PureScript

PureScript is a strongly-typed, purely-functional programming language that compiles to JavaScript. It can be used to develop web applications, server side apps, and also desktop applications with use of Electron. Its syntax is mostly comparable to that of Haskell. In addition, it introduces row polymorphism and extensible records.[2] Also, contrary to Haskell, PureScript adheres to a strict evaluation strategy.

Contents

PureScript	
Paradigm	Functional
Designed by	Phil Freeman
First appeared	2013
Stable release	0.14.0 / March 1, 2021
Typing discipline	Inferred, static, strong
License	BSD[1]
Filename extensions	.purs
Website	www.purescript.org (http://www.purescript.org)
Influenced by	
Haskell, JavaScript	

History

PureScript was initially designed by Phil Freeman in 2013. He started to work on PureScript as various attempts to compile Haskell to JavaScript while preserving its semantics (e.g. using Fay, Haste, or GHCJS) didn't work to his satisfaction.[3]

Since then it has been picked up by the community and is developed on GitHub. Additional core tools developed by the community include the dedicated build tool "Pulp",[4] the documentation directory "Pursuit",[5] and the package manager "Spago"[6]

Features

PureScript features strict evaluation, persistent data structures and type inference. The PureScript type system shares many features with those of similar functional languages like Haskell: algebraic data types and pattern matching, higher kinded types, type classes and functional dependencies, and higher-rank polymorphism. PureScript's type system adds support for row polymorphism and extensible records.[7] However, PureScript lacks support for some of the more advanced features of Haskell like GADTs and type families.

The PureScript compiler attempts to produce readable JavaScript code, where possible. Through a simple FFI interface, it also allows the reuse of existing JavaScript code.[8]

PureScript supports incremental compilation, and the compiler distribution includes support for building source code editor plugins for iterative development.[9] Editor plugins exist for many popular text editors, including Vim, Emacs, Sublime Text, Atom and Visual Studio Code.

PureScript supports type-driven development via its typed holes feature,[10] in which a program can be constructed with missing subexpressions. The compiler will subsequently attempt to infer the types of the missing subexpressions, and report those types to the user. This feature has inspired similar work in the GHC Haskell compiler.[11]

Examples

Here is a minimal "Hello world" program in PureScript:

```
1  module Main where
2
3  import Effect.Console (log)
4
5  main = log "Hello World!"
```

Here, the type of the program is inferred and checked by the PureScript compiler. A more verbose version of the same program might include explicit type annotations:

```
1  module Main where
2
3  import Prelude
4
5  import Effect (Effect)
6  import Effect.Console (log)
7
8  main :: Effect Unit
9  main = log "Hello World!"
```

See also

- Elm: A domain-specific programming language for declaratively creating web browser-based graphical user interfaces
- Reason: A syntax extension and toolchain for OCaml that can also compile to JavaScript

References

1. "purescript/purescript" (https://github.com/purescript/purescript/blob/master/LICENSE). *GitHub*.
2. "PureScript" (http://www.purescript.org/). *www.purescript.org*. Retrieved 2019-03-14.
3. "Read PureScript by Example | Leanpub" (https://leanpub.com/purescript/read#leanpub-auto-about-the-author). *leanpub.com*. Retrieved 2019-03-14.
4. *A build tool for PureScript projects* (https://github.com/purescript-contrib/pulp), PureScript Contrib, 2019-03-08, retrieved 2019-03-14
5. "Pursuit" (https://pursuit.purescript.org/). *pursuit.purescript.org*. Retrieved 2019-03-14.
6. 🍝 *PureScript package manager and build tool powered by Dhall and package-sets: spacchetti/spago* (https://github.com/spacchetti/spago), spacchetti, 2019-03-14, retrieved 2019-03-14
7. *Documentation for the PureScript language, compiler, and tools.: purescript/documentation* (https://github.com/purescript/documentation), PureScript, 2019-03-14, retrieved 2019-03-14

8. *Documentation for the PureScript language, compiler, and tools.: purescript/documentation* (https://github.com/purescript/documentation), PureScript, 2019-03-14, retrieved 2019-03-14

9. *purs ide: Editor and tooling support for the PureScript programming language* (https://github.com/purescript/purescript/tree/dfbd19da85c92d82ed4c3184f950f74da5332e3f/psc-ide), 2019-04-21, retrieved 2019-04-21

10. *Implementing type directed search for PureScript*, 2016-10-27

11. Matthías Páll Gissurarson (2018-09-27), *Suggesting Valid Hole Fits for Typed-Holes (Experience Report)* (https://dl.acm.org/citation.cfm?id=3242760)

External links

- Official website (http://www.purescript.org)
- PureScript Playground (https://app.qvault.io/playground/purescript)