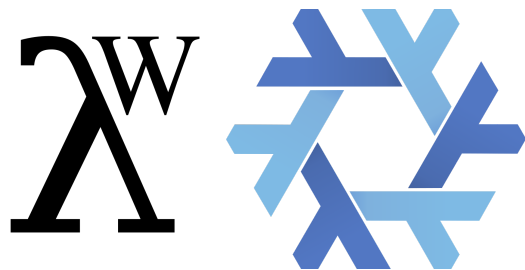


Can't Hack This

A hard-headed introduction to Nix



Patryk Wychowaniec

Sekrety Elektroniki



Sekrety Elektroniki



Sekrety Elektroniki



Nix(OS)

Me, Myself and I

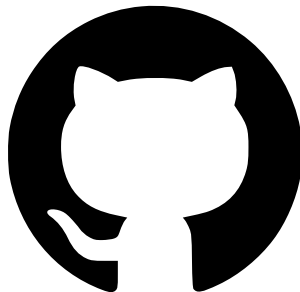
My name's Patryk Wychowaniec, a.k.a. Patryk27:



keybase.io/patryk27



reddit.com/u/patryk27



github.com/patryk27



[4programmers.net \(patryk27\)](https://4programmers.net/patryk27)

Plan

- Theoretical: What's Nix?
- Practical: How I'm using Nix on a daily basis?

Nix

- Nix (as a language)
- Nix (as a package manager),
- NixOS (as an operating system based on Nix).

Nix, The Language

Nix, The Language

Nix is a lazy, functional, and pure language.

Nix, The Language

Nix has strings:

```
"catch" + " 22"
```

```
# catch 22
```

NIX

```
"catch" + " 22"
```

```
// catch 22
```

JAVASCRIPT

Nix, The Language

Nix has integers and booleans:

```
2*2 > 3*3
```

```
# false
```

NIX

```
2*2 > 3*3
```

```
// false
```

JAVASCRIPT

Nix, The Language

Nix has a structural, static type system:

```
100 > "25"
```

```
# error: cannot compare a string with an integer
```

NIX

```
100 > "25"
```

```
// true
```

JAVASCRIPT

Nix, The Language

Nix has functions:

```
builtins.isInt "twenty two"  
# false
```

NIX

```
Number.isInteger("twenty two")  
// false
```

JAVASCRIPT

Nix, The Language

```
x: x * x
```

NIX

```
function (x) {  
  return x * x;  
};
```

```
(x) => x * x;
```

```
x => x * x;
```

JAVASCRIPT

Nix, The Language

```
(x: x * x) 4
```

```
# 16
```

NIX

```
(x => x * x)(4)
```

```
// 16
```

JAVASCRIPT

Nix, The Language

```
(x: y: x * y) 3 4
```

```
# 12
```

NIX

```
(x: y: x * y) 3
```

```
# «lambda @ (string):1:5»
```

```
function fn(x) {  
  return function (y) {  
    return x * y;  
  }  
}
```

```
fn(3)(4)
```

```
// 12
```

JAVASCRIPT

Nix, The Language

Nix has `lists`:

```
[ 1 2 ] ++ [ "three" ]  
# [ 1 2 "three" ]
```

NIX

```
[ 1, 2 ].concat("three")
```

JAVASCRIPT

Nix, The Language

Nix has attribute sets:

```
{  
  name = "chrome";  
  version = "1024-beta";  
}
```

NIX

```
{  
  name: "chrome",  
  version: "1024-beta",  
}
```

JAVASCRIPT

Nix, The Language

Attribute sets can be recursive:

```
rec {  
  full-name = name + " " + version;  
  name = "chrome";  
  version = "1024-beta";  
}
```

NIX

Nix, The Language

Nix has paths:

foo/bar/zar

/home/patryk/somewhere/foo/bar/zar

NIX

./.

