

```
{:title "Haskell - Introduction" :layout :post :date "2016-09-25" :tags ["Haskell"
"Code"]}
```

[Haskell](#) is a purely functional programming language, statically typed, and lazy.



Purely Functional being every function in Haskell is a function in a mathematical sense. Meaning there is no side effects and a function receiving an argument will always do the same thing. This is very different from object oriented programs which is all about mutating data. Since Haskell is a Pure functional programming language you can not mutate variables. GHC is the compiler behind the Haskell language. You'll be testing out most of your code on a repl called ghci. The GHC is actually one behind C in terms of the fastest compilers. If you haven't already take a look at their website and download the Haskell Platform which includes GHC, Cabal and some other tools along with a set of libraries [here](#). Haskell is often times known to be the elite functional programming language as it is designed with mathematics in mind. Haskell is named after logician Haskell Curry and his books regarding logic. However that doesn't mean that you can only do math in Haskell. It is said that learning Haskell will however improve programming skills in general and open a new way of seeing and viewing code. Code in Haskell is brief and since it is pure it is easy to reuse code in Haskell. Like C Haskell is a strong static typed language but unlike C garbage collection is handled implicitly so you don't have to worry about those lingering pointers. So after you have downloaded the Haskell platform. Go ahead open up the GHCI by the

following...

```
ghci
```

And type ...

```
putStrLn "My first line of code in Haskell!"
```

You have written a line of code in Haskell!