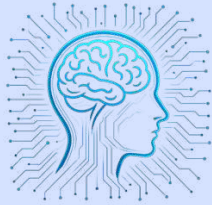


11 – Funktionen mit Übergabeparameter

Übersicht



Was lerne ich hier?

- Welchen Nutzen haben Übergabeparameter in Funktionen?
- Werte korrekt an Funktionen übergeben
- Diese Werte innerhalb der Funktion verwenden

Was du können musst:

- Du kannst Funktionen schreiben, die einen oder mehrere Übergabeparameter erhalten.
- Du kannst solche Funktionen korrekt aufrufen.
- Du kannst innerhalb der Funktion mit den übergebenen Werten arbeiten.
- Du kannst erklären, wozu man Übergabeparameter verwendet.

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/winf13php - Kapitel 11 – Übergabeparameter in PHP	siehe Infoblatt: 11-functions-parameter-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, Übergabeparameter korrekt zu verwenden.

Aufgaben

Dein Job für Thema 11 (Übergabeparameter)

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

1. Siehe Aufgabe auf dem Infoblatt (Passwortprüfung).

2. Ein Fitnessstudio nutzt ein Online-Formular zur Altersprüfung seiner Mitglieder. Der Zutritt zum Trainingsbereich ist erst ab 18 Jahren erlaubt.

Ein Formular zur Eingabe des Alters liegt bereits vor als *2-fitnessstudio.html*. Ebenso ist das Grundgerüst der Auswertungsseite vorhanden als *2-fitnessstudio-auswertung.php*.

Erstelle in der Auswertungsseite eine PHP-Funktion **alterPruefen (\$pAlter)**.

Die Funktion soll prüfen, ob das übergebene Alter mindestens 18 ist, und eine passende Meldung ausgeben:

- Zutritt erlaubt (volljährig)
- Zutritt nicht erlaubt (minderjährig)

3. Eine Paketstation-Terminalseite soll prüfen, ob eine eingegebene Postnummer korrekt formatiert ist. Die Postnummer ist ein Code, der genau 10 Zeichen lang sein muss. Ein Formular zur Eingabe der Postnummer liegt bereits vor als *3-paketstation.html*.

Erstelle in *3-paketstation-auswertung.php* eine PHP-Funktion **postnummerPruefen (\$pCode)**

Die Funktion soll mit **strlen (\$pCode)** prüfen, ob der eingegebene Code genau 10 Zeichen lang ist, und eine passende Meldung ausgeben:

PAKET1 Ergebnis
Postnummer-Check (strlen)

Auswertung
Eingegeben: grunz

Postnummer ungültig – falsche Länge (benötigt: 10 Zeichen).

[← zurück zur Eingabe](#)

Terminalmodus: Retro • Modul: strlen()

PAKET1 Ergebnis
Postnummer-Check (strlen)

Auswertung
Eingegeben: 1234567890

Postnummer gültig – Länge passt (10 Zeichen).

[← zurück zur Eingabe](#)

Terminalmodus: Retro • Modul: strlen()

- Postnummer gültig
- Postnummer ungültig (falsche Länge)

Hinweis:

strlen(\$variable) gibt die Anzahl der Zeichen von **\$variable** zurück. Beispiel:

```
$x = "hanswurst";
echo "<p>$x hat " . strlen($x) . " Zeichen.</p>";
if(strlen($x) > 1000) {
    echo "<p>Wow - $x hat mehr als 1000 Zeichen!</p>";
}
```

Code

4. Du sollst ein Registrierungsmodul simulieren, bei der ein Benutzer ein Profil anlegt. Dabei gelten folgende Regeln:

- Der **Profilname** muss mindestens **3 Zeichen** lang sein.
- Die **Jugendschutz-PIN** muss mindestens **6 Zeichen** lang sein.

Aufgabe

1. Erstelle ein HTML-Formular mit dem Dateinamen *4-register.html*
Das Formular soll zwei Eingabefelder enthalten:
 - Profilname
 - Jugendschutz-PIN
 Das Formular soll die eingegebenen Daten per GET senden.
2. Erstelle die Auswertungsseite *4-register-auswertung.php*
3. Schreibe in der Auswertungsseite zwei PHP-Funktionen, mit denen geprüft wird, ob der Profilname mindestens 3 Zeichen lang ist und ob die Jugendschutz-PIN mindestens 6 Zeichen lang ist.
4. Erstelle ein ordentliches Stylesheet.

