

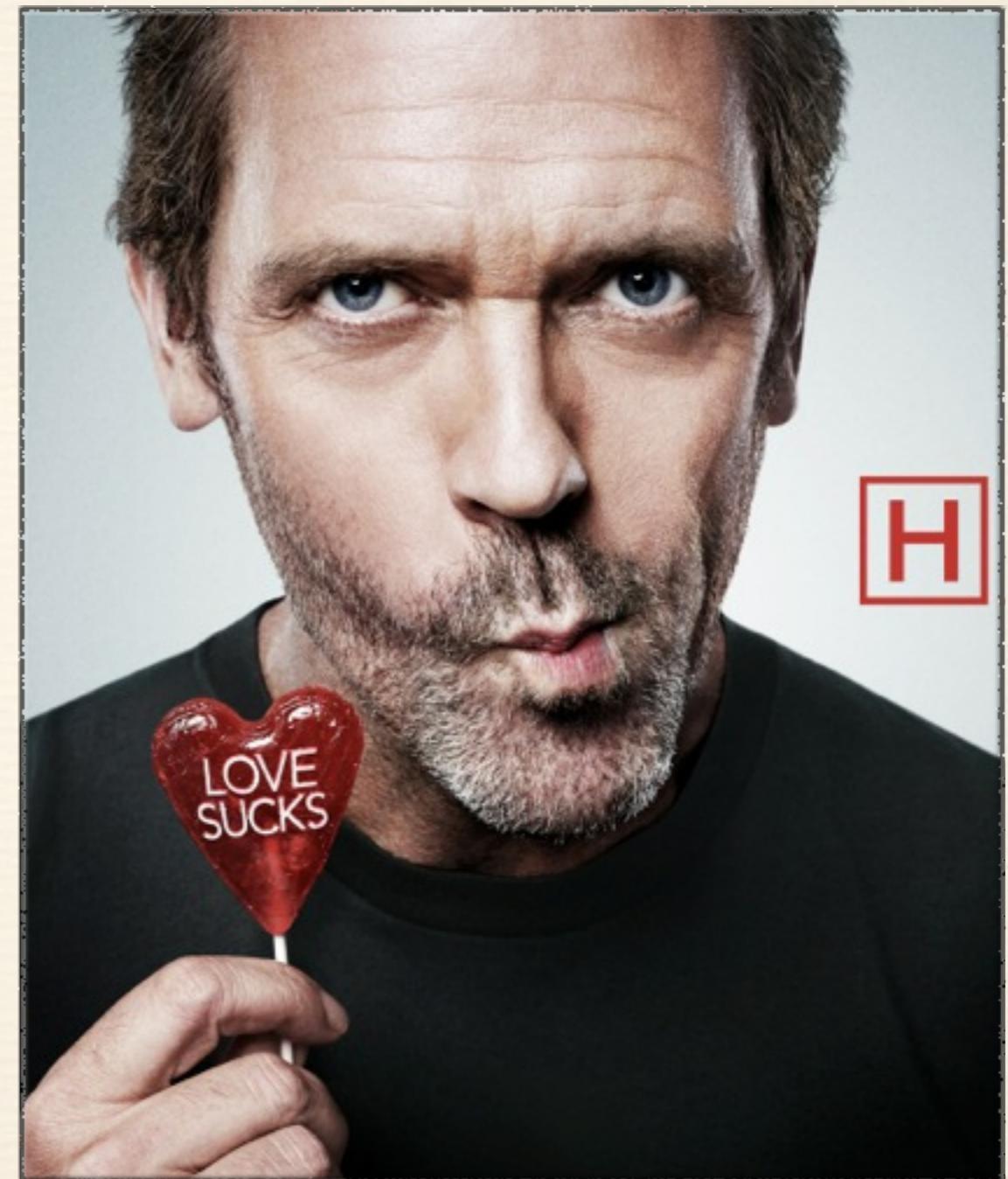
D3+MALEVICH=SPA



*маленький фреймворк для человека, но огромный
скачок для разработки SPA*

Алексей Охрименко

- ❖ Work@: Exadel
- ❖ Twitter: #Ai_boy
- ❖ <3 House M.D
- ❖ <3 ES6 - к сожалению,
пока не взаимно :\



Exadel ???

Наши клиенты

С 1998 Exadel завершил более 500 проектов.



Leveraging Open
Source



DIGITall Excellent!



The way to pay



ERICSSON



The Future of Your
Home



Web 2.0 Center of
Excellence



gordon food service

Managing the long tail for food
services



Continuous Process Improvement



Pure. Safe.
Beneficial



Hewlett
Packard



Over 75 years of Family
Winemaking



Creating a new customer experience



ABN·AMRO

www.Mortgage.com



What ever it is,
you can get it on eBay, and
Shop victoriously!



