

Быстро о быстром

Роман Дворнов, Ostrovok.ru

Новосибирск, 12 апреля 2014

О себе

- Работаю в [Ostrovok.ru](https://ostrovok.ru)
- Автор фреймворка basis.js



Что делать чтобы работало
быстро?

Если откинуть все лишнее,
то **основные** затраты это
представления и работа с данными

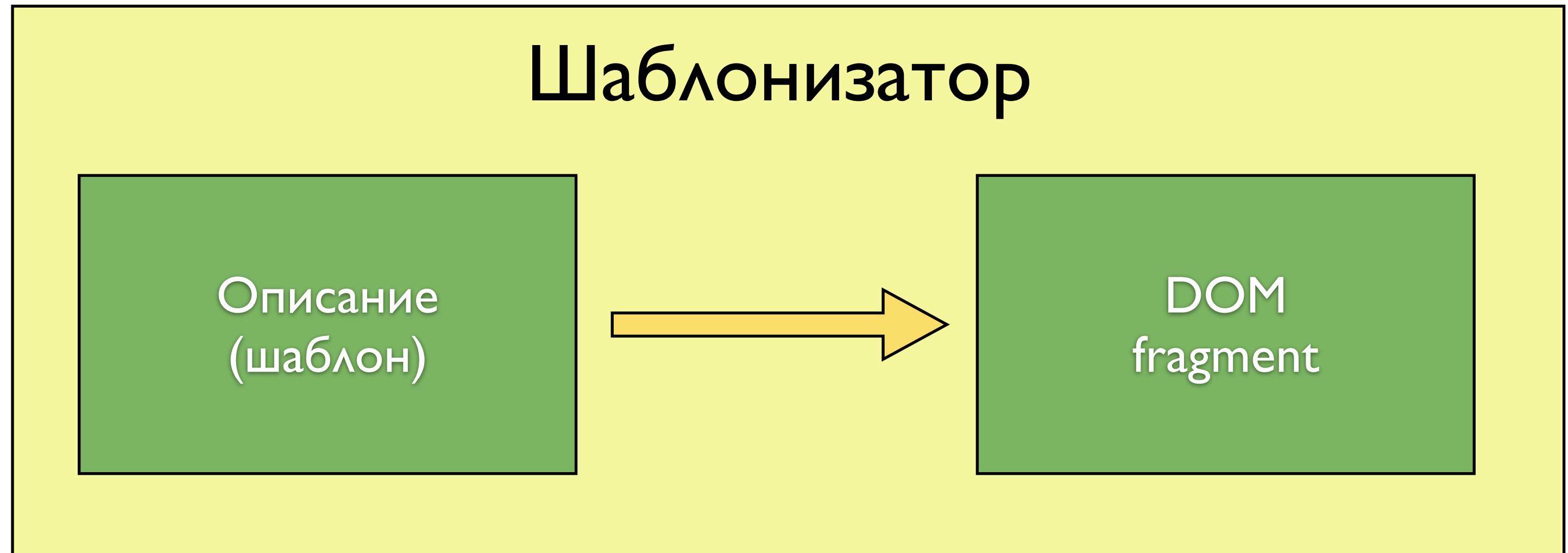
Представления

Самая **ресурсоемкая** задача –
создание и обслуживание **DOM**

Для упрощения работы
с DOM используются
шаблонизаторы

Шаблонизаторы

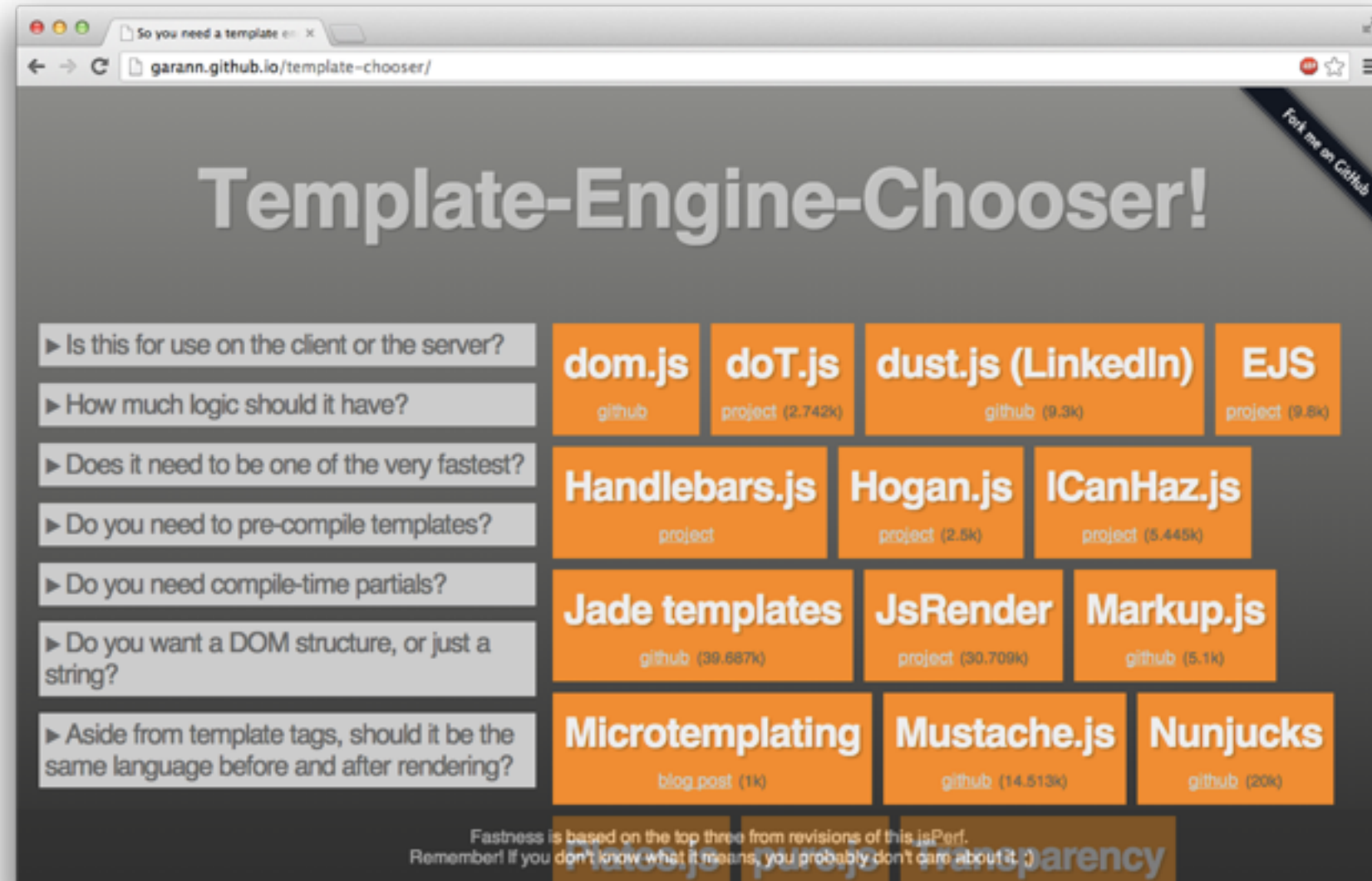
Задача шаблонизатора



Чтобы не делать так...

```
function renderList(data){  
  var html = '';  
  html = html + '<ul class="mylist' +  
    (cls ? ' ' + cls : '') + '>';  
  for (var i = 0; i < items.length; i++)  
    html = html + '<li class="item ' +  
      (items[i].selected ? ' selected' : '') + '>' +  
      items[i].title +  
      '</li>';  
  html = html + '</ul>';  
  return html;  
};  
...  
element.innerHTML = renderList({ .. });
```

