Automatische Überprüfung der Anwendung mathematischer Näherungsverfahren durch Coderunner

-

18. Workshop Mathematik in ingenieurwissenschaftlichen Studiengängen

Alexander Dominicus

10.11.2023

Table of Contents

Studierende sollen den Umgang mit iterativen (mathematische) Verfahren erlernen

Iterative Verfahren...

- näherungsweise Lösen einer Gleichung oder eines Gleichungssystems
- näherungsweise Darstellung von Zahlen
- numerische Lösen von (gewöhnlichen) Differentialgleichungen
- Approximation von Funktionen durch Polynome
- usw.

Studierende sollen den Umgang mit iterativen (mathematische) Verfahren erlernen

Iterative Verfahren...

- näherungsweise Lösen einer Gleichung oder eines Gleichungssystems
- näherungsweise Darstellung von Zahlen
- numerische Lösen von (gewöhnlichen) Differentialgleichungen
- Approximation von Funktionen durch Polynome
- usw.

...werden in verschiedenen Bereichen der Hochschullehre angewendet

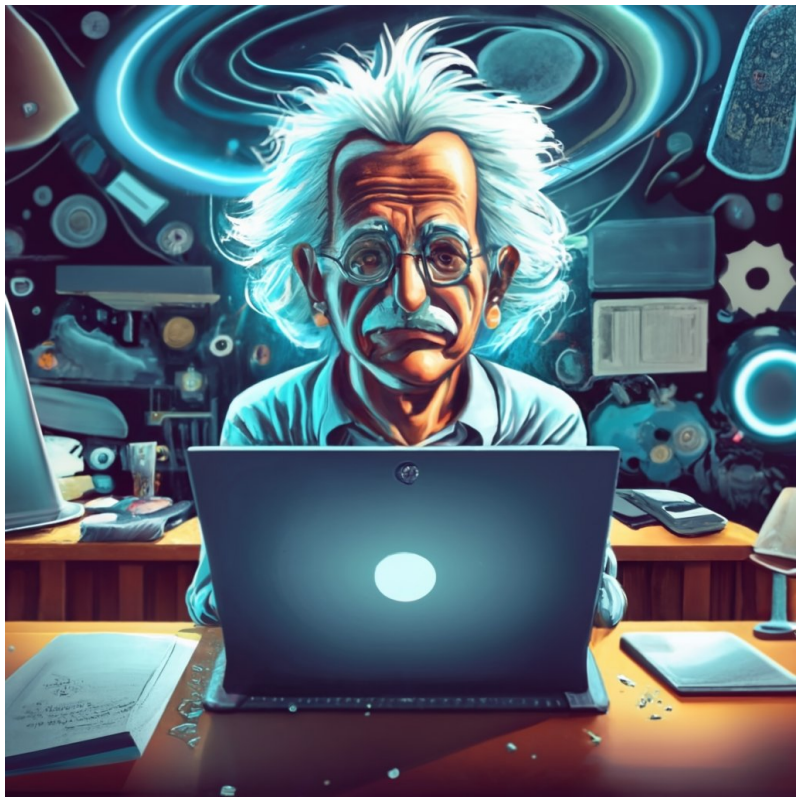
- Mathematik
- Ingenieurswissenschaften
- Informatik
- usw.

Musteraufgabe

Erstellen Sie einen Python-Programmcode, der mit Hilfe des Heron/Newton-Verfahrens für beliebiges $a \in \mathbb{N} > 0$ eine Approximation `my_sqrt(a)` berechnet, sodass

$$|\texttt{my\_sqrt(a)} - \sqrt{a}| < 10^{-6}$$

gilt. Der Code sollte effizient sein d.h. so wenig Iterationen wie möglich verwenden.



verschiedene Problemlösungen

- **klassischer Ansatz:** Studierende erstellen Programmcode, Ausführung und Evaluation durch Dozierende/Mitarbeitende
  - zeitaufwändig
  - subjektiv in der Bewertung
- **automatisierter/modernen Ansatz:** Studierende erstellen Ihren Programmcode innerhalb eines Moodle Test und bekommen direkt eine objektive Bewertung Ihrer Leistung (inkl. Feedback)
- **Wie ist das möglich?**

verschiedene Problemlösungen

- **klassischer Ansatz:** Studierende erstellen Programmcode, Ausführung und Evaluation durch Dozierende/Mitarbeitende
  - zeitaufwändig
  - subjektiv in der Bewertung
- **automatisierter/modernen Ansatz:** Studierende erstellen Ihren Programmcode innerhalb eines Moodle Test und bekommen direkt eine objektive Bewertung Ihrer Leistung (inkl. Feedback)
- **Wie ist das möglich?**

