungspläne.
10. **Abschluss:** Fasse die wichtigsten Punkte zusammen und motiviere die Schüler:innen, sich dem lebenslangen Lernen und der Anpassung zu verpflichten.

Lösung

Für die Recherche-Aktivität sollen die Schüler:innen eine Liste mit Workshops, Online-Kursen und Tech-Communities erstellen, denen sie beitreten möchten. Zusätzlich entwickeln sie einen persönlichen Plan für kontinuierliches Wachstum mit konkreten Maßnahmen und Zeitangaben. In der Q&A-Session stellen sie ihre Pläne vor und erhalten Feedback, um ihre Strategien zu verfeinern und in der Tech-Branche relevant zu bleiben.