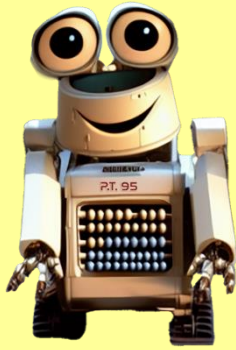


# FUNKTIONEN MIT ÜBERGABEPARAMETER



## PT95 erklärt: Was sind Übergabeparameter?

Stell dir folgendes vor: Funktionen sind wie kleine Helfer, die Aufgaben für dich erledigen. Manchmal brauchen diese Helfer Informationen, um besser zu verstehen, was du von ihnen willst. Hier kommen die Übergabeparameter ins Spiel.

Die Übergabeparameter sind wie Nachrichten, die du den Helfern gibst. Wenn du zum Beispiel möchtest, dass ein Helfer die Summe von zwei Zahlen berechnet, gibst du ihm diese Zahlen als Übergabeparameter.

## Braucht man das wirklich?

Wir schauen uns das mal in einem Beispiel an: Die folgende Funktion soll prüfen, ob das geheime Passwort „Marshmallow“ korrekt eingegeben wurde:

```
<?php
    $passwort = $_POST["txtPasswort"];

    passwortPruefen();

    function passwortPruefen()
    {
        if($passwort == "Marshmallow")
        {
            echo "Passwort korrekt :-)";
        }
    }
?>
```




1. Probieren Sie den obigen Code aus! Erstellen Sie dafür die PHP-Seite *passwort\_pruefen.php* und rufen Sie diese Seite über einen Browser auf. Die HTML-Seite mit dem Formular wurde bereits erstellt: Verwenden Sie hier die Datei *passwort.html*.

Passwort eingeben:



## Was ist denn hier los?

Wenn Sie das ausprobieren, werden Sie feststellen, dass das nicht funktioniert! Sie werden eine Fehlermeldung wie diese erhalten:

**Notice:** Undefined variable:  *passwort*

### Das Problem liegt hier:

*Eine Variable (\$password) ist nur dort sichtbar, wo Sie erstellt wurde. Das heißt: Eine Funktion „sieht“ die Variablen nicht, die im Hauptprogramm erstellt wurden. Ebenso gilt: Das Hauptprogramm sieht die Variablen nicht, die in einer Funktion erstellt werden.*

### Die Lösung: Übergabeparameter

Wenn eine Funktion nicht auf die Variable zugreifen kann, die wir außerhalb erstellt haben, dann müssen wir der Funktion den Wert der Variablen eben „in die Hand drücken“. So etwas nennt man einen Übergabeparameter.

#### Funktion mit Übergabeparameter:

```
function NAME-DER-FUNKTION(ÜBERGABEPARAMETER)
{
    // Hier passiert irgendwas;
}
```

#### Beispiel:

```
<?php
$password = $_POST["txtPasswort"];
passwordPruefen($password);
function passwordPruefen($meinPasswort)
{
    if($meinPasswort == "Marshmallow")
    {
        echo "Passwort korrekt :-)";
    }
    else
    {
        echo "ACCESS DENIED!";
    }
}
?>
```

**Funktionsaufruf mit Parameter**

**Wert von \$password wird an \$meinPasswort übergeben**

### Aufgabe:



1. Probieren Sie den obigen Code aus, indem Sie Ihre php-Seite entsprechend ändern.  
Je nachdem, ob Sie das korrekte Passwort eingegeben haben, müssen Sie dann eine der beiden Meldungen sehen:

Passwort korrekt :-)

ACCESS DENIED!

2. Ihre Aufgabe ist es, in PH eine Registrierung der Benutzer zu programmieren. Hierfür muss man eine Mailadresse angeben und ein Passwort wählen.

Das Formular zur Eingabe der Benutzerdaten (*passwortLaenge.html*) wurde bereits erstellt:

Deine Mailadresse:

Vergib ein Passwort:

**In einem ersten Schritt sollen Sie nach dem Abschicken der Daten mit einer Funktion prüfen, ob das Passwort lang genug ist. Dieses muss mindestens 10 Zeichen haben!**

Gehen Sie dabei so vor:

- Erstellen sie die Seite *passwortLaenge.php*
- Schreiben Sie die **Funktion** *passwortLaengePruefen()*. Dieser Funktion soll das **Passwort als Parameter** übergeben werden.
- Testen Sie in der Funktion, ob die Länge des Passworts ausreichend ist, indem Sie den Befehl *strlen()* verwenden!

Hier ein Beispiel zur Verwendung:

```
if(strlen($variable) > 100)
{
    echo "Die Variable enthält mehr als 100 Zeichen!"
}
```

Je nach Ergebnis der Prüfung soll eine der beiden Meldungen ausgegeben werden:

Länge des Passworts okay! :-)

Passwort zu kurz!



### **Hoppla!**

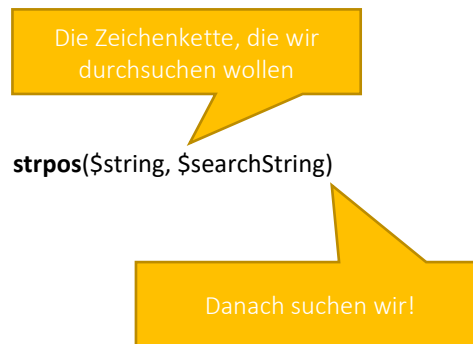
Vielleicht haben Sie es gemerkt: Bei **strlen()** handelt es sich ebenfalls um eine Funktion, die uns PHP zur Verfügung stellt. Der Name der Funktion ist übrigens eine Abkürzung von *String Length* (String: Zeichenkette).

3. Prüfen Sie mit der **Funktion** `mailadressePruefen()`, ob die eingegebene Mailadresse ein @-Zeichen enthält. Verwenden Sie hier für das Formular aus Aufgabe 2.

Deine Mailadresse:

Vergib ein Passwort:

Um eine Zeichenkette (englisch: String) zu durchsuchen, können Sie diesen Befehl verwenden:



Diese Funktion liefert uns den Wert **FALSE** zurück, falls die gesuchte Zeichenkette **nicht enthalten** ist! Andernfalls erhalten wir die Position der gesuchten Zeichenkette.

Beispiel:

```
if(strpos('Müllwagen', 'ü') == false)
{
    echo "KEIN 'ü' enthalten!";
}
else
{
    echo "Gefunden! Buchstabe 'ü' enthalten!";
}
```

Die Ausgabe soll so aussehen:

