

Недостатки BEAM и Эликсир

- ▶ Недостатки BEAM

- ▶ Скорость выполнения кода
- ▶ Небольшая экосистема
- ▶ Не мейнстрим

- ▶ Недостатки Эликсир

- ▶ Проблемы в дизайне языка
- ▶ Динамическая типизация
- ▶ Разница между исходным кодом и байткодом

Скорость выполнения кода

Языки по скорости выполнения:

- ▶ интерпретируемые (**Python, Ruby**)
- ▶ компилируемые в байткод (**Elixir, Java**)
- ▶ компилируемые в нативный код (**Go, C++**)

Скорость выполнения кода

Эликсир быстрее **Python** и **Ruby**,
но медленнее **Java** и **Go**.

Скорость выполнения кода

Это про скорость одного потока на одном ядре.

А у нас многопоточность.

Тут важнее средства языка,

архитектура виртуальной машины и приложения.

Небольшая экосистема

Библиотеки, инструменты и сообщество разработчиков.

Небольшая экосистема

Количество репозиторийов на Github:

- ▶ Эрланг – **20K**
- ▶ Эликсир – **36K**
- ▶ Ruby – **1.5M**
- ▶ JavaScript – **5M**

Небольшая экосистема

Качество важнее количества.

Не мейнстрим

Компании не хотят использовать технологию, в которой мало разработчиков.

Это риск для бизнеса.

Разработчики не хотят изучать технологию, которую используют мало компаний.

Это риск для карьеры.

Проблемы в дизайне языка

Тщательно продуманные дизайн, семантика,
синтаксис и реализация – это не про Эликсир.
(мнение Эрлангистов :)

Проблемы в дизайне языка

Вызов функций без круглых скобок:

```
my_fun(arg1 , arg2)  
my_fun arg1 , arg2
```

Проблемы в дизайне языка

Два синтаксиса для словарей:

```
m1 = %{a: 42}
```

```
m1.a
```

```
m2 = %{"a" => 42}
```

```
m2["a"]
```

Проблемы в дизайне языка

Однострочный *do..end*:

```
my_fun(a, b), do: a + b
```

```
my_fun(a, b) do  
  a + b  
end
```

Проблемы в дизайне языка

Вызов анонимной функции:

```
my_fun = fn(a) -> a * a end  
my_fun.(42)
```

Проблемы в дизайне языка

Хорошая новость в том, что ко всему привыкаешь :)

Динамическая типизация

Статическая типизация поддерживается в некоторой степени.

Dialyzer ограничен в возможностях, и его легко игнорировать.

Динамическая типизация

Языки для BEAM со статической типизацией:

Alpaka, Gleam, Haskerl, Caramel.

Не зрелые, не для реальных проектов.

Разница между исходным кодом и байткодом

Абстракции высокого уровня,
широкое использование макросов.
Код сильно отличается от байткода.

Разница между исходным кодом и байткодом

Трудно понять, что именно работает в рантайме.

Это усложняет интроспекцию и трассировку.