Шаблонизаторов много



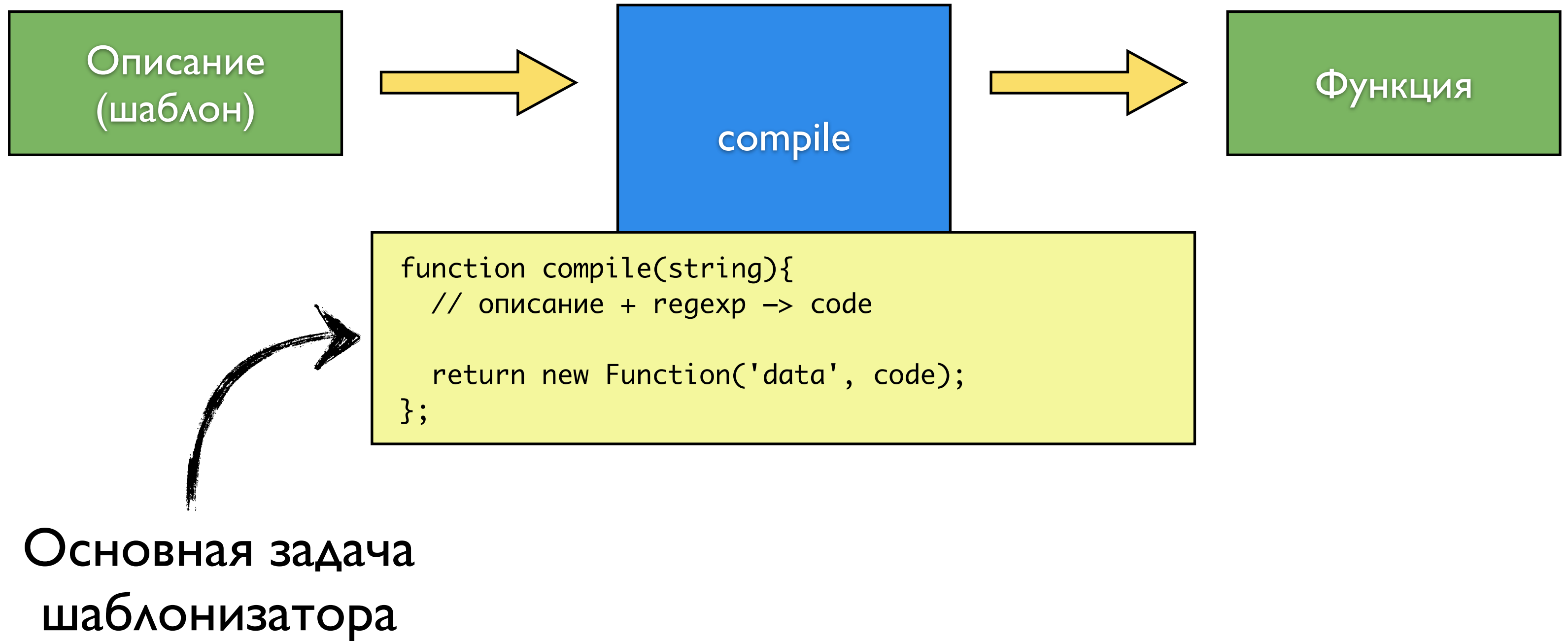
garann.github.io/template-chooser/



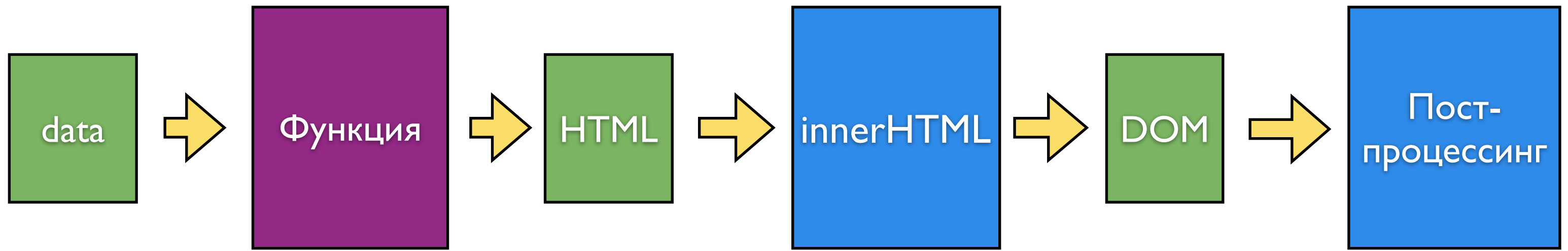
Я тут **новый шаблонизатор** замутил...