CoderRunner-Plugin (Fragetyp):

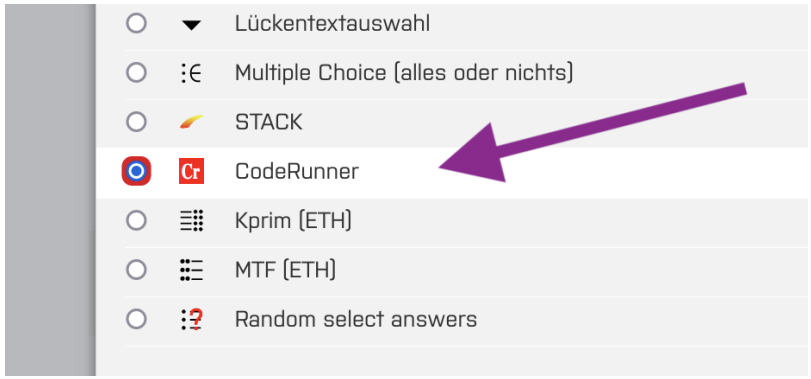


Table of Contents

Was ist Coderunner?

- **Moodle Fragetyp**
- **Studierende erstellen Programmcode**
  - Code wird auf Fehler ausgeführt und auf Fehler geprüft (Jobe-Server)
  - Ausgabe des Codes wird automatisch bewertet
- Unterschiedliche Programmiersprachen unterstützt (Python3, C, C++, Java, PHP, JavaScript (NodeJS), SQL, Octave/Matlab)

Beispiel I: Octave

Schreiben Sie eine Octave-Funktion *quadriere(x)*, welche die Quadratzahl von $x \in \mathbb{R}$ berechnet.

Zum Beispiel:

Test	Resultat
`disp(quadriere(7))`	`49`
`disp(quadriere(-11))`	`121`

Antwort: (Abzugssystem: 33.3, 66.7, ... %)

```
1 function sq=quadriere(x)
2     sq = x * x;
3 end
```

Prüfen

	Test	Erwartet	Erhalten	
✔	`disp(quadriere(7))`	`49`	`49`	✔
✔	`disp(quadriere(-11))`	`121`	`121`	✔
✔	`disp(quadriere(0))`	`0`	`0`	✔
✔	`disp(quadriere(-101))`	`10201`	`10201`	✔
✔	`disp(quadriere(20))`	`400`	`400`	✔

Alle Tests bestanden! ✔

Abbildung: Beispiel: Octave-Programmcode

Beispiel II: Python

Schreiben Sie ein Python-Programm `print_sqrs_1_to_n(n)`, welches tabellarisch die Quadrate einer Zahl $n \in \mathbb{N}$ auflistet.

Zum Beispiel:

Test	Resultat

`print_sqrs_1_to_n(5)`	`1 * 1 = 1`
	`2 * 2 = 4`
	`3 * 3 = 9`
	`4 * 4 = 16`
	`5 * 5 = 25`

Antwort: (Abzugssystem: 10, 20, ... %)

```
1 def print_sqrs_1_to_n(n):
2     for i in range(1,n):
3         quad=i*i
4         print(str(i)+" * "+str(i)+" = " +str(i*i))
5     print(str(n)+" * "+str(n)+" = "+str(n*n))
```

Prüfen

	Test	Erwartet	Erhalten	
✔	`print_sqrs_1_to_n(2)`	`1 * 1 = 1` `2 * 2 = 4`	`1 * 1 = 1` `2 * 2 = 4`	✔

✔	`print_sqrs_1_to_n(5)`	`1 * 1 = 1`
		`2 * 2 = 4`
		`3 * 3 = 9`
		`4 * 4 = 16`
		`5 * 5 = 25`
			`1 * 1 = 1`
			`2 * 2 = 4`
			`3 * 3 = 9`
			`4 * 4 = 16`
			`5 * 5 = 25`
				✔

