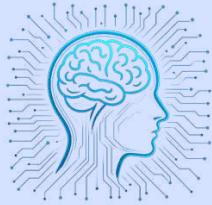


12 – Funktionen mit Rückgabewert

Übersicht



Was lerne ich hier?

- Was sind Funktionen mit Rückgabewert?
- Wie schreiben und verwenden wir solche Funktionen?
- Welche Vorteile hat ihre Verwendung?

Was du können musst:

- Du kannst Funktionen schreiben, die einen Wert zurückgeben.
- Du kannst solche Funktionen korrekt aufrufen.
- Du kannst erklären, welche Vorteile die Verwendung von Funktionen mit Rückgabewert hat.

Du musst nach der Bearbeitung die Aufgaben unten selbstständig lösen können!

Informiere dich mit einem der folgenden Lernwege. Du kannst die Lernwege auch wechseln oder kombinieren. Wichtig ist, dass du die oben genannten Themen selbstständig anwenden kannst – **siehe Aufgabenstellungen ganz unten im Dokument!**

Erklärvideo	Informationen	Lehrer	KI
https://tinyurl.com/win13php - Kapitel 12 – Rückgabewerte	siehe Infoblatt: 12-functions-return-2-info.pdf	Frag deinen Lehrer	Frag die KI – Hinweise siehe unten!

Frag die KI

Die KI (z.B. duck.ai) kann dir wie immer all das erklären. Ziel ist jedoch, dass du dir die obigen Themen so erklären lässt, dass du in der Lage bist, Funktionen mit Rückgabewert korrekt anzuwenden.

Aufgaben

Dein Job für Thema 12 (Rückgabewerte)

Du hast das Thema dann vollständig bearbeitet, wenn du die Aufgaben selbstständig lösen kannst!

1. Siehe Aufgabe auf dem Infoblatt (Tageslohn berechnen)

2. Bremsweg

Wir wollen den Bremsweg¹ und den Reaktionsweg² eines Autos berechnen. Die Formeln lauten:

Bremsweg = (Geschwindigkeit / 10) ^2

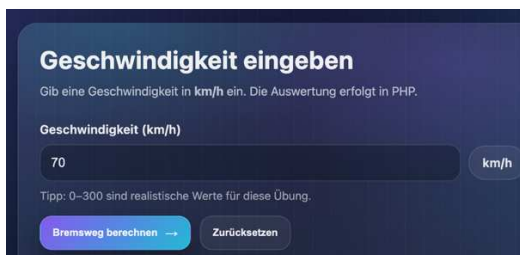
Reaktionsweg = Geschwindigkeit / 10 * 3

Hinweise für die folgenden Aufgaben:

- Potenzen berechnest du in PHP mit dem Potenzierungsoperator **:
`$zweiHochsieben = 2 ** 7;`
- Die Wurzel bestimmst du in PHP mit `sqrt(...)` (steht für *squareroot*):
`$wurzelAus16 = sqrt(16);`

2a. Schreibe eine Funktion **bremswegBerechnen (...)**, die den Bremsweg zurückgibt.

Verwende diese Funktion für die Ausgabe. In der schon vorgefertigten Ergebnisdatei *2a-bremsweg-auswertung.php* findest du weitere Hinweise in den Kommentaren.



2b. Jetzt wollen wir auch noch den Reaktionsweg berechnen (Formel siehe oben). Schreibe eine neue Funktion mit Rückgabewert – Name der Funktion: **reaktionsweg (...)** Lass dir auf der Ergebnisseite die Geschwindigkeit, den Bremsweg und den Reaktionsweg ausgeben.



¹ Strecke, die ein Fahrzeug vom Beginn des Bremsens bis zum Stillstand zurücklegt.

² Strecke, die ein Fahrzeug während der Reaktionszeit des Fahrers bis zum Beginn des Bremsens zurücklegt.

2c Der gesamte Anhalteweg berechnet sich so:

$$\text{Anhalteweg} = \text{Reaktionsweg} + \text{Bremsweg}$$

Schreibe eine Funktion **anhalteweg (...)**, die den Anhalteweg ermittelt und zurückgibt. In der Berechnung des Anhaltewegs sollen die beiden in 2a und 2b geschriebenen Funktionen verwendet werden!

Ergänze die Datei aus Aufgabe 2b entsprechend.

Ergebnis

Geschwindigkeit: 80 km/h

Bremsweg: 64 m

Reaktionsweg: 24 m

Anhalteweg: 88 m

2d Bei einer **Vollbremsung** halbiert sich der Bremsweg. Ein verändertes HTML-Formular wurde bereits erstellt (*2d-bremsweg-vollbremsung.html*):

Bremsweg berechnen (optional: Vollbremsung)

Geschwindigkeit eingeben und optional „Vollbremsung“ aktivieren.

Geschwindigkeit (km/h)

80

☒ Vollbremsung

Berechnen Zurücksetzen

Je nach Eingabe (Vollbremsung ja/nein) soll der entsprechende Anhalteweg ausgegeben werden:

Ergebnis

Geschwindigkeit: **80** km/h

Bremsart: **Vollbremsung**

Bremsweg: **32,00** m

Ergebnis

Geschwindigkeit: **80** km/h

Bremsart: **normales Bremsen**

Bremsweg: **64,00** m

Die Checkbox wertest Du so aus:

```
if (isset($_POST['cbVollbremsung'])) {  
    // Checkbox ist angeklickt  
} else {  
    // Checkbox ist nicht angeklickt  
}
```

Code

3. Notenberechnung

Du sollst ein PHP-Programm schreiben, das aus einer erreichten Punktzahl eine Schulnote berechnet.

Im Formular werden zwei Werte eingegeben: die **erreichte Punktzahl** und die **maximale Punktzahl** der Prüfung.

Note berechnen

Erreichte Punktzahl:

Maximale Punktzahl:

Die Note wird mit folgender Formel berechnet:

$$\text{Note} = 6 - 5 \cdot \text{erreichtePunktzahl} / \text{maximalPunktzahl}$$

Schreibe eine Funktion, die aus der erreichten Punktzahl und der maximalen Punktzahl die Note berechnet und **als Rückgabewert** zurückgibt. Verwende den Rückgabewert der Funktion, um die Note auf der Auswertungsseite auszugeben.

Ergebnis

Erreichte Punktzahl: 100

Maximale Punktzahl: 111

Note: 1,50

[Nochmal](#)

Wenn die berechnete Note **besser als 1,2** ist, soll zusätzlich ein Pokal-Emoji 🏆 ausgegeben werden.

Ergebnis

Erreichte Punktzahl: 111

Maximale Punktzahl: 111

Note: 1,00 🏆

[Nochmal](#)

Übungsaufgaben

Mit diesen Aufgaben kannst du das Thema weiter üben und vertiefen.

Sehr guter Tipp vorweg

Du wirst in diesen Aufgaben dauernd Zahlen mit `number_format(...)` formatieren müssen. Das nervt natürlich ungemein.

Warum schreibst du dir nicht eine Funktion, die die Zahl als Parameter bekommt und die formatierte Zahl zurückgibt? Also etwa

```
function fmt($x)
{
    return number_format((float)$x, 0, ",", ".");
}
```

Code

4 Nerds:

`(float)` wandelt `$x` um in eine Kommazahl (Casting). Das machen wir deshalb:

- Typ-Sicherheit: Falls `$x` ein String ist (bspw. aus einem Eingabefeld), würde `number_format(...)` einen Fehler werfen. Dem beugen wir damit vor.
- Code-Klarheit: Jeder, der das `(float)` sieht, weiß: `$x` soll eine Zahl sein.

Du kannst das `(float)` aber auch weglassen, es wird vermutlich funktionieren.

Diese Funktion kannst du jedesmal verwenden, wenn du eine Zahl ausgeben willst:

```
echo "<p>Kontostand: " . fmt($kontostand) . " Dollar.</p>";
```

Code

- 4a. Du willst für einen Online-Shop eine einfache Bestellabrechnung programmieren. Es soll die Rechnungssumme abhängig von Stückzahl und Preis eines Artikels berechnet werden.