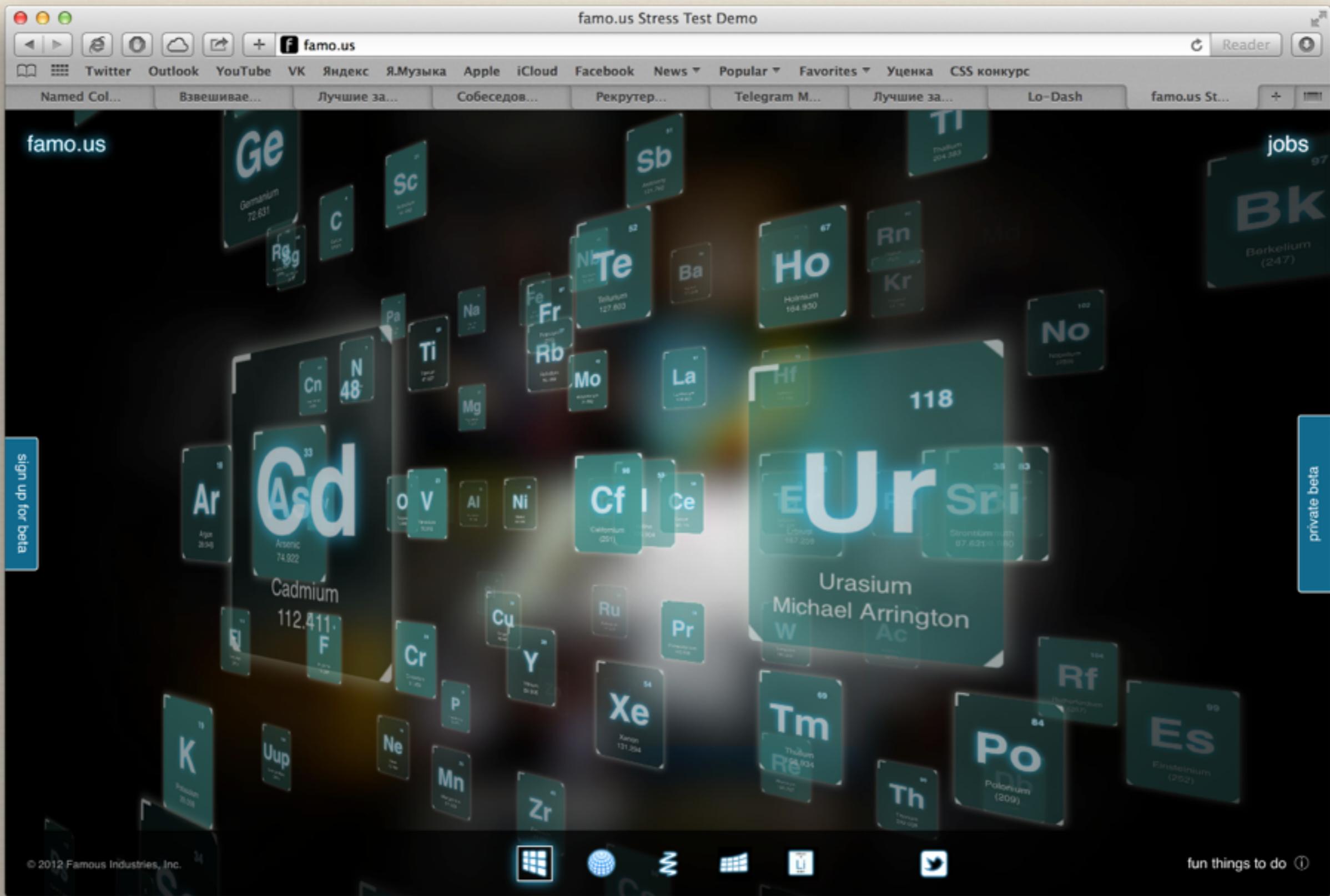
Decide with Confidence



Wolters Kluwer

Putting information at your fingertips

D3 - зачем???



D3 ???

D3 - это jQuery наоборот

Twitter Outlook YouTube VK Яндекс Я.Музыка Apple iCloud Facebook News Popular Favorites Уценка CSS конкурс

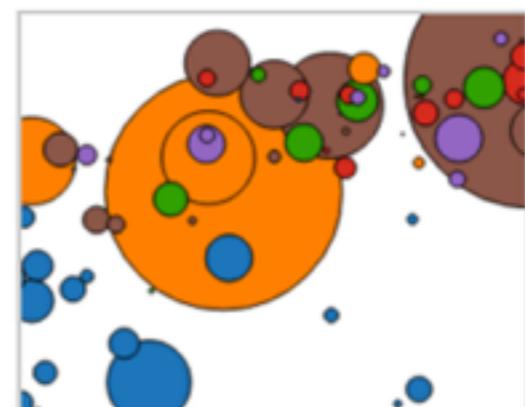
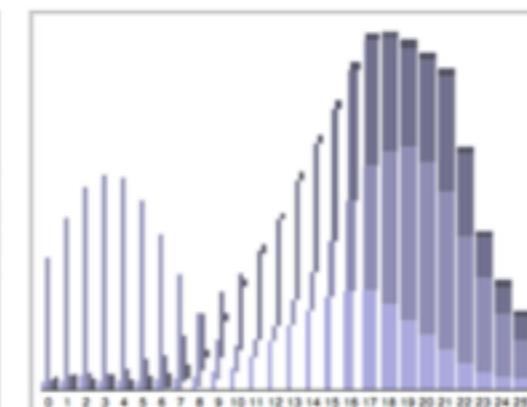