/home/patryk/somewhere

NIX

Nix, The Language

Nix has "constants" (in so-called `let binding`):

```
let
  square = x: x * x;

in
  square 4

# 16
```

NIX

Nix, The Language

NIX

let

```
say = { what, who }: "life is a ${what}, ${who}";
```

in

```
say {  
  what = "cabaret";  
  who = "old chum";  
}
```

```
# "life is a cabaret, old chum"
```

Nix, The Language

NIX

let

```
say = {  
  what ? "cabaret",  
  who,  
  ...  
}: "life is a ${what}, ${who}";
```

in

```
say {  
  who = "old chum";  
  why = "dunno";  
}
```

```
# "life is a cabaret, old chum"
```

Nix, The Package Manager

Nix, The Package Manager

\$ bash

SHELL

Nix, The Package Manager

```
$ echo "$PATH"
```

```
# /usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

SHELL

Nix, The Package Manager

```
$ which bash  
# /usr/local/bash
```

SHELL

Nix, The Package Manager

```
$ ldd $(which bash)
# libtinfo.so.6
#  => /lib/x86_64-linux-gnu/libtinfo.so.6
#
# libdl.so.2
#  => /lib/x86_64-linux-gnu/libdl.so.2
```

SHELL

Nix, The Package Manager

/bin
/boot
/dev
/etc
/home
/lib
...

SHELL

Nix, The Package Manager

```
$ which bash
```

SHELL

```
# /nix/store/i2qc2285icqrsri78saafssp2vjrr5mh-bash-  
interactive-4.4-p23/bin/bash
```

```
$ ldd $(which bash)
```

SHELL

```
# libreadline.so.7
```

```
# => /nix/store/3xzvmd7y4rjnkbyhdjalvq7462x30p3r-  
readline-7.0p5/lib/libreadline.so.7
```

```
#
```

```
# libdl.so.2
```

```
# => /nix/store/qvf11lymvw6n8g66xgj1wsm28z1viqdv-glibc-  
2.30/lib/libdl.so.2
```

Nix, The Package Manager

/nix/store/\$hash-\$name

SHELL

Nix, The Package Manager

```
$ ldd $(which sample-app-1)
# leftpad.so
# => /nix/store/abcabcabc-leftpad-1/lib/leftpad.so
```

SHELL

```
$ ldd $(which sample-app-2)
# leftpad.so
# => /nix/store/xyzxyzxyz-leftpad-2/lib/leftpad.so
```

SHELL

Nix, Language + Package Manager

NIX

```
mkDerivation {  
  pname = "netris";  
  version = "0.52";  
  src = fetchFromGitHub { /* ... */ };  
  buildInputs = [ ncurses ];  
  
  configureScript = "./Configure";  
  
  installPhase = ''  
    mkdir -p $out/bin  
    cp ./netris $out/bin  
  '';  
}
```

Nix

<https://github.com/NixOS/nixpkgs>

Nix

Let's get practical, practical.

nix-shell -p

SHELL

```
$ node --version
```

```
# command not found: node
```

```
$ nix-shell -p nodejs
```

```
[nix-shell] $ node --version
```

```
[nix-shell] # v12.18.1
```

```
[nix-shell] $ exit
```

```
$ node --version
```

```
# command not found: node
```

nix-shell -p

```
$ nix-shell -p firefox  
[nix-shell] $ ./firefox
```

SHELL

or

```
$ echo 'main = putStrLn "Hello, World!'" > example.hs  
$ nix-shell -p ghc  
[nix-shell] $ ghc ./example.hs  
[nix-shell] $ ./example  
[nix-shell] # Hello, World!
```

SHELL