Принцип работы
один и тот же

Подготовка



Производство



А это на ваших плечах, или плечах фреймворка

Плюсы подхода

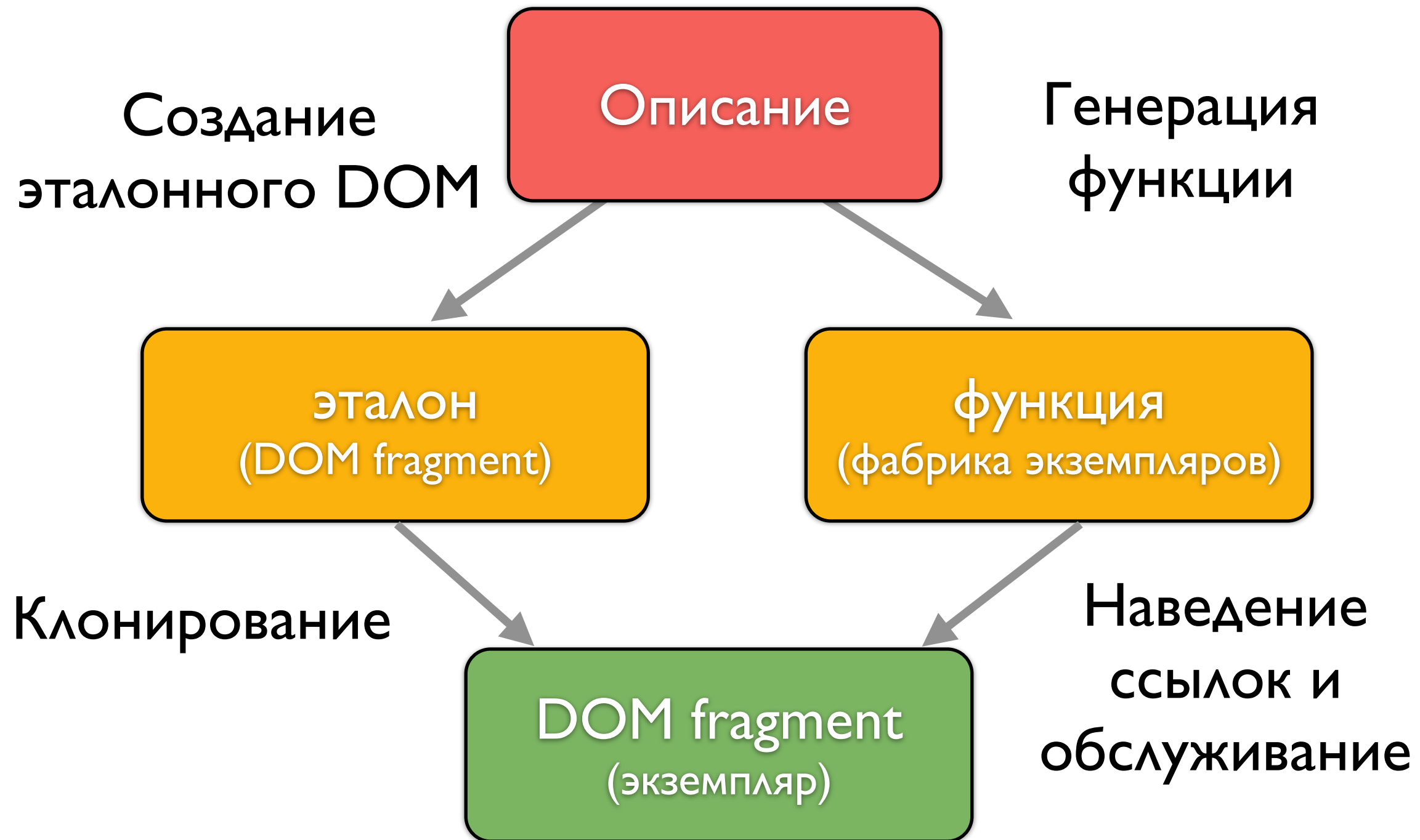
- Работает
- Привычно
- Просто

Здесь был список
из 146 пунктов насколько этот
подход **неэффективен**



А как надо то?

