Sessions in PHP

https://www.php-einfach.de/php-tutorial/php-sessions/

HTTP ist ein zustandsloses Protokoll, das bedeutet, Informationen werden zwischen den verschiedenen Aufrufen eines Besuchers nicht zwischengespeichert. Dies ist natürlich unpraktisch wenn wir gewisse Informationen zu einem Besucher speichern müssen, beispielsweise mit welchem Benutzernamen sich dieser eingeloggt hat. Um dies zu lösen, verwendet man in PHP Sessions.

Auf dem Server ist dann für jede Session-ID lokal ein Speicher eingerichtet, der beliebige Variablen für den Besucher beinhalten kann. So können wir in der Session für einen Besucher beispielsweise abspeichern, mit welchem Benutzernamen sich dieser eingeloggt hat, welche Waren in seinem Warenkorb liegen usw.

Der Benutzer hat **keine** Möglichkeiten, die Variablen in seiner Session zu sehen oder zu manipulieren. Er besitzt nur die Information, welche Session ID ihm vom Webserver zugewiesen wurde.

Session registrieren

Möchte man in einem Script auf Sessions zurückgreifen, muss man, bevor man irgendeine Ausgabe macht, den Befehl *session_start();* aufrufen:

```php
<?php
session_start();
?>
```

Es empfiehlt sich, diesen Code immer ganz oben der Scripts stehen zu haben.

Möchte man einen Wert / eine Variable über mehrere Seitenaufrufe hinweg in der Session speichern, dann geht dies wie folgt:

```php
<?php
$_SESSION['name'] = "wert";
?>
```

Diesen Wert kann man später, auch auf anderen Seiten, wie folgt ausgeben:

```php
<?php
$name = $_SESSION['name'];
echo $name;
?>
```

Wichtig, immer wenn man irgendwo mit Sessions arbeitet, muss zuvor *session_start()* ausgeführt worden sein. Man kann dabei die Session-Variable wie jede andere Variable in PHP verwenden. Man kann darin Zahlen, Zeichenketten oder sogar Arrays abspeichern.

Mit **isset** kann man überprüfen, ob die Session registriert wurde. Dies lässt sich bei Logins verwenden:

```php
<?php
session_start();


if (isset($_SESSION['username'])) {
   echo "Herzlich Willkommen ".$_SESSION['username'];
} else {
   echo "Bitte erst einloggen";
}
?>
```


```
<?php
session_start();


if (isset($_SESSION['username'])) {
   echo "Herzlich Willkommen ".$_SESSION['username'];
} else {
   echo "Bitte erst einloggen";
}
?>
```


Beispiel mit Login

Dies ist ein ganz simples Beispiel, zeigt aber gleich die Stärke von Sessions.

Bei dem Formular gibt man unseren Namen an, diese Daten werden dann an login.php gesendet. Dort werden sie abgefragt und der *name* aus dem Formular wird in der Session username gespeichert. Wenn man dann auf den Link klickt und auf seite2.php gelangt, so kann man dort weiter den Namen des Besuchers (der Name aus dem Formular) ausgeben. Dies könnte man über beliebig viele Seiten weiterführen, mit unbegrenzt vielen Session-Variablen.

formular.html

```html
<html>
    <head></head>
    <body>
    <form action="login.php" method="post">
        <h2>Dein Name:</h2>
        <input type="text" name="name">
        <input type="Submit">
    </form>
    </body>
</html>
```

login.php

```php
<?php

session_start();
$name = $_POST['name'];

if(!isset($name) OR empty($name)) {
   $name = "Gast";
    }

//Session registrieren
$_SESSION['username'] = $name;

//Text ausgeben
echo "Hallo $name <br>";
echo "<a href='seite2.php'>weitere Seite</a><br>";
echo "<a href='logout.php'>Logout</a>";
?>
```

seite2.php

```php
<?php
session_start();

if(!isset($_SESSION['username'])) {
   die("Bitte erst einloggen"); //Mit die beende den Scriptablauf
    }

//In $name den Wert der Session speichern
$name = $_SESSION['username'];

//Text ausgeben
echo "Du heißt immer noch: $name <br>";
echo "<a href='logout.php'>Logout</a>";
?>
```

logout.php

```php
<?php
session_start();
session_destroy();
echo "Logout erfolgreich";
?>
```