✔	`print_sqrs_1_to_n(11)`	`1 * 1 = 1`
		`2 * 2 = 4`
		`3 * 3 = 9`
		`4 * 4 = 16`
		`5 * 5 = 25`
		`6 * 6 = 36`
		`7 * 7 = 49`
		`8 * 8 = 64`
		`9 * 9 = 81`
		`10 * 10 = 100`
		`11 * 11 = 121`
			`1 * 1 = 1`
			`2 * 2 = 4`
			`3 * 3 = 9`
			`4 * 4 = 16`
			`5 * 5 = 25`
			`6 * 6 = 36`
			`7 * 7 = 49`
			`8 * 8 = 64`
			`9 * 9 = 81`
			`10 * 10 = 100`
			`11 * 11 = 121`
				✔

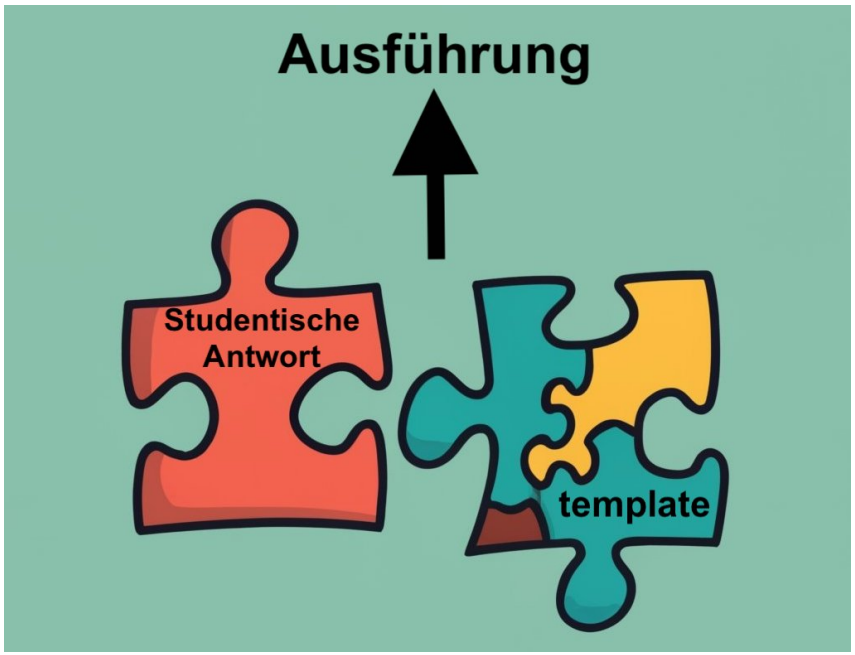
✔	`print_sqrs_1_to_n(0)`	`0 * 0 = 0`	`0 * 0 = 0`	✔

Alle Tests bestanden! ✔

Abbildung: Beispiel: Python-Programmcode

Codeausführung und Benotung: *Exakte Übereinstimmung*

1. Der studentische Programmcode wird in einen *template* eingebettet und damit ausführbar.
2. Der erweiterte Programmcode wird ausgeführt und bewertet:
   - Grundlage der Bewertung bei den gegebenen Beispielen ist die **exakte Übereinstimmung** der Studentischen Antwort mit der hinterlegten Antwort
   - Nur bei exakter Übereinstimmung wird die Frage als richtig gewertet, sonst als falsch



Bewertung: *Exakte Übereinstimmung*

	Test	Erwartet	Erhalten	
✔	print_sqrs_1_to_n(2)	1 * 1 = 1 2 * 2 = 4	1 * 1 = 1 2 * 2 = 4	✔
✔	print_sqrs_1_to_n(5)	1 * 1 = 1 2 * 2 = 4 3 * 3 = 9 4 * 4 = 16 5 * 5 = 25	1 * 1 = 1 2 * 2 = 4 3 * 3 = 9 4 * 4 = 16 5 * 5 = 25	✔
✔	print_sqrs_1_to_n(11)	1 * 1 = 1 2 * 2 = 4 3 * 3 = 9 4 * 4 = 16 5 * 5 = 25 6 * 6 = 36 7 * 7 = 49 8 * 8 = 64 9 * 9 = 81 10 * 10 = 100 11 * 11 = 121	1 * 1 = 1 2 * 2 = 4 3 * 3 = 9 4 * 4 = 16 5 * 5 = 25 6 * 6 = 36 7 * 7 = 49 8 * 8 = 64 9 * 9 = 81 10 * 10 = 100 11 * 11 = 121	✔
✔	print_sqrs_1_to_n(0)	0 * 0 = 0	0 * 0 = 0	✔

