# Nix

## Язык, пакетный менеджер и экосистема

# Александр Бантьев

# Вступление

## Что такое nix?



Figure 1: Nix

Nix - это чистый функциональный язык с динамическим типизированием и пакетный менеджер на основе этого языка.

Философия Nix как пакетного менеджера - атомарность, повторяемость и чистота сборки.

Как язык, Nix отличается упором на ленивые вычисления и ориентированность на данные: большая часть кода на Nix представляет из себя напоминающие JSON множества аттрибутов (attrsets).

#### Экосистема

## Основы Nix

### **Nix** как язык

Рассмотрим основы синтаксиса Nix.

## Краткая инструкция для желающих повторить

```
$ curl https://nixos.org/nix/install | sh
<...>
```

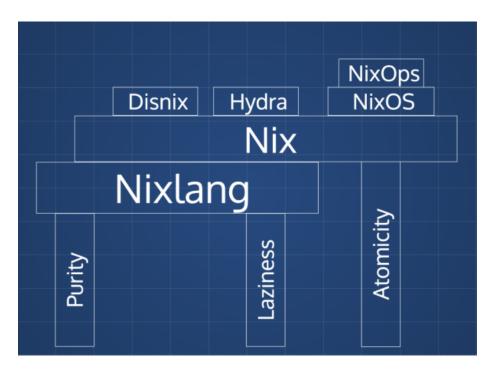


Figure 2: Экосистема

```
$ nix repl
```

## Числа и арифметика

```
nix-repl> 1 + 2 - 3 * 2
-3
nix-repl> 1 / 2
0
nix-repl> builtins.add 1 2
3
```

#### Замечание

Арифметические операторы лучше обносить пробелами! У символов / и – есть особые значения в Nix

```
nix-repl> 6/2
/home/balsoft/6/2
```

## Строки

## Переменные и стороковая интерполяция

```
nix-repl> foo = "hello" # Special syntax of nix-repl
nix-repl> "${foo}, world!"
"hello, world!"
```

#### Замечание

Названия переменных могут содержать знак -

```
nix-repl> foo-bar = 10
nix-repl> foo-bar
10
```

## Как избежать интерполяции?

Довольно часто встречаются ситуации, когда мы хотим вставить в строку последовательность \${...} (например, в bash-код).

```
nix-repl> "\${foo}"
"${foo}"
nix-repl> ''test ''${foo} test''
"test ${foo} test"
```

### Списки

Списки в Nix иммутабельны и могут содержать значения любых типов вперемешку.

```
nix-repl> [ 2 "foo" true (2+3) ]
[ 2 "foo" true 5 ]
```

#### Замечание

Обратите внимание: элементы списка не разделены запятыми!

```
nix-repl> [ 1 + 2 ]
error: syntax error, unexpected '+', at (string):1:5
nix-repl> [ (1 + 2) ]
[ 3 ]
```

## Заключение

## Ссылки

https://nixos.org/ https://imgur.com/0gPQ4zg