Der Kunde gibt in einem Formular die Stückzahl eines Artikels sowie den Artikelpreis ein:

Rechnungssumme berechnen

Gib Artikelanzahl und Artikelpreis ein. Danach kannst du wahlweise die Auswertung 4a oder 4b starten.

Artikelanzahl	Artikelpreis (€)
8	2,50
Ganzzahl ≥ 1	Zahl ≥ 0

Schreibe eine PHP-Funktion **rechnungssumme (...)**, die aus diesen beiden Werten die Rechnungssumme berechnet (Stückzahl multipliziert mit Artikelpreis). Beachte, dass der Funktion ZWEI Parameter übergeben werden (Stückzahl und Artikelpreis).

Die Funktion soll das Ergebnis zurückgeben. Verwende den Rückgabewert der Funktion, um die Rechnungssumme auf der Auswertungsseite anzuzeigen.

Auswertung 4a	
Rechnungssumme mit einer Funktion mit Rückgabewert.	
Artikellanzahl:	8
Artikelpreis:	2,50 €
Rechnungssumme:	20,00 €

4b. Versandkostenaufschlag

Der Online-Shop möchte nun zusätzlich Versandkosten berücksichtigen.

Schreibe dafür weitere Funktion **rechnungsbetragInklusiveVersandkosten (...)**. In dieser Funktion wird geprüft, ob die Rechnungssumme mindestens 100 Euro beträgt. Verwenden Sie zur Ermittlung der Rechnungssumme die Funktion **rechnungssumme (...)** aus Aufgabe 3a (copy-paste ☺).

Ab 100 Euro ist der Versand kostenfrei. Liegt der Rechnungsbetrag darunter, werden 4,99 Euro Versandkosten berechnet.

Die Funktion soll den gesamten Rechnungsbetrag inklusive Versandkosten zurückgeben. Verwenden Sie auch hier ausschließlich den Rückgabewert der Funktion für die Ausgabe auf der Webseite.

Hinweis:

*Die Schwierigkeit dieser Aufgabe liegt darin, dass Sie genau überlegen müssen, an welcher Stelle die Funktion **rechnungssumme (...)** zum Einsatz kommt.*

Auswertung 4b	
Gesamtbetrag mit Versandkostenregel (ab 100 € frei).	
Artikellanzahl:	10
Artikelpreis:	2,50 €
Rechnungssumme:	25,00 €
Gesamtbetrag inkl. Versand:	29,99 €
Versandkosten:	4,99 €

Auswertung 4b	
Gesamtbetrag mit Versandkostenregel (ab 100 € frei).	
Artikellanzahl:	1000
Artikelpreis:	2,50 €
Rechnungssumme:	2.500,00 €
Gesamtbetrag inkl. Versand:	2.500,00 €
Versandkosten:	keine (versandkostenfrei)

In der untersten Zeile willst du die Versandkosten separat anzeigen (4,99€ oder »keine (versandkostenfrei)«). Du hast verschiedene Möglichkeiten, das zu bewerkstelligen.

So richtig pro-mäßig wäre es, wenn Du eine Funktion schreibst, die erst mal prüft, ob es Versandkosten gibt (true/false).

5a. Prüfungsauswertung

Du sollst für eine Prüfungsauswertung ein kleines PHP-Programm schreiben.

Im Formular gibt der Prüfling an, wie viele Punkte er in der Prüfung erreicht hat (maximal 100). Außerdem wird eingegeben, ab wie viel Prozent die Prüfung als bestanden gilt.

