

# 99 questions/Solutions/5

## From HaskellWiki

< 99 questions | Solutions

(\*) Reverse a list.

```
reverse      :: [a] -> [a]
reverse      = foldl (flip (:)) []
```

The standard definition, found in the prelude, is concise, but not very readable. Another way to define reverse is:

```
reverse :: [a] -> [a]
reverse [] = []
reverse (x:xs) = reverse xs ++ [x]
```

However this definition is more wasteful than the one in Prelude as it repeatedly recones the result as it is accumulated. The following variation avoids that, and thus computationally closer to the Prelude version.

```
reverse :: [a] -> [a]
reverse list = reverse' list []
  where
    reverse' [] reversed      = reversed
    reverse' (x:xs) reversed = reverse' xs (x:reversed)
```

And my favorite, although the most unreadable for sure :)

```
myReverse'' :: [a] -> [a]
myReverse'' xs = foldr (\x fId empty -> fId (x : empty)) id xs []
```

Retrieved from "[https://wiki.haskell.org/index.php?title=99\\_questions/Solutions/5&oldid=59316](https://wiki.haskell.org/index.php?title=99_questions/Solutions/5&oldid=59316)"

Category:

- Programming exercise spoilers

- 
- This page was last modified on 15 February 2015, at 13:21.
  - Recent content is available under a simple permissive license.