

Fortgeschrittene funktionale Programmierung in Haskell

Übungszettel 4

Aufgabe 4.1:

In der Vorlesung haben wir besprochen, dass man eine Lens auch für nicht-existente Felder schreiben kann. Gegeben ein Zeitdatentyp

```
data TimeSplice = TS { days :: Int
                      , hours :: Int
                      , minutes :: Int
                      , seconds :: Float
                      }
                      deriving (Show, Eq)
```

Schreiben sie einen Isomorphismus, welcher eine Operation auf Sekunden vollführt, aber weiterhin einen TimeSlice nimmt:

```
convert :: Iso TimeSplice TimeSplice Float Float
convert = iso undefined
```

```
alterSeconds :: (Float -> Float) -> TimeSplice -> TimeSplice
alterSeconds = over convert
```