

- 1) header: en la cabecera está el logo de la página y hay 4 links para hacer a algunas partes de la página y una imagen de un ejemplo de función en haskell



main content :mestra distintas características del lenguaje

Features

Statically typed

Every expression in Haskell has a type which is determined at compile time. All the types composed together by function application have to match up. If they don't, the program will be rejected by the compiler. Types become not only a form of guarantee, but a language for expressing the construction of programs.

[Click to expand](#)

Purely functional

Every function in Haskell is a function in the mathematical sense (i.e., "pure"). Even side-effecting IO operations are but a description of what to do, produced by pure code. There are no statements or instructions, only expressions which cannot mutate variables (local or global) nor access state like time or random numbers.

[Click to expand](#)

Type inference

You don't have to explicitly write out every type in a Haskell program. Types will be inferred by unifying every type bidirectionally. However, you can write out types if you choose, or ask the compiler to write them for you for handy documentation.

[Click to expand](#)

Concurrent

Haskell lends itself well to concurrent programming due to its explicit handling of effects. Its flagship compiler, GHC, comes with a high-performance parallel garbage collector and light-weight concurrency library containing a number of useful concurrency primitives and abstractions.

[Click to expand](#)

Lazy

Functions don't evaluate their arguments. This means that programs can compose together very well, with the ability to write control constructs (such as if/else) just by writing normal functions. The purity of Haskell code makes it easy to fuse chains of functions together, allowing for performance benefits.

Packages

Open source contribution to Haskell is very active with a wide range of packages available on the public package servers.

[Click to expand](#)

2)

<https://www.haskell.org/>

si porque tiene:

- protocol
- subdomain
- domain
- top level domain

3) el objetivo es promover la educación y el progreso científico sobre el lenguaje de programación de haskell

4)

a) "Every function in haskell is a function in the mathematica sense"

"cada función en haskell es una función en el sentido matemático"

b) el verbo "is"

c) "la funciones en haskell están expresadas en términos matemáticos"

"todas la funciones de haskell pueden ser descritas como funciones matemáticas"

5)

a)

Purely functional Every function/ in/ Haskell/ is/ a function /in/ the mathematical sense/ (i.e., "pure")./ Even side-effecting IO operations/ are/ but/ a description/ of/ what/ to/ do/, produced by/ pure code/. There/ are/ no/ statements /or instructions/, only expressions /which /cannot mutate /variables/ (local or global) / nor/ access state /like /time/ or/ random numbers.

b)

"a **function** in the mathematical sense" = "una función en el sentido matemático"

"random **numbers**" = números aleatorios

"pure **code**" = código puro

6) "integer" = entero

"char" = caracter

"bool" = booleano

"lazy" = perezoso

"type" = tipo

"decode" = decodificar

"hosting" = alojamiento

"compiler" = compilador