Alle Tests bestanden! ✔

	Test	Erwartet	Erhalten	
✖	print_sqrs_1_to_n(2)	1 * 1 = 1 2 * 2 = 4	1 * 1 = 1 2 * 2 = 4 3 * 3 = 9 4 * 4 = 16 5 * 5 = 25	✖
✔	print_sqrs_1_to_n(5)	1 * 1 = 1 2 * 2 = 4 3 * 3 = 9 4 * 4 = 16 5 * 5 = 25	1 * 1 = 1 2 * 2 = 4 3 * 3 = 9 4 * 4 = 16 5 * 5 = 25	✔
✖	print_sqrs_1_to_n(11)	1 * 1 = 1 2 * 2 = 4 3 * 3 = 9 4 * 4 = 16 5 * 5 = 25 6 * 6 = 36 7 * 7 = 49 8 * 8 = 64 9 * 9 = 81 10 * 10 = 100 11 * 11 = 121	1 * 1 = 1 2 * 2 = 4 3 * 3 = 9 4 * 4 = 16 5 * 5 = 25	✖
✖	print_sqrs_1_to_n(0)	0 * 0 = 0	1 * 1 = 1 2 * 2 = 4 3 * 3 = 9 4 * 4 = 16 5 * 5 = 25	✖

Ihr Code muss alle Tests bestehen, um eine Bewertung zu erhalten. Versuchen Sie es noch einmal.

Unterschiede anzeigen

Abbildung: Vergleich: korrekt und falsch gewertete Antworten

Zurück zur Musteraufgabe...

Musteraufgabe

Erstellen Sie einen Python-Programmcode, der mit Hilfe des Heron/Newton-Verfahrens für beliebiges $a \in \mathbb{N} > 0$ eine Approximation `my_sqrt(a)` berechnet, sodass

$$|\texttt{my\_sqrt(a)} - \sqrt{a}| < 10^{-6}$$

gilt. Der Code sollte effizient sein d.h. so wenig Iterationen wie möglich verwenden.

Musteraufgabe $\leftrightarrow$ exakte Übereinstimmung ↯↯

Zurück zur Musteraufgabe...

Musteraufgabe

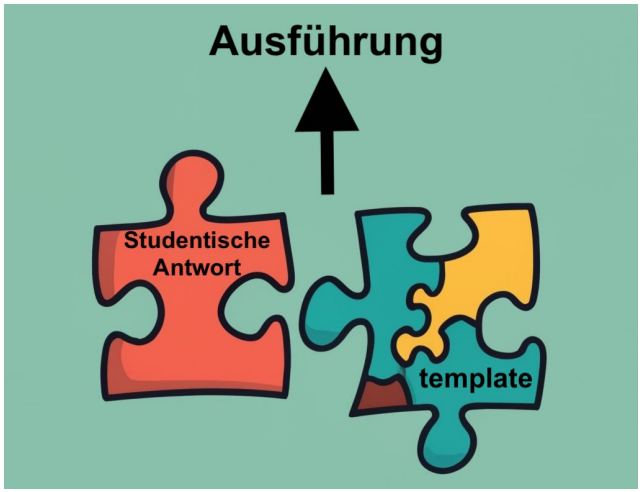
Erstellen Sie einen Python-Programmcode, der mit Hilfe des Heron/Newton-Verfahrens für beliebiges $a \in \mathbb{N} > 0$ eine Approximation `my_sqrt(a)` berechnet, sodass

$$|\texttt{my\_sqrt(a)} - \sqrt{a}| < 10^{-6}$$

gilt. Der Code sollte effizient sein d.h. so wenig Iterationen wie möglich verwenden.

Musteraufgabe $\leftrightarrow$ exakte Übereinstimmung ↯↯

Lösung:



Zurück zur Musteraufgabe...

Musteraufgabe

Erstellen Sie einen Python-Programmcode, der mit Hilfe des Heron/Newton-Verfahrens für beliebiges $a \in \mathbb{N} > 0$ eine Approximation `my_sqrt(a)` berechnet, sodass

$$|\texttt{my\_sqrt(a)} - \sqrt{a}| < 10^{-6}$$

gilt. Der Code sollte effizient sein d.h. so wenig Iterationen wie möglich verwenden.

