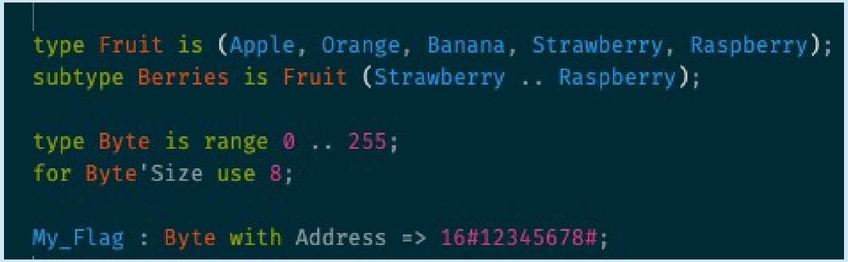
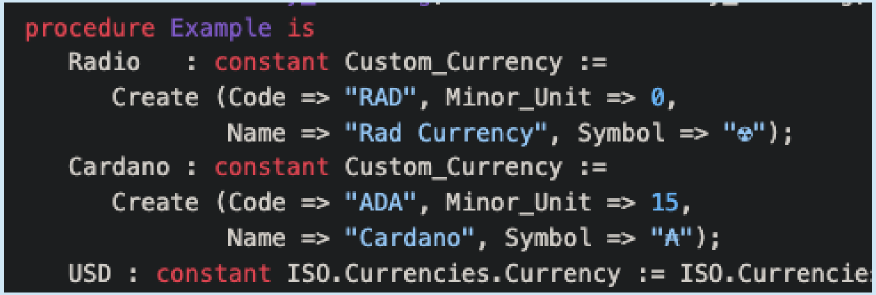
Get started with Ada in 2 minutes or less!

A.J. Ianozi

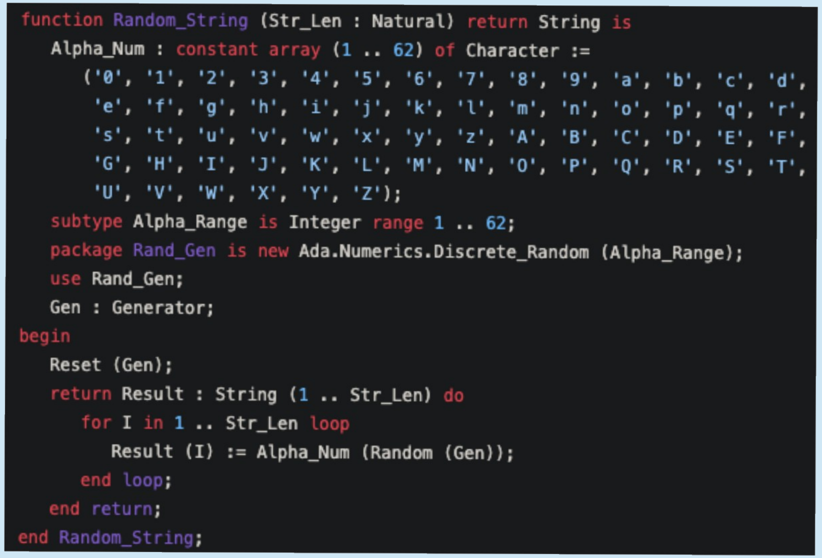
Why Ada? - Some Background



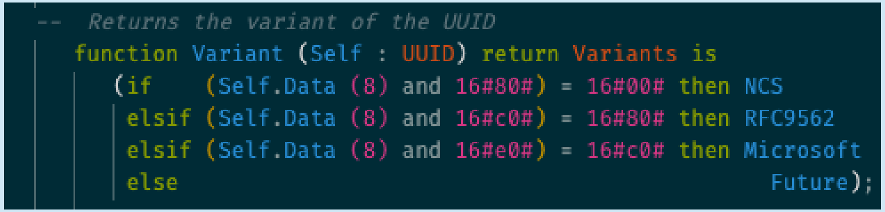
```ada
type Fruit is (Apple, Orange, Banana, Strawberry, Raspberry);
subtype Berries is Fruit (Strawberry .. Raspberry);

type Byte is range 0 .. 255;
for Byte'Size use 8;

My_Flag : Byte with Address => 16#12345678#;
```



```ada
procedure Example is
   Radio   : constant Custom_Currency :=
      Create (Code => "RAD", Minor_Unit => 0,
                 Name => "Rad Currency", Symbol => "☢");
   Cardano : constant Custom_Currency :=
      Create (Code => "ADA", Minor_Unit => 15,
                 Name => "Cardano", Symbol => "₳");
   USD : constant ISO.Currencies.Currency := ISO.Currencie
```



```ada
function Random_String (Str_Len : Natural) return String is
   Alpha_Num : constant array (1 .. 62) of Character :=
      ('0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd',
       'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',
       's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F',
       'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T',
       'U', 'V', 'W', 'X', 'Y', 'Z');
   subtype Alpha_Range is Integer range 1 .. 62;
   package Rand_Gen is new Ada.Numerics.Discrete_Random (Alpha_Range);
   use Rand_Gen;
   Gen : Generator;
begin
   Reset (Gen);
   return Result : String (1 .. Str_Len) do
      for I in 1 .. Str_Len loop
         Result (I) := Alpha_Num (Random (Gen));
      end loop;
   end return;
end Random_String;
```