```

$ apt install haskell-platform
[...]
The following NEW packages will be installed:
  alex binutils binutils-common binutils-x86-64-linux-gnu ca-certificates cabal-install cpp cpp-9
  curl
  freeglut3 freeglut3-dev gcc gcc-9 gcc-9-base ghc happy haskell-platform hscolor krb5-locales
  libasan5
  libasn1-8-heimdal libatomic1 libbinutils libbrotli1 libbsd-dev libbsd0 libc-dev-bin libc6-dev
  libcc1-0
  libcrypt-dev libctf-nobfd0 libctf0 libcurl4 libdrm-amdgpu1 libdrm-common libdrm-intel1 libdrm-
nouveau2
  libdrm-radeon1 libdrm2 libedit2 libegl-dev libegl-mesa0 libegl1 libelf1 libexpat1 libffi-dev
  libgbm1
  libgcc-9-dev libghc-async-dev libghc-attoparsec-dev libghc-call-stack-dev libghc-case-insensitive-
dev
  libghc-erf-dev libghc-fgl-dev libghc-fixed-dev libghc-gluraw-dev libghc-glut-dev libghc-half-dev
  libghc-hashable-dev libghc-haskell-src-dev libghc-html-dev libghc-http-dev libghc-hunit-dev
  libghc-integer-logarithms-dev libghc-network-dev libghc-network-uri-dev libghc-objectname-dev
  libghc-opengl-dev libghc-openglraw-dev libghc-parallel-dev libghc-primitive-dev libghc-
quickcheck2-dev
  libghc-random-dev libghc-regex-base-dev libghc-regex-compat-dev libghc-regex-posix-dev
  libghc-scientific-dev libghc-split-dev libghc-statevar-dev libghc-syb-dev libghc-tf-random-dev
  libghc-unordered-containers-dev libghc-vector-dev libghc-zlib-dev libgl-dev libgl1 libgl1-mesa-dev
  libgl1-mesa-dri libglapi-mesa libgles-dev libgles1 libgles2 libglu1-mesa libglu1-mesa-dev
  libglvnd-dev
  libglvnd0 libglx-dev libglx-mesa0 libglx0 libgmp-dev libgmpxx4ldbl libgomp1 libgssapi-krb5-2
  libgssapi3-heimdal libhcrypto4-heimdal libheimbase1-heimdal libheimntlm0-heimdal libhx509-5-
heimdal
  libice-dev libice6 libisl22 libitm1 libk5crypto3 libkeyutils1 libkrb5-26-heimdal libkrb5-3
  libkrb5support0 libldap-2.4-2 libldap-common libllvm9 liblsan0 libmpc3 libmpfr6 libncurses-dev
  libncurses5-dev libnghttp2-14 libopengl-dev libopengl0 libpciaccess0 libpsl5 libpthread-stubs0-dev
  libquadmath0 libroken18-heimdal librtmp1 libsasl2-2 libsasl2-modules libsasl2-modules-db
  libsensors-config libsensors5 libsm-dev libsm6 libsqlite3-0 libssh-4 libssl1.1 libtsan0 libubsan1
  libvulkan1 libwayland-client0 libwayland-server0 libwind0-heimdal libx11-6 libx11-data libx11-dev
  libx11-xcb1 libxau-dev libxau6 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-
randr0
  libxcb-sync1 libxcb-xfixes0 libxcb1 libxcb1-dev libxdamage1 libxdmcp-dev libxdmcp6 libxext-dev
  libxext6 libxfixes-dev libxfixes3 libxi-dev libxi6 libxmu-dev libxmu-headers libxmu6 libxshmfence1
  libxt-dev libxt6 libxxf86vm1 linux-libc-dev manpages manpages-dev mesa-vulkan-drivers openssl

```

nix-shell -p

```
$ vim --version  
# VIM - Vi IMproved 8.2
```

SHELL

```
$ nix-shell \  
  -p vim \  
  -I nixpkgs=https://github.com/NixOS/nixpkgs-  
channels/archive/4bba6650a6a5a2009e25bdeed8c1e871601a9bfb  
.tar.gz
```

```
[nix-shell] $ vim --version  
[nix-shell] # VIM - Vi IMproved 8.0
```

```
$ apt-cache madison vim  
# vim | 2:8.1.2269-1ubuntu5
```

SHELL

nix-shell -p

SHELL

```
function drun() {  
    docker run \  
        --rm \  
        -v "$(pwd):/mnt" \  
        "${@:2}" \  
        -it "$1" \  
        sh -c "cd /mnt; bash"  
}
```

```
$ drun php:latest
```

nix-shell

```
$ nix-shell
```

SHELL

nix-shell

`nix-shell` to the rescue: dependencies

nix-shell

shell.nix:

