

## Übung 3: ASCII Grafiken

In dieser Übung implementieren Sie Funktionen um mir ASCII Grafiken zu arbeiten. Hier ist ein Beispiel einer solchen Grafik:

```
####
###
###
####
## ###
##   ###
###   ###
```

Eine ASCII Grafik besteht also aus Text Zeichen. In Haskell können wir den Typ einer solchen Grafik als Liste von Strings definieren:

```
type Picture = [String]
```

In einem Haskell Script kann das obige Bild also wie folgt definiert werden:

```
lambda :: Picture
lambda = [
"      ",
"      ####",
"      ###",
"      ###",
"      ####",
"      ## ###",
"      ##   ###",
"      ###   ###",
"      "
]
```

Wir codieren das Bild also zeilenweise. Wie Sie bereits wissen ist ein String in Haskell auch nur eine Liste von Buchstaben [Char].

Auf dem AD finden Sie unter 03\_Lists/Assignment/ die Datei PicturesASCII.hs. Sie beinhaltet bereits die Typsignaturen für die folgende Aufgabe. Zusätzlich sind noch verschlüsselte Hinweise dabei. Versuchen Sie es aber zuerst ohne die Hinweise.

### 1) Implementieren Sie folgende Funktionen:

- a) Horizontal spiegeln: `flipH :: Picture -> Picture`
- b) Vertikal spiegeln: `flipV :: Picture -> Picture`
- c) Zwei Bilder übereinander: `above :: Picture -> Picture -> Picture`
- d) [Bonus I] Zwei Bilder nebeneinander: `beside :: Picture -> Picture -> Picture`
- e) [Bonus II] Gestalten Sie selber ein FProg ASCII Bild. Suchen Sie auf Google nach ASCII Art um sich inspirieren zu lassen.

### 2) Implementieren Sie folgende Funktionen

- a) Ein Element hinten in eine Liste einfügen.
- b) Reverse auf einer Liste, jedoch ohne das erste und ohne das letzte Element zu drehen. Sie können annehmen, dass die Liste mindestens eine Länge von zwei hat.
- c) Ein Element an einer bestimmten Position (Index) einfügen. Das erste Element hat den Index 0. Sie können annehmen, dass die Liste, in die eingefügt wird, lange genug ist.