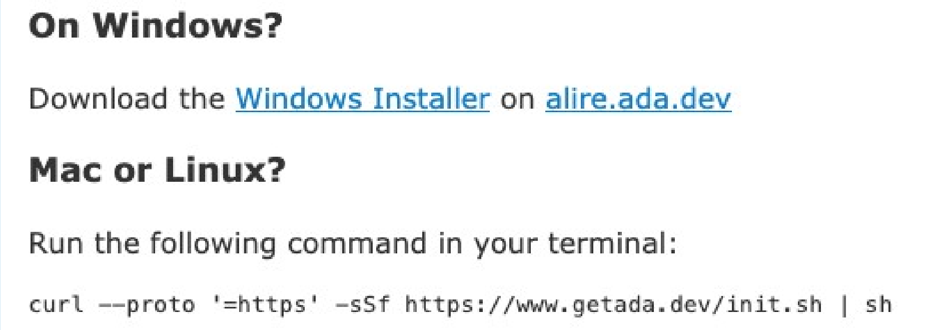
```ada
--  Returns the variant of the UUID
   function Variant (Self : UUID) return Variants is
      (if    (Self.Data (8) and 16#80#) = 16#00# then NCS
       elsif (Self.Data (8) and 16#c0#) = 16#80# then RFC9562
       elsif (Self.Data (8) and 16#e0#) = 16#c0# then Microsoft
       else                                           Future);
```

Why Rust and not Ada?

- Easy to install/setup: https://rustup.rs/, run this command, congratulations, you now have cargo.
  - Ada on the other hand: Which license does gnat pro use? Community edition? What runtime license?
- In Rust it's easy to start a new project:
  cargo new foo; cd foo; cargo build
  - In Ada this is solved with Alire but that's only recently.
- Rust makes it easy to find a library for your project: https://crates.io/
  - In Ada, Alire has also solved this problem! https://alire.ada.dev/crates.html
- Rust had accessible documentation and a vibrant community, even for absolute beginner coders. https://doc.rust-lang.org/book/
  - Check out https://learn.adacore.com/

Ada’s transition to modern tooling

- Community Edition Retired
  - No more confusing licensing
- Alire: Ada now has a package manager!
  - alr init –bin foo; cd foo; alr build
- Getada.dev: The Unofficial Installer for Alire!
  "a door to open rather then a hill to climb"



On Windows?