```
let
  pkgs = import <nixpkgs> {};

in
  pkgs.mkShell {
    buildInputs = [
      pkgs.openssl
      pkgs.pkg-config
    ];

    LD_LIBRARY_PATH="${pkgs.openssl.out}/lib";
  }
```

NIX

Terminal:

```
$ nix-shell
[nix-shell] $ cargo build
```

SHELL

nix-shell

NIX

let

```
pkgs = import <nixpkgs> {};
```

```
python-with-packages = pkgs.python3.withPackages  
  (python-packages: [  
    python-packages.pandas  
    python-packages.requests  
  ]);
```

in

```
pkgs.mkShell {  
  buildInputs = [  
    python-with-packages  
  ];  
}
```

nix-shell

NIX

let

```
pkgs = import <nixpkgs> {};
```

```
python-packages = pp: [  
  pp.pandas  
  pp.requests  
];
```

```
python-with-packages = pkgs.python3.withPackages  
python-packages;
```

in

```
pkgs.mkShell {  
  buildInputs = [  
    python-with-packages  
  ];  
}
```

nix-shell

Extra stuff

- <https://github.com/target/lorri> (*must-have!*)
- <https://github.com/haslersn/any-nix-shell>
- `nix run` command

nix search

```
$ nix search meme
# * nixpkgs.meme (meme-unstable)
#   A command line utility for creating image macro style
memes
#
# * nixpkgs.wownero (wownero-0.7.0)
#   A privacy-centric memecoin that was fairly launched
on April 1, 2018 with no pre-mine, stealth-mine or ICO

$ nix search git 'frontend|gui'
# ^ supports fancy queries too
```

SHELL

```
$ nix-shell -p meme
```

SHELL

nix search

Extra stuff

- <https://github.com/bennofs/nix-index>

nix-build

nix-build

nix-build

NIX

```
let  
  pkgs = import <nixpkgs> {};  
  
in  
  pkgs.stdenv.mkDerivation {  
    /* ... */  
  }
```

nix-build

NIX

```
/* ... */

pkgs.stdenv.mkDerivation {
  pname = "netris";
  version = "0.52";

  /* ... */
}
```

nix-build

NIX

```
/* ... */

pkgs.stdenv.mkDerivation {
  /* ... */

  src = pkgs.fetchFromGitHub {
    owner = "naclander";
    repo = "netris";
    rev = "677...d2b"; # cut to fit
    sha256 = "0gm...9hm"; # cut to fit
  };

  /* ... */
}
```

nix-build

NIX

```
/* ... */

pkgs.stdenv.mkDerivation {
  /* ... */

  buildInputs = [ pkgs.ncurses ];

  /* ... */
}
```

nix-build

NIX

```
/* ... */

pkgs.stdenv.mkDerivation {
  /* ... */

  configureScript = "./Configure";
  dontAddPrefix = true;

  /* ... */
}
```

nix-build

NIX

```
/* ... */

pkgs.stdenv.mkDerivation {
  /* ... */

  installPhase = ''
    mkdir -p $out/bin
    cp ./netris $out/bin
  '';
}
```

nix-build

default.nix:

```
let
  pkgs = import <nixpkgs> {};

in
  pkgs.stdenv.mkDerivation {
    pname = "netris";
    version = "0.52";

    src = pkgs.fetchFromGitHub {
      owner = "naclander";
      repo = "netris";
      rev = "6773c9b2d39a70481a5d6eb5368e9ced6229ad2b";
      sha256 = "0gmxbpn50pnffidwjchkszph9rh2jm4wfq7hj8msp5vhdq5h0z9hm";
    };

    buildInputs = [ pkgs.ncurses ];

    configureScript = "./Configure";
    dontAddPrefix = true;

    installPhase = ''
      mkdir -p $out/bin
      cp ./netris $out/bin
    '';
  }
```

NIX

nix-build

NIX

