

FUNKTIONEN: WIR SCHREIBEN UNTERPROGRAMME



Unsere bisherigen Programme haben zwei entscheidende Nachteile:

- Je mehr wir programmieren, desto **unübersichtlicher** wird das Programm
- Es ist kaum möglich, Programmteile **wiederverwenden**

Für diese Probleme gibt es eine einfache Lösung: Funktionen!

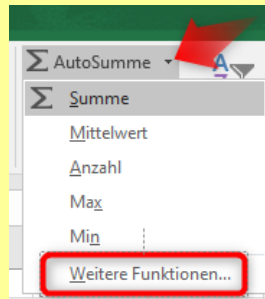
Was ist eine Funktion?

Eine **Funktion** ist ein kleines *Unterprogramm*, das etwas für uns tut. Wir haben auch schon Funktionen verwendet, die uns PHP zur Verfügung stellt (z.B. `number_format()` zur Formatierung von Zahlen).



Das ist nichts Neues!

Funktionen haben Sie schon öfter verwendet, wahrscheinlich ohne zu merken, dass es sich dabei um Funktionen handelt. Zum Beispiel in Excel:



Benutzerdefinierte Funktionen (=User Defined Functions)

Was jetzt neu ist: Wir werden selbst Funktionen erstellen! Diese werden unsere Programme übersichtlicher machen und erleichtern uns die Wiederverwendbarkeit von Teilen des Programms.

So erstellt man eine einfache Funktion:

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

Die Funktion wird im Programm dann über ihren Namen aufgerufen!

Beispiel:

```
<?php
function ausgabe_datum()
{
    echo "Heutiges Datum: ". date("d.m.y");
}

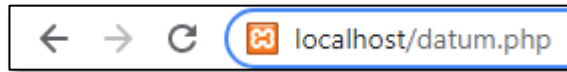
ausgabe_datum();
?>
```

Hier rufen wir die Funktion auf!

Aufgabe:



1. Probieren Sie den obigen Code aus!
Erstellen Sie dafür die PHP-Seite *datum.php* und rufen Sie diese Seite über einen Browser auf.



2. Erstellen Sie eine Funktion *sayHello*, die nach Eingabe eines Namens eine persönliche Begrüßung mit aktueller Uhrzeit realisiert. Das entsprechende Formular (*hallo.html*) wurde bereits entworfen. So soll es aussehen:

Tipp:

Die aktuelle Uhrzeit wird so ermittelt: `date('H:i:s')`

3. Ihre Aufgabe ist es, in PHP eine Registrierung der Benutzer zu programmieren. Wer sich auf unserer Webseite registrieren möchte, muss dafür einen Benutzernamen und ein Passwort wählen. In einem weiteren Schritt sollen diese Daten überprüft werden.



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

In einem ersten Schritt sollen Sie nach dem Abschicken der Daten prüfen, ob tatsächlich ein Passwort eingegeben wurde! Gehen Sie dabei so vor:

- Erstellen sie die Seite *registrierung_pruefen.php*
- Schreiben Sie die Funktion *testenObPasswortLeer()*
- Testen Sie in der Funktion, ob ein Passwort eingegeben wurde, indem Sie den Befehl *empty()* verwenden! Diesen können Sie so einsetzen:

```
if(empty($password)==true)
{
    // Mach irgendwas!
}
```

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

Passwort darf nicht leer sein!

Es wurde ein Passwort eingegeben!



Hoppla!

Vielleicht haben Sie es gemerkt: Bei ***empty()*** handelt es sich ebenfalls um eine Funktion, die uns PHP zur Verfügung stellt. Funktionen haben immer hinten die beiden Klammern! Welchen Zweck diese Klammern haben, werden wir uns später noch genauer anschauen.