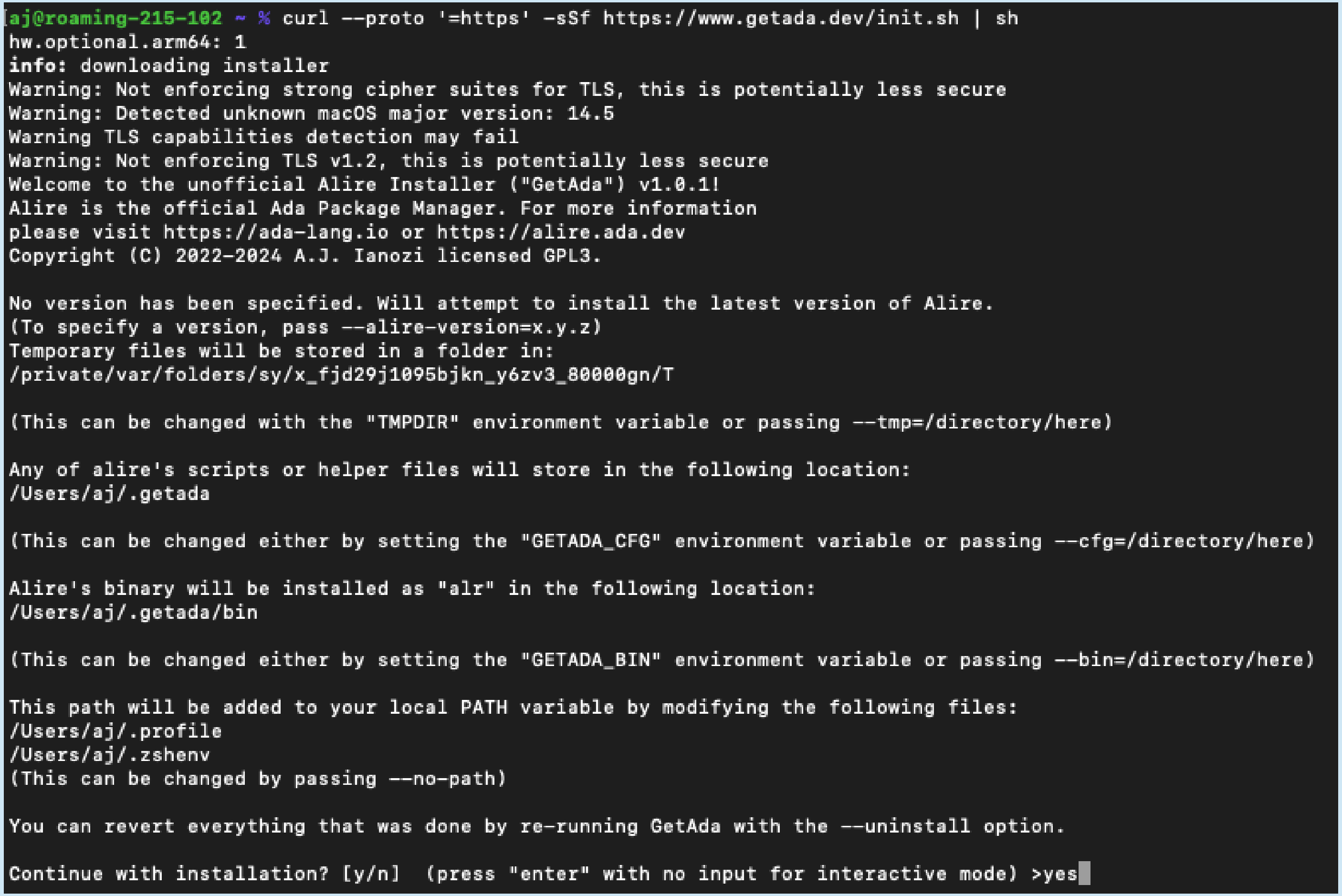
Download the Windows Installer on alire.ada.dev

Mac or Linux?

Run the following command in your terminal:

