
Learn You a Haskell for Great Good!
Learn u a Haskell for great Good!

Haskell, eine rein funktionale Sprache

```haskell
qsort []     = []
qsort (x:xs) =
    qsort kleinere ++ [x] ++ qsort groessere
    where
    kleinere  = [y | y <- xs, y <= x]
    groessere = [y | y <- xs, y > x]
```

Haskell, eine rein funktionale Sprache

```haskell
qsort []     = []
qsort (x:xs) =
    qsort kleinere ++ [x] ++ qsort groessere
    where
    kleinere  = [y | y <- xs, y <= x]
    groessere = [y | y <- xs, y > x]
```

Die Fibonaccizahlen: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

```haskell
fibs = 0 : 1 : zipWith (+) fibs (tail fibs)
```

Haskell, eine rein funktionale Sprache

```haskell
qsort []     = []
qsort (x:xs) =
    qsort kleinere ++ [x] ++ qsort groessere
    where
    kleinere  = [y | y <- xs, y <= x]
    groessere = [y | y <- xs, y > x]
```

Die Fibonaccizahlen: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

```haskell
fibs = 0 : 1 : zipWith (+) fibs (tail fibs)
```

Statisches Typsystem mit Typerschließung

rein funktional • nebenläufig • lazy • 7000⁺ Pakete