Haskell Cheatsheet

Overview of unique features

- Lazy evaluation
- Purely functional programming language
- Strong static typing
- Type inference
- Higher-order functions
- Pattern matching
- Monads

Variables

```haskell
-- Declare a variable
x = 42

-- Declare a constant
y :: Int
y = 10

-- Declare a list
myList = [1, 2, 3]

-- Declare a tuple
myTuple = (1, "hello")
```

Functions

```haskell
-- Declare a function
add :: Int -> Int -> Int
add x y = x + y

-- Call a function
result = add 3 4
```

Loops

Haskell uses recursion instead of loops.

```haskell
-- Define a recursive function to iterate over a list
sumList :: [Int] -> Int
sumList [] = 0
sumList (x:xs) = x + sumList xs
```

Conditionals

```haskell
-- Define an if-else statement
max :: Int -> Int -> Int
max x y = if x > y then x else y
```

File manipulation

```haskell
import System.IO

-- Open a file
main = do
  handle <- openFile "file.txt" ReadMode
  contents <- hGetContents handle
  putStr contents
  hClose handle
```

Resources

- Haskell documentation
- Learn You a Haskell for Great Good!
- Real World Haskell