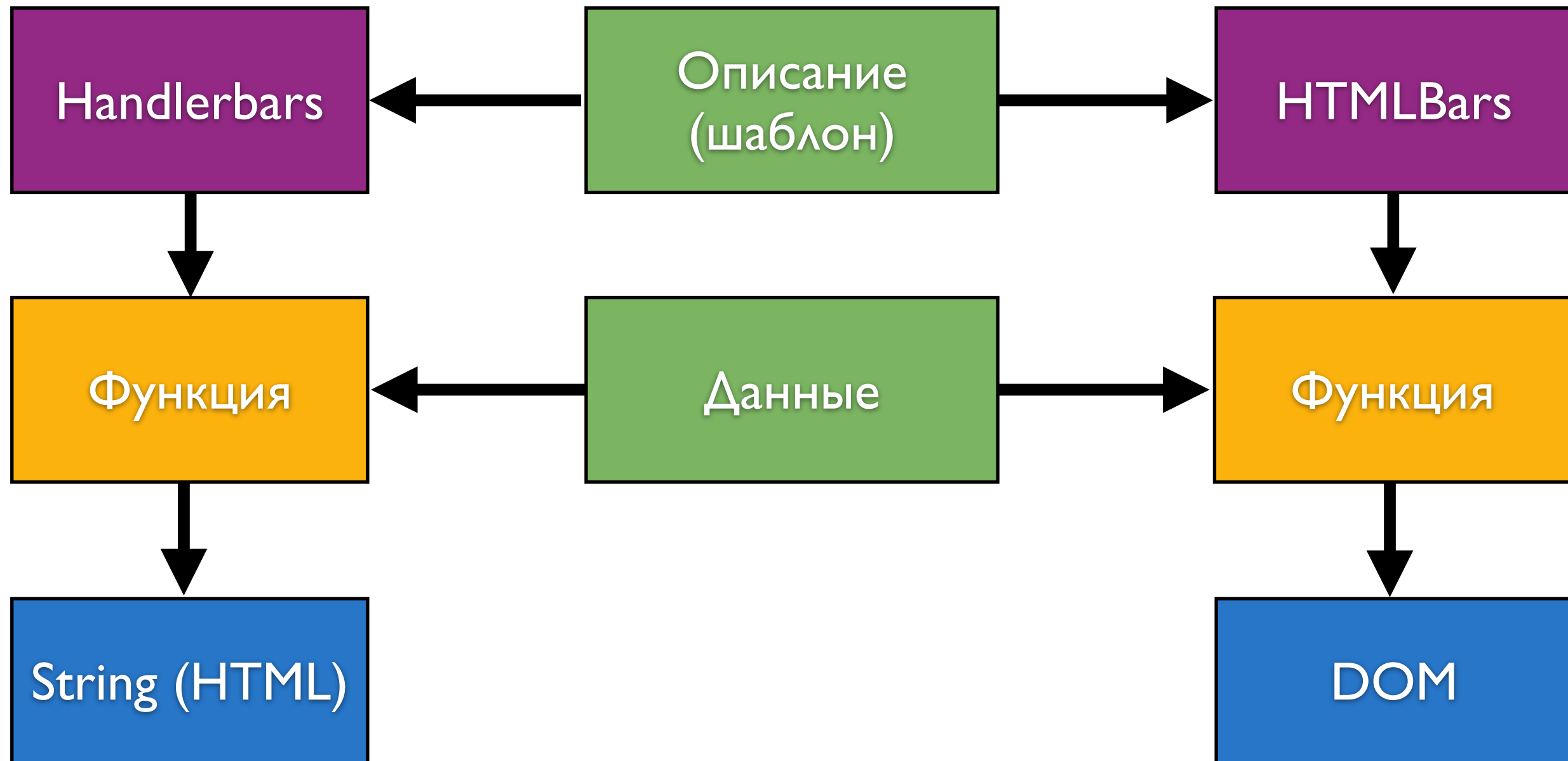
Шаблон → DOM



Это упрощенная
схема работы
шаблонизатора в **basis.js**

Разработчики **Ember** пытаются
создать что-то **подобное**

HTMLBars



Принципы работы
`basis.template` и `HTMLBars`
очень похожи

Основные отличия:
формат описания и реализация

basis.template

Production ready

HTMLBars

В разработке

HTML, DOM...

К чему все это?



Скорость!

basis-templates.js

basis.js **template** & **110n** modules
as **standalone library**

basisjs.com/templates

TodoMVC

1 000 пунктов

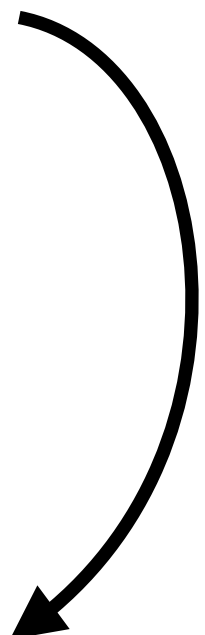
backbone.js — 510 ms

backbone.js + bbt.js — 202 ms

bbt.js — backbone basis-templates plugin

TodoMVC

	100 todo	1000 todo
AngularJS	125 ms	1491 ms
Backbone	53 ms	510 ms
Knockout	39 ms	489 ms
vanilla	23 ms	1882 ms
jQuery	20 ms	184 ms
Backbone + bbt.js	18 ms	202 ms
basis.js	8 ms	95 ms



Подробнее в докладе

Как построить DOM

tinyurl.com/build-dom

Немного не про скорость



Возможности DOM-based шаблонов — это улет!

Вот несколько примеров

- Live update – обновление DOM фрагментов без перезагрузки
- Темы

Данные

Под работой с данными
подразумевается
МОДЕЛИ, КОЛЛЕКЦИИ И Т.Д.

Один и тот же функционал
можно реализовать по-разному

