Contents

Introduction

This tutorial is all about viash! What is viash?

viash is software that helps you turn a script into a reusable component, which you can use as a standalone executable or as part of a pipeline.

Use cases for viash

Phew! What does that mean? Here are a few typical use cases have already caused indescribable frustrations among software developers, but can be solved quite easily by using viash.

- You developed a Jupyter notebook report for a data analysis. You wish to share it with your colleague, only to spend two hours installing your Conda stack on their laptop.
- You want to combine a couple of tools in a pipeline and every tool has specific requirements on how they should be run. Even worse: some requirements might directly conflict with each other.
- Your next data analysis project is very similar to the previous project, so you copy and paste the source code. Unfortunately, you detect a bug in some of your code, so now you need to go back and fix the same bug in all the different projects.
- You want to look back at a data analysis you performed two years ago. Unfortunately, the software you used back then is not supported anymore, or the newest version produces totally different results.

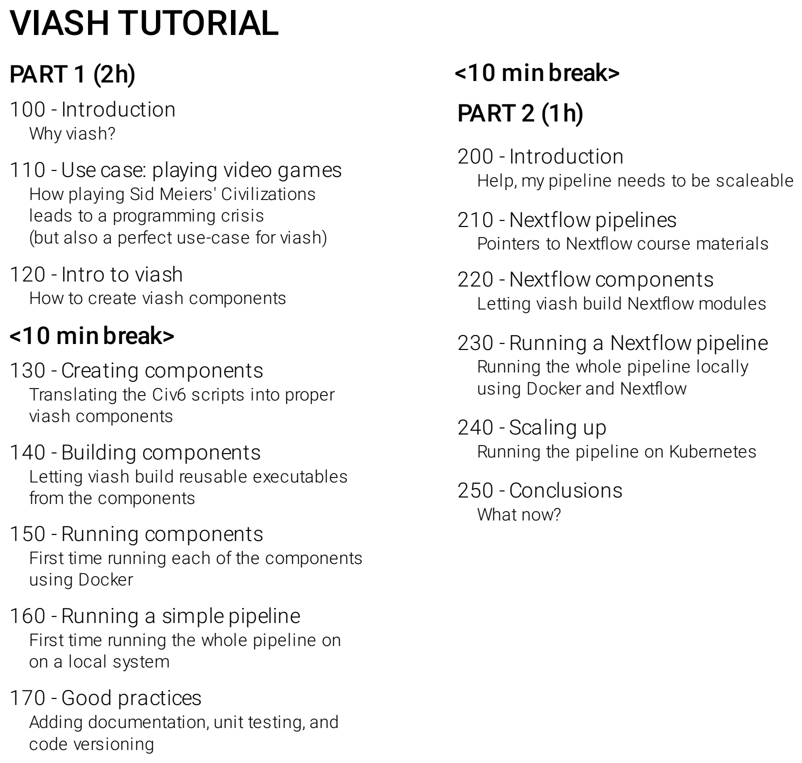
How viash works

By providing some meta-data regarding its functionality and the platform on which you want to run the software, viash can help you:

- wrap your script in an executable with a CLI and `--help` functionality,
- seamlessly execute your component natively on the host platform or in a Docker container,

- combine multiple components in a Nextflow pipeline, and
- unit-test your component to ensure that it works at all times.

Outline of this tutorial



VIASH TUTORIAL

PART 1 (2h)

100 - Introduction
Why viash?

110 - Use case: playing video games
How playing Sid Meiers' Civilizations
leads to a programming crisis
(but also a perfect use-case for viash)

120 - Intro to viash
How to create viash components

<10 min break>

130 - Creating components
Translating the Civ6 scripts into proper
viash components

140 - Building components
Letting viash build reusable executables
from the components

150 - Running components
First time running each of the components
using Docker

160 - Running a simple pipeline
First time running the whole pipeline on
on a local system

170 - Good practices
Adding documentation, unit testing, and
code versioning

<10 min break>

PART 2 (1h)

200 - Introduction
Help, my pipeline needs to be scaleable

210 - Nextflow pipelines
Pointers to Nextflow course materials

220 - Nextflow components
Letting viash build Nextflow modules

230 - Running a Nextflow pipeline
Running the whole pipeline locally
using Docker and Nextflow

240 - Scaling up
Running the pipeline on Kubernetes

250 - Conclusions
What now?