Musteraufgabe $\leftrightarrow$ exakte Übereinstimmung ↯↯

Lösung:

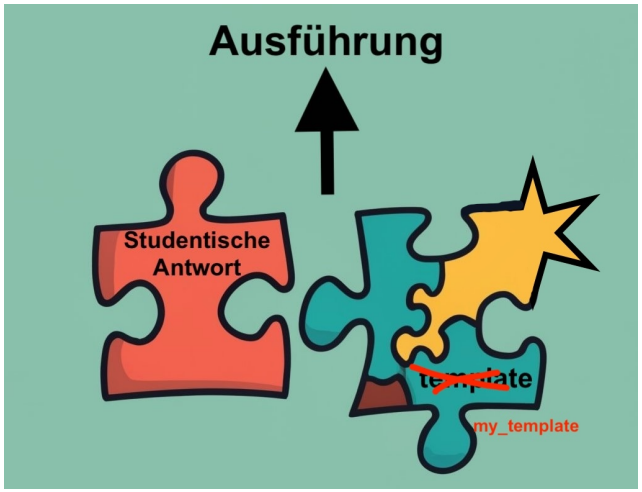


Table of Contents

Anpassung der Benotung

- Coderunner bietet die Möglichkeit das template anzupassen in den der studentische Programmcode eingebettet wird
  - **Bewertung/Benotung** anpassen
  - spezifisches **Feedback**
  - Antworten werten, die keine exakte Übereinstimmung benötigen
- **Musteraufgabe kann in Moodle abgeprüft werden!**

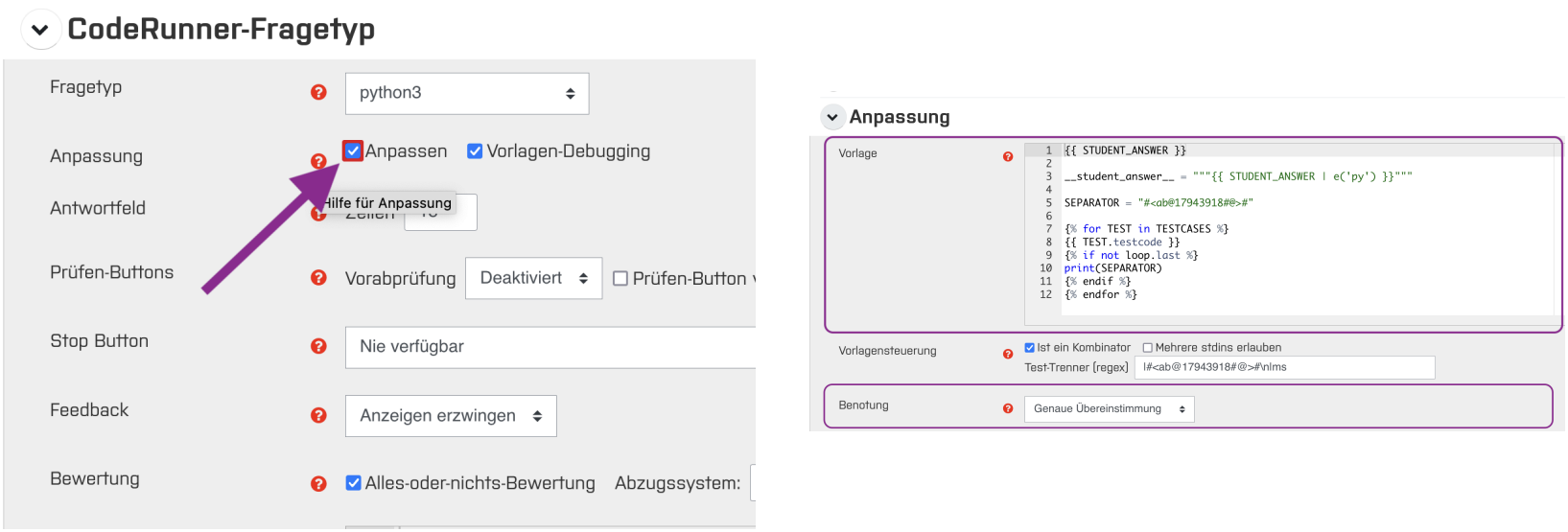
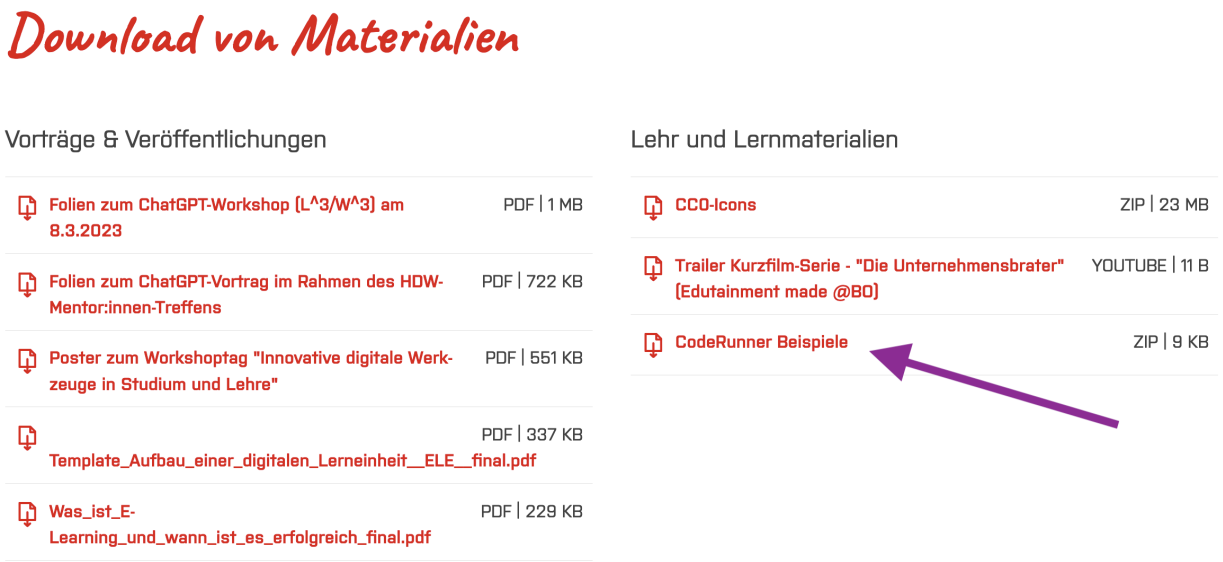


