

Übung Terminkalender

Im Folgenden wollen wir eine Terminanzeige für den eigenen Internetauftritt programmieren, bei der der Besucher zwischen verschiedenen Sortierungen wählen kann.

Den fertigen Terminkalender als Vorgeschmack darauf, was wir nun erstellen. Dazu werden wir Schritt für Schritt vorgehen.

Die einzelnen Schritte:

Erstellen eines mehrdimensionalen Arrays mit Termin, Ort, Bandname

Kontrolle, ob bereits ein Sortierwunsch vorliegt; wenn nicht, sortieren nach Termin

Anzeige als Tabelle

Tabellenkopf mit Links zur Auswahl der Sortierung

bei der Ausgabe der Zeilen wird jede zweite Zeile als Lesehilfe eingefärbt

Datum in deutscher Schreibweise ausgeben

Bandname ausgeben

Ort ausgeben

Erstellen eines mehrdimensionalen Arrays mit Termin, Ort, Bandname

Übernehmen Sie bitte folgende Einträge, da sonst die Übung unnötig schwierig wird.

PHP-Quellcode: Termineinträge für Beispiel

```
$termin[] = array('Datum' => 20221208,  
                'Ort'   => „Wangen“,  
                'Band'  => „cOoL RoCk oPaS“);
```

```
$termin[] = array('Datum' => 20220311,  
                'Ort'   => „Stuttgart“,  
                'Band'  => „Die Hosenbodenband“);
```

```
$termin[] = array('Datum' => 20220628,  
                'Ort'   => „Tübingen“,  
                'Band'  => „flying socks“);
```

```
$termin[] = array('Datum' => 20220628,  
                'Ort'   => „Stuttgart“,  
                'Band'  => „flying socks“);
```

Bitte zwei weitere Auftritte hinzufügen.

Für die Sortierung von mehrdimensionalen Arrays wird dieser wie folgt vorbereitet:

PHP-Quellcode: für Sortierung vorbereiten

```
foreach ($termin as $nr => $inhalt)  
{  
    $band[$nr] = $inhalt['Band'];  
    $sort[$nr] = $inhalt['Ort'];  
    $datum[$nr] = $inhalt['Datum'];  
}
```

Kontrolle, ob bereits ein Sortierwunsch vorliegt;

wenn nicht, sortieren nach Termin

Die Sortierung wird vom Benutzer dadurch gewählt,

dass dieser einen Link anklickt. Im Link wird nach dem Dateiname „terminkalender.php“ nach dem „?“ der Variablenname „sortierung“ und ein Kürzel übergeben.

Beispiel

PHP-Quellcode: Art der Sortierung durch Nutzer auswählbar

```
<p>sortieren nach ...  
<a href="terminkalender.php?sortierung=d">Datum</a>  
<a href="terminkalender.php?sortierung=b">Band</a>  
<a href="terminkalender.php?sortierung=o">Ort</a>  
</p>
```

Der Variablenname kann in PHP über `$_GET[,sortierung']` abgefragt werden und in der Kontrollstruktur verwendet werden.

Als Kontrollstruktur nutzen wir hier switch - kontrolliert wird auf d (für Datum), b (für Band) und o (für Ort).

Die Sortierung erfolgt durch folgende Befehle:
(siehe mehrdimensionale Arrays)

PHP-Quellcode: Sortierung erfolgt durch folgende Befehle

```
// Sortierung nach Datum aufsteigend  
array_multisort($datum, SORT_ASC, $termin);
```

```
// Sortierung nach Band aufsteigend  
array_multisort($band, SORT_ASC, $termin);
```

```
// Sortierung nach Ort aufsteigend  
array_multisort($ort, SORT_DESC, $termin);
```

Nach dem Sortieren können Sie sich die Termine zur Kontrolle ausgeben lassen über:

PHP-Quellcode: Kontrolle des Inhalts des Array
`print_r ($termin);`

Erstellen der Tabelle

Lassen Sie das Array `$termin` ausgeben (sinnvoll ist, eine eigene Funktion für die Tabellenausgabe zu erstellen).

Innerhalb der Tabelle können Sie eine Variable hochzählen lassen - anhand dieser Zahl können Sie bestimmen, ob diese gerade oder ungerade ist und dann entsprechend die Zeile einfärben lassen.

Dazu bedienen wir uns der Mathematik und dem Modulo von 2 verwendet. Der Modulo ist eine mathematische Funktion, die als Ergebnis den Rest aus der Division zweier ganzer Zahlen angibt. Somit erhalten wir bei der Division durch 2 entweder 0 (somit haben wir eine gerade Zahl) oder nicht 0, somit haben wir eine ungerade Zahl.

PHP-Quellcode: Tabelle mit Zebramuster erzeugen

```
if ( bcmmod ( $zeilenr , 2' ) == 0 )  
{  
    $hintergrundfarbe = , bgcolor="#ACC8F0" ,;  
}  
else  
{  
    $hintergrundfarbe = , bgcolor="#DDE8F9" ,;  
}
```

Ausgabe des Datums in deutscher Form

Das Datum ist aus Gründen der Sortierbarkeit in der Form JahrMonatTag abgelegt - diesen String können Sie zerschneiden und wieder frisch zusammensetzen.

Dazu gibt es die PHP-Funktion substr.

Nach substr folgt die Variable, die behandelt werden soll, dann der Startpunkt und wieviele Stellen berücksichtigt werden.

Es ist auch möglich, von hinten zu zählen - siehe im folgenden Beispiel bei Tag (letzte 2 Stellen nehmen)

PHP-Quellcode: Datum in deutsch

```
$jahr = substr ( $datum, 0, 4 );  
$monat = substr ( $datum, 4, 2 );  
$tag = substr ( $datum, -2 );
```