The screenshot shows a web form with a dark blue background. The title is 'Bestanden oder nicht?'. Below it, a paragraph explains the task: 'Gib die erreichten Punkte (max. 100) und die Bestehensgrenze in Prozent an. Die Auswertung erfolgt mit einer Funktion, die true oder false zurückgibt.' There are two input fields: 'Erreichte Punkte (0-100)' with the value '99' and 'Bestehensgrenze in %' with the value '50'. Below these are two buttons: 'Auswerten' (highlighted in blue) and 'Zurücksetzen'. A tip text says: 'Tipp: 50% bedeutet: ab 50 von 100 Punkten gilt die Prüfung als bestanden.' At the bottom, it shows file paths: 'Datei: 5a-pruefung.html · Bild: ../img/pruefungsergebnisse.png'. To the right of the form is a photo of a smiling male student in a red plaid shirt holding a certificate, while a male teacher in a suit looks on in a classroom setting.

Schreibe eine PHP-Funktion **pruefungBestanden (...)**. Die Funktion soll prüfen, ob die erreichte Punktzahl für das Bestehen reicht. Wenn die Prüfung bestanden ist, soll die Funktion **true** zurückgeben, andernfalls **false**.

Der Rückgabewert der Funktion soll anschließend in einer if-Verzweigung verwendet werden:

- Gibt die Funktion **true** zurück, soll die Meldung »Bestanden!!« ausgegeben werden.
- Gibt die Funktion **false** zurück, soll die Meldung »Nur x% der möglichen Punktzahl – das sind y% zu wenig. Leider nicht bestanden.« ausgegeben werden.

Hinweis:

Die Funktion gibt einen **booleschen Wert** (**true** oder **false**) zurück. Dieser Rückgabewert wird direkt in einer if-Abfrage geprüft.

The screenshot shows a dark blue box titled 'Ergebnis'. It displays 'Erreichte Punkte: 99 / 100' and 'Bestehensgrenze: 50 %'. At the bottom, a green button-like area contains the text 'Bestanden!!'.

The screenshot shows a dark blue box titled 'Ergebnis'. It displays 'Erreichte Punkte: 30 / 100' and 'Bestehensgrenze: 50 %'. At the bottom, a purple area contains the text 'Nur 30% der möglichen Punktzahl – das sind 20% zu wenig. Leider nicht bestanden.'

5b. Prüfungsauswertung mit Bild

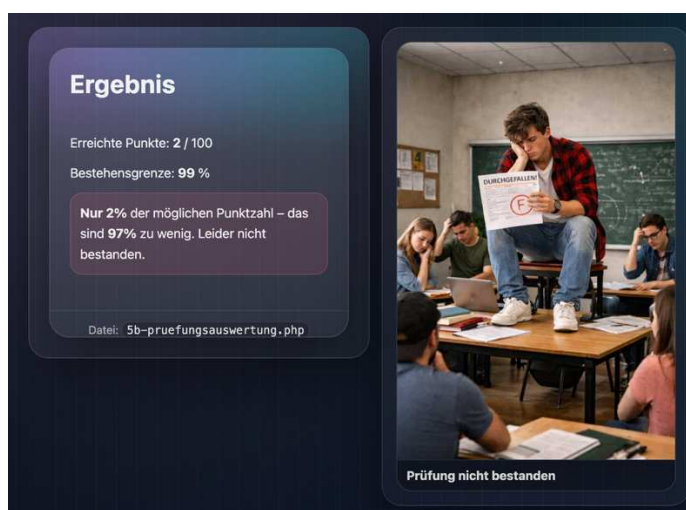
Ergänze 5a wie folgt:

In der Auswertung soll zusätzlich ein Bild angezeigt werden, das vom Ergebnis abhängt:

- Wenn du nicht bestanden hast, soll das Bild `../img/durchgefallen.jpg` angezeigt werden.
- Wenn du bestanden hast, soll das Bild `../img/bestanden.jpg` angezeigt werden.

Dazu nutzt du weiterhin die Funktion `pruefungBestanden(...)`, die `true` oder `false` zurückgibt. Der Rückgabewert wird in einer if-Verzweigung geprüft. In dieser Verzweigung wird nicht nur der Text ausgegeben, sondern auch das passende Bild.

Das Formular bleibt gleich, es liegt dir als `5b-pruefung.html` bereits vor. Verwende für die Ergebnisseite `5b-pruefungsauswertung.php`.



oder eben

