

Nix Devshells

“

It works on my machine

”

Random commands

```
$ pacman -S node  
$ npm i
```

Undeclerative

- Imperativ
- Selbe commands immer wieder

Es gibt Hoffnung

Niemand braucht sich zu fürchten

Nix



nixpkgs

- Mega aktives GitHub-Repo
- Riesig, unzählige (aktuelle) packages
- Sonst häufig leicht selber zu packagen
 - Insbesondere mit Makefile
 - Erst recht mit buildern: rust, go, ...
- Leicht zu nutzen: `pkgs.firefox`

Dev shells


```
{pkgs ? import <nixpkgs> {}}:
pkgs.mkShell {
  name = "rust-dev-shell";

  buildInputs = with pkgs; [
    cargo
    rustc
    rustfmt
    rustPackages.clippy
    bacon
    rust-analyzer
  ];

  shellHook = ''
    echo "Welcome to the Rust development environment!"
  '';

  # Environment variablen
  CARGO_INCREMENTAL = 1';
}
```

Was ist mit genauen Versionen?

Flakes

- Vergleichbar mit normalen Programmier Package Managern
- **Lock file** ⇒ Feste Versionen ⇒ Vollständig Reproduzierbar
- Man kann viele Nix sachen auf einmal festlegen
 - Packages
 - Devshells
 - NixOs Modules
- In andere Flakes importierbar

- Ein attribute set mit `inputs` setzen
- `outputs` ist eine Funktion
 - Jeden Input als "parameter"
 - Rückgabe ist ein attribute set mit `devShell` ,
`packages` , ...

```
{
  inputs = {
    nixpkgs.url = "github:NixOS/nixpkgs/nixpkgs-unstable";
  };

  outputs = {
    self,
    nixpkgs,
  }: let
    system = "x86_64-linux";
    pkgs = nixpkgs.legacyPackages.${system};
  in {
    devShell.${system}.default = pkgs.mkShell {
      buildInputs = with pkgs; [
        cargo
        rustc
        rustfmt
        pre-commit
        rustPackages.clippy
        bacon
      ];
      RUST_SRC_PATH = pkgs.rustPlatform.rustLibSrc;
    };
  }
}
```

```
{
  inputs = {
    nixpkgs.url = "github:NixOS/nixpkgs/nixpkgs-unstable";
  };

  outputs = {
    self,
    nixpkgs,
  }: let
    system = "x86_64-linux";
    pkgs = nixpkgs.legacyPackages.${system};
  in {
    devShell.${system}.default = pkgs.mkShell {
      buildInputs = with pkgs; [
        cargo
        rustc
        rustfmt
        pre-commit
        rustPackages.clippy
        bacon
      ];
      RUST_SRC_PATH = pkgs.rustPlatform.rustLibSrc;
    };
  }
}
```

```

{
  inputs = {
    nixpkgs.url = "github:NixOS/nixpkgs/nixpkgs-unstable";
  };

  outputs =
  {
    self,
    nixpkgs,
  }:
  let
    forAllSystems =
      function:
      nixpkgs.lib.genAttrs [
        "aarch64-darwin"
        "aarch64-linux"
        "x86_64-darwin"
        "x86_64-linux"
      ] (
        system: function nixpkgs.legacyPackages.${system}
      );
  in
  {
    devShells = forAllSystems (pkgs: {
      default = pkgs.mkShell {
        buildInputs = with pkgs; [
          cargo
          rustc
          rustfmt
          rustPackages.clippy
          bacon
          rust-analyzer
        ];
        RUST_SRC_PATH = pkgs.rustPlatform.rustLibSrc;
      };
    });
  };
}

```

Nützliche Nix(OS) Ressourcen

- **Vimjoyer YouTube**
- Fremden Nix code
 - Gezielte GitHub Code Suchen
 - Nixpkgs source code
- noogle.dev
 - Ähnlich wie hoogle für haskell
 - Erleichtert das finden von Funktionen
- Nix Leute Sachen Fragen
 - z.b. Auf dem NixOS Discord
- Gerne mit mir über Nix quatschen <3

Danke <3