



Die Switch-Anweisung

Description

Bei der [if-Anweisung](#) haben wir gelernt, wie wir bestimmte Bedingungen unterscheiden und entsprechende Befehle ausgeben können. Es gibt aber auch Fälle, bei denen if-Anweisungen viel zu kompliziert wären. Hier hilft oft die Switch-Anweisung.

Nehmen wir an, wir hätten ein Strategiespiel, einen Manager oder eine Wirtschaftssimulation, bei der es wichtig ist, dass der aktuelle Monat angezeigt wird. Außerdem hat nicht jeder Monat den selben Tag, also müssen wir wissen, welchen Monat wir haben um den entsprechenden Namen auszugeben und die korrekte Anzahl der Tage zu berücksichtigen.

Das Jahr hat bekanntlich 12 Monate. Nun könnten wir hin gehen, und dieses Problem mit if-Anweisungen lösen. Der Anfang würde dann so aussehen:

```
if (monat = 1)
{
    tage = 31;
    monatName = 'Januar';
}

if (monat = 2)
{
    tage = 28;
    monatName = 'Februar';
}
```

Wir bemerken dabei drei Probleme. Erstens, dass wir Glück haben, nur Monate zu unterscheiden und nicht gleich alle 365 Tage. Wenn dem so wäre, könnte man die Abfragen nicht mehr handhaben, was die zweite Feststellung ist und sofort zur dritten führt. Mit jedem weiteren *if* wird der Code immer komplexer und es braucht nicht viel Fantasie um zu merken, dass es für die Performance nicht unbedingt gut ist. Warum? Wenn wir den Januar haben, muss der Computer dennoch alle 12 Abfragen bearbeiten. Wenn wir das vermeiden wollen, müssen wir alles mit if-else verschachteln. Spätestens beim Juni wird der eine oder andere Programmierer den Code in die Tonne treten und dieses Feature aus dem Spiel streichen.

Das ist nicht notwendig! Mit **switch** kann man das elegant lösen. Mit switch fragen wir einen Wert ab. Der kann eine Zahl oder eine Zeichenkette sein. Dann rattert der Computer eine Liste ab, bis es gefunden wurde und führt die Anweisungen aus. Bis hierhin ist es noch kein großer Unterschied zu if. Der wesentliche Unterschied ist: bei switch kann man abbrechen, sobald das Richtige gefunden wurde. Der Rest wird nicht mehr bearbeitet. Für den Computer ist damit der Januar sehr entspannt und der Dezember nicht stressiger, als bei der if-Lösung. Und dazu ist es auch noch viel übersichtlicher.

Aufbau

```
switch (variable)
{
    case wert1:
        anweisung;
        break;
    case wert2:
        anweisung;
        break;
}
```

Nach **switch** folgt in der runden Klammer die Variable, die geprüft wird. Bei jedem **case** wird ein möglicher Wert / Zustand abgefragt. Wenn der stimmt, folgen Anweisungen darunter. Sobald diese Anweisungen erfolgt sind, kommt ein **break**. Ist case richtig, bricht switch beim nächsten break ab. Wenn also *wert1* richtig ist, wird *wert2* nicht mehr geprüft. Auch wenn darunter 100 weitere Werte folgen, passiert mit denen nichts mehr.

Auch am Aufbau merkt man, wie einfach dieser ist. Wir haben viel weniger Klammern und können uns so auf das Wesentliche konzentrieren.

Was ist aber, wenn kein Wert zutrifft? Dafür gibt es **default**, den man an das Ende setzt. Was nach default kommt, wird nur dann ausgeführt, wenn kein Wert zutrifft.

```
switch (variable)
{
    case wert1:
        anweisung;
        break;
    case wert2:
        anweisung;
        break;
    default: show_message('Da gibt es keinen passenden Wert!');
}
```

In dem Fall würde eine Nachricht kommen, wenn ein Wert kommt, der weder *wert1* noch *wert2* entspricht.

Die Monate

Jetzt schauen wir uns einmal an, wie das beim eingangs erwähnten Beispiel funktioniert. Da es ausgeschlossen ist, dass es einen Wert kleiner 1 und größer 12 geben wird, habe ich den default-Fall

weggelassen.

```
switch (monat)
{
    case 1:
        tage = 31;
        monatName = 'Januar';
        break;
    case 2:
        tage = 28;
        monatName = 'Februar';
        break;
    case 3:
        tage = 31;
        monatName = 'März';
        break;
    case 4:
        tage = 30;
        monatName = 'April';
        break;
    case 5:
        tage = 31;
        monatName = 'Mai';
        break;
    case 6:
        tage = 30;
        monatName = 'Juni';
        break;
    case 7:
        tage = 31;
        monatName = 'July';
        break;
    case 8:
        tage = 31;
        monatName = 'August';
        break;
    case 9:
        tage = 30;
        monatName = 'September';
        break;
    case 10:
        tage = 31;
        monatName = 'Oktober';
        break;
    case 11:
        tage = 30;
        monatName = 'November';
        break;
    case 12:
        tage = 31;
        monatName = 'Dezember';
        break;
}
```

Wenn wir das im Step-Event haben, können wir die richtigen Variablen für das Umschalten der Tage

verwenden und den korrekten Namen auf dem Bildschirm ausgeben.

Die switch-Anweisung wird noch in mehreren Tutorials verwendet. Das angehängte Beispiel zeigt den Monat. Mit Plus und Minus auf dem Nummernblock kann man die Monate hoch und runter schalten.



[switch-Anweisung](#)

1 Datei(en) 211.75 KB

[Download](#)

Date Created

21. Oktober 2016

Author

sven