

The Futhark Programming Language

High-performance purely functional data-parallel array programming

Overview

Examples

Docs

Publications

Gotta Go Fast!

Get Involved

Blog

Why Futhark?

Because it's nicer than writing CUDA or OpenCL by hand!

Futhark is a small programming language designed to be compiled to efficient parallel code. It is a **statically typed, data-parallel, and purely functional array language** in the ML family, and comes with a **heavily optimising ahead-of-time compiler** that presently generates either GPU code via [CUDA](#) and [OpenCL](#), or multi-threaded CPU code. As a simple example, this function computes the average of an array of 64-bit floating-point numbers:

```
def average (xs: []f64) = reduce (+) 0 xs / f64.i64 (length xs)
```

Futhark is not designed for graphics programming, but can instead use the compute power of the GPU to accelerate data-parallel array computations. The language supports **regular nested data-parallelism** and **automatic differentiation**, as well as a form of imperative-style in-place modification of arrays, while still preserving the purity of the language via the use of a **uniqueness type system**.

While the Futhark language and compiler is an **ongoing research project**, it is quite usable for real programming, and can compile nontrivial programs which then run on real machines at high speed.

Futhark is a simple language and is designed to be easy to learn, although it omits some common features in order to generate high-performance parallel code. Nevertheless, **Futhark can already be used for**

Quick links

[Playground](#)

[REPL](#)

[Installation](#)

[Tutorial](#)

[User's guide](#)

[Language reference](#)

[Prelude](#)

[More packages](#)

nontrivial programs. We are actively looking for more potential applications as well as people who are interested in contributing to the language design.

Futhark is not intended to replace existing general-purpose languages. The intended use case is that Futhark is only used for relatively small but compute-intensive parts of an application. The Futhark compiler generates code that can be **easily integrated with non-Futhark code**. For example, you can compile a Futhark program to a Python module that internally uses [PyOpenCL](#) to execute code on the GPU, yet looks like any other Python module from the outside ([more on this here](#)). The Futhark compiler will also generate more conventional C code, which can be accessed from any language with a basic FFI ([an example here](#)).

For more information, you can look at [code examples](#), details on [performance](#), our [devblog](#), or maybe the [docs](#), which also contains a list of our [publications](#). You can of course also visit our [main repository on Github](#), or our repository of [benchmarks](#).

Developed at [DIKU](#) ([copyright](#), [acknowledgements](#), [hedgehogs](#))