Ваш К.О.

Создание моделей

10 полей, 1 listener

Решение	1 000	10 000	100 000
Basis	2 ms	22 ms	220 ms
Backbone	25 ms	248 ms	2 480 ms
Ember	25 ms	250 ms	2 500 ms

А еще...

- JSON.parse
- GC
- Создание коллекций
- Логика и расчеты
- ...

TodoMVC

	100 todo	1000 todo
Backbone.js + bbt.js	18 ms	202 ms
basis.js	8 ms	95 ms

Основная разница
из-за **медленной** работы
с данными в **Backbone**

Подробнее в докладе

**Не бойся, это всего лишь данные...
просто их много**

Чуть позже, не пропустите ;)

Как добиться **хороших**
результатов?

Простые советы

- Делать как можно меньше
- Хранить как можно проще
- Не создавать лишних структур, не клонировать
- Агрегировать события
- ...

Основной совет

Профилировать

Профилировать!

Профилировать!!!

Профилпро

Истина где-то рядом

Проблема может быть не там где
вам кажется — **разбейтесь**

Как делать быстрый код

1.Кодим

2.Профилируем

3.Думаем как оптимизировать

10 000 раз



= **быстрый код**

Или используем фреймворк,
где это уже сделано





Чем **быстрее работает** базовый код —
тем **больше полезного** можно сделать

Вопросы?

Роман Дворнов
[@rdvornov](mailto:rdvornov@gmail.com)
rdvornov@gmail.com

basis.js
basisjs.com
github.com/basisjs