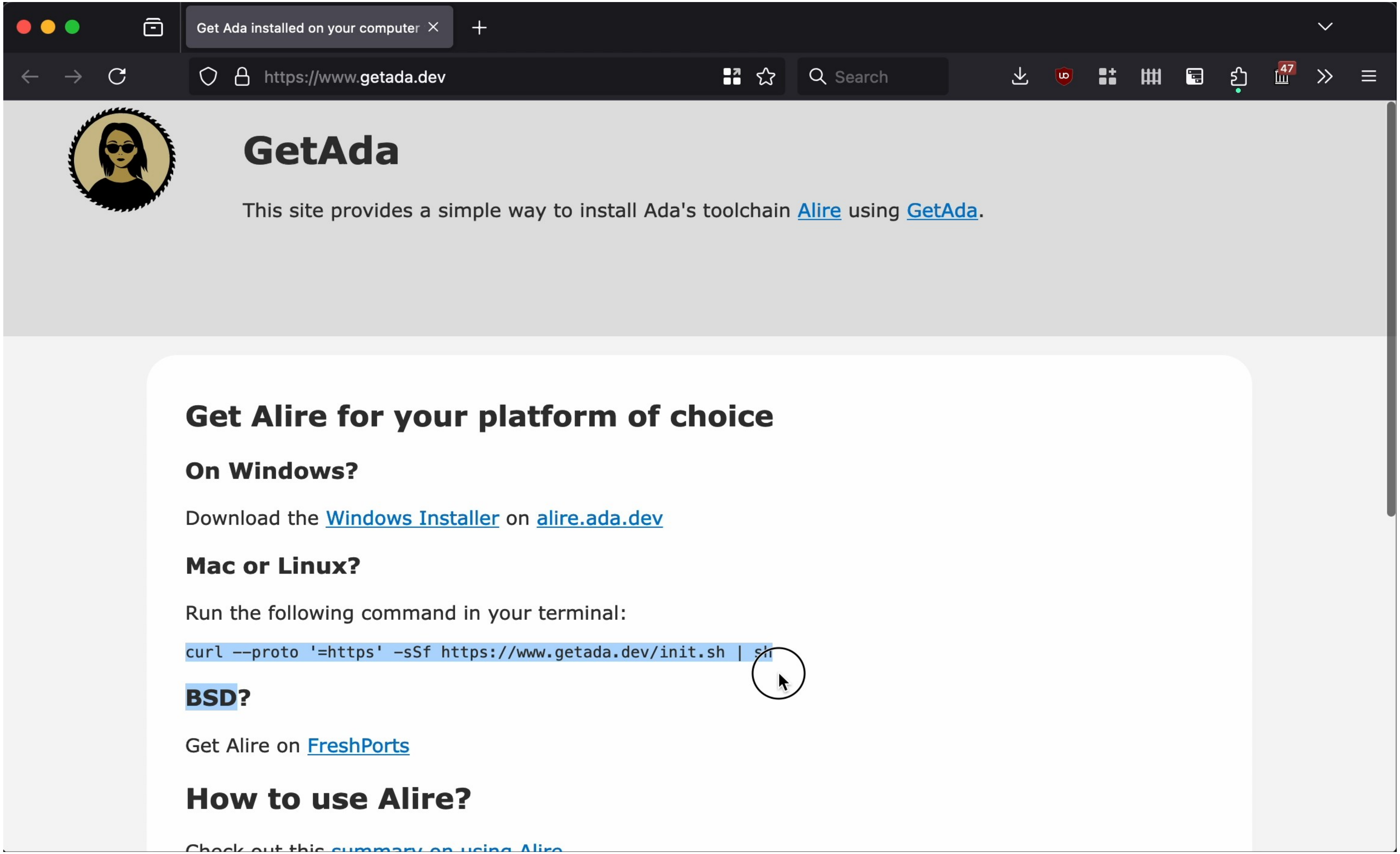
```
curl --proto '=https' -sSf https://www.getada.dev/init.sh | sh
```

Getada



```
aj@roaming-215-102 ~ % curl --proto '=https' -sSf https://www.getada.dev/init.sh | sh
hw.optional.arm64: 1
info: downloading installer
Warning: Not enforcing strong cipher suites for TLS, this is potentially less secure
Warning: Detected unknown macOS major version: 14.5
Warning TLS capabilities detection may fail
Warning: Not enforcing TLS v1.2, this is potentially less secure
Welcome to the unofficial Alire Installer ("GetAda") v1.0.1!
Alire is the official Ada Package Manager. For more information
please visit https://ada-lang.io or https://alire.ada.dev
Copyright (C) 2022-2024 A.J. Ianozi licensed GPL3.

No version has been specified. Will attempt to install the latest version of Alire.
(To specify a version, pass --alire-version=x.y.z)
Temporary files will be stored in a folder in:
/private/var/folders/sy/x_fjd29j1095bjkn_y6zv3_80000gn/T

(This can be changed with the "TMPDIR" environment variable or passing --tmp=/directory/here)

Any of alire's scripts or helper files will store in the following location:
/Users/aj/.getada

(This can be changed either by setting the "GETADA_CFG" environment variable or passing --cfg=/directory/here)

Alire's binary will be installed as "alr" in the following location:
/Users/aj/.getada/bin

(This can be changed either by setting the "GETADA_BIN" environment variable or passing --bin=/directory/here)

This path will be added to your local PATH variable by modifying the following files:
/Users/aj/.profile
/Users/aj/.zshenv
(This can be changed by passing --no-path)

You can revert everything that was done by re-running GetAda with the --uninstall option.

Continue with installation? [y/n]  (press "enter" with no input for interactive mode) >yes
```

From Nothing to Hello World in Two Minutes or Less



GetAda

This site provides a simple way to install Ada's toolchain Alire using GetAda.

Get Alire for your platform of choice

On Windows?

Download the Windows Installer on alire.ada.dev

Mac or Linux?

Run the following command in your terminal:

```
curl --proto '=https' -sSf https://www.getada.dev/init.sh | sh
```

BSD?

Get Alire on FreshPorts

How to use Alire?

Check out this summary on using Alire

Thank you!

Questions?

A.J. Ianozi
aj@ianozi.com
https://getada.dev