```
let
  pkgs = import <nixpkgs> {};
  deps = import ./deps.nix;

in
  pkgs.stdenv.mkDerivation {
    name = "stdout";
    src = ./src;

    buildInputs = with deps; [ asciidoctor hugo rsync sass ];

    phases = [ "buildPhase" ];

    buildPhase = ''
      /* ... */

      hugo -s "$out/src" --gc --minify

      /* ... */
    '';
  }
```


nix-build

NIX

```
let
  pkgs = import <nixpkgs> {};

in
  pkgs.stdenv.mkDerivation {
    name = "pdfs";
    src = ./.;

    phases = [ "buildPhase" ];

    buildInputs = with pkgs; [ ocrmypdf ];

    buildPhase = ''
      mkdir $out

      for input in $src/*.pdf; do
        name=$(basename "$input")
        output="$out/$name"

        echo $input
        echo $output

        ocrmypdf -l pol --force-ocr "$input" "$output"
      done
    '';
  }
```

nix-build

Extra stuff

- nix-bundle
- nix-copy-closure
- <https://github.com/nix-community/nixpkgs-fmt>

home-manager

home-manager

```
vim
```

home-manager

NIX

```
{ ... }: {  
  programs = {  
    vim = {  
      enable = true;  
  
      extraConfig = ''  
        filetype plugin indent on  
        set tabstop=4  
        set shiftwidth=4  
        set expandtab  
      '';  
    };  
  };  
}
```

```
{ ... }: {
  programs = {
    vscode = {
      enable = true;

      extensions = with pkgs.vscode-extensions; [
        bbenoist.Nix
      ] ++ pkgs.vscode-utils.extensionsFromVscodeMarketplace [
        {
          name = "sort-lines";
          publisher = "Tyriar";
          version = "1.9.0";
          sha256 = "0l4wibsjnlbzbrrl1wcj18vnm1q4ygvxmh347jvzziv8f1l790qjl";
        }
        {
          name = "vscode-openc1";
          publisher = "galarius";
          version = "0.6.5";
          sha256 = "1w6g8q5x72dy8g5hbrzr9kpsv7m2nnjarszdp69g37dbpmsrdcn2";
        }
      ];
    };
  };
}
```

```
{ pkgs, ... }:  
  
let  
  kitty-themes = builtins.fetchGit {  
    url = "https://github.com/dexpota/kitty-themes.git";  
    rev = "fca3335489bdbab4cce150cb440d3559ff5400e2";  
  };  
  
in  
{  
  programs = {  
    kitty = {  
      enable = true;  
  
      font.package = pkgs.fira-code;  
  
      extraConfig = ''  
        include ${kitty-themes}/themes/gruvbox_dark.conf  
  
        kitty_mod ctrl+alt  
  
        font_size 12.0  
        font_family FiraCode-Light  
  
        /* ... */  
      '';  
    };  
  };  
}
```

```
{ pkgs, ... }:  
  
let  
  k = builtins.fetchGit { /* ... */ };  
  
in  
{  
  fonts.fonts = with pkgs; [ powerline-fonts ];  
  
  programs = {  
    autojump.enable = true;  
  
    zsh = {  
      enable = true;  
      autocd = true;  
      enableAutosuggestions = true;  
      enableCompletion = true;  
  
      oh-my-zsh = {  
        enable = true;  
  
        plugins = [  
          "autojump"  
          /* ... */  
        ];  
  
        extraConfig = ''  
          COMPLETION_WAITING_DOTS="true"  
          DISABLE_AUTO_TITLE="true"  
          FZF_BASE="$(fzf-share)"  
          /* ... */  
        '';  
      };  
    };  
  };  
}
```


home-manager

Get started

```
$ curl -L https://nixos.org/nix/install | sh
```

SHELL

- <https://nixcloud.io/tour>
- <https://nixos.org/nixos/nix-pills>
- <https://learnxinyminutes.com/docs/nix/>
- <https://www.sam.today/blog/environments-with-nix-shell-learning-nix-pt-1/>

Can't Hack This
~ Patryk Wychowaniec, 2020

<https://fp-wroc.github.io/>

Thank you!