Registrierung

Profilname:

Jugendschutz-PIN:

Auswertung

Profilname: **fritz**
PIN: **123**
Profilname gültig
Jugendschutz-PIN ungültig (mindestens 6 Zeichen)
[Zurück](#)

Übungsaufgaben

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

- | | |
|------------|---|
| 3b. | Erweitere 3a folgendermaßen: Es soll ausgegeben werden, wie viele Zeichen zu viel oder zu wenig eingegeben wurden, etwa in Form
<i>Postnummer zu kurz: 12345678 – es fehlen 2 Zeichen!</i>
<i>Postnummer zu lang: 1234567890123 – das sind 3 Zeichen zu viel!</i> |
| 4b. | Erweitere 4a so, dass beide Kriterien erfüllt sein müssen: Eine Erfolgsmeldung wird nur dann ausgegeben, wenn der Name mindestens drei Zeichen hat <u>und</u> die PIN mindestens 6 Zeichen hat. Schreibe dazu eine eigene Funktion.

<u>Tipp:</u>
Dieser Funktion musst du natürlich beide Werte geben; du arbeitest also mit ZWEI Übergabeparametern, die du einfach durch Komma trennst. Wenn du das nicht verstehst – das Internet ist dein Freund und erklärt es dir gerne. |
| 5. | ... siehe folgende Seite ... |

5. Ein Lieferdienst verschickt Bestellbestätigungen per E-Mail.
Damit eine E-Mail-Adresse akzeptiert wird, müssen **zwei Bedingungen** erfüllt sein:
- Die E-Mail-Adresse muss ein **@-Zeichen** enthalten.
 - Die E-Mail-Adresse muss einen **Punkt (.)** enthalten.

Aufgabe

1. Schreibe in der Auswertungsseite eine PHP-Funktion `emailPruefen($pMail)`.
Die Funktion soll prüfen, ob **beide Bedingungen erfüllt sind**:
 - @ ist enthalten UND
 - . ist enthaltenNur wenn beide Bedingungen erfüllt sind, gilt die E-Mail-Adresse als gültig.
2. Lies den Formularwert aus, rufe die Funktion auf und gib eine passende Meldung aus: »E-Mail-Adresse gültig« oder »E-Mail-Adresse ungültig«

Hinweis:

Du brauchst dazu die PHP-Funktion `strpos(...)`. Diese Funktion sucht in einem Text nach einem bestimmten Zeichen oder einem Teilstring. Beispiele:

```
strpos("test@mail.de", "@");  
    // liefert eine Zahl (Position des @)  
strpos("test@mail.de", ".");  
    // liefert eine Zahl (Position des Punktes)  
strpos("testmailde", "@");  
    // liefert false (weil kein @ gefunden)
```

Code

Wichtig:

- Liefert `strpos()` **false**, dann wurde das Zeichen **nicht gefunden**.
- Liefert `strpos()` eine Zahl, dann ist das Zeichen **vorhanden**.

SwiftDrop Lieferdienst

Sendung verfolgen Preise Hilfe

App öffnen

Bestellbestätigung per E-Mail

Wir prüfen kurz, ob die Adresse ein @ und einen Punkt enthält – damit die Bestätigung sicher ankommt.

Schnell Sicher Transparent

E-Mail prüfen

Bitte E-Mail-Adresse eingeben.

E-Mail-Adresse

■ friedrich@mail

Jetzt prüfen

Hinweis: Diese Aufgabe prüft nur, ob @ und . enthalten sind.

Warum die Prüfung?
Ohne gültige Adresse können wir keine Bestellbestätigung schicken.

Was wird geprüft?
Ob ein @ und ein Punkt (.) im Text vorkommen – mehr nicht.

Datenschutz
Die Eingabe wird nur für diese Auswertung verwendet.

© SwiftDrop - Demo-Seite für PHP-Übungen Impressum Datenschutz Kontakt

SwiftDrop Lieferdienst

Sendung verfolgen Preise Hilfe

App öffnen

Ergebnis: E-Mail-Check

Die Adresse ist gültig, wenn sie ein @ und einen . enthält.

Auswertung

Prüfung mit strpos()

Eingegeben: friedrich@mail

E-Mail-Adresse ungültig

← zurück zum Formular

© SwiftDrop - Demo-Seite für PHP-Übungen Impressum Datenschutz Kontakt