Abbildung: Bewertung Anpassen bei Coderunner Fragetyp


LIVE
DEMO!

Download der Beispiele

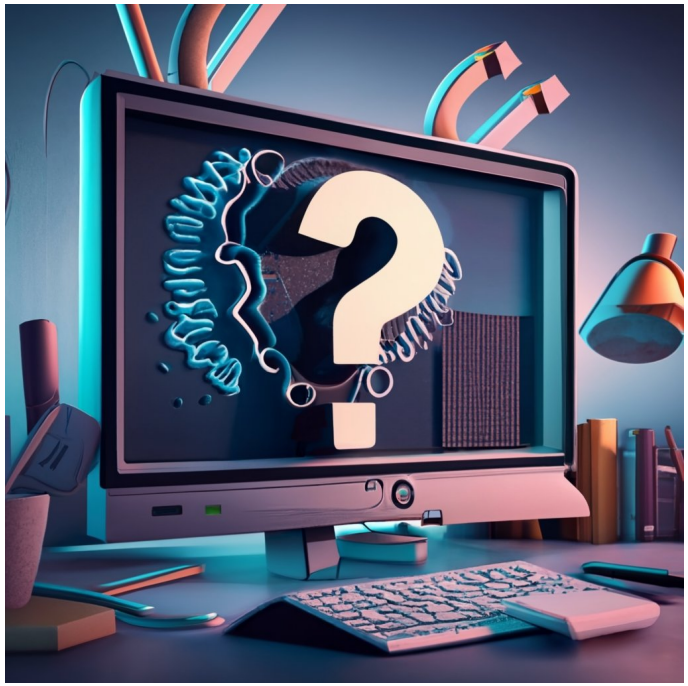
Alle hier gezeigten Beispiele finden Sie zum Download auf der Homepage des DigiTeach-Instituts
https://www.hochschule-bochum.de/digiteach/materialien-downloads/



Ende: Zeit für Diskussionen!

Vielen Dank für Ihre Aufmerksamkeit!

Gibt es Fragen?



Email: alexander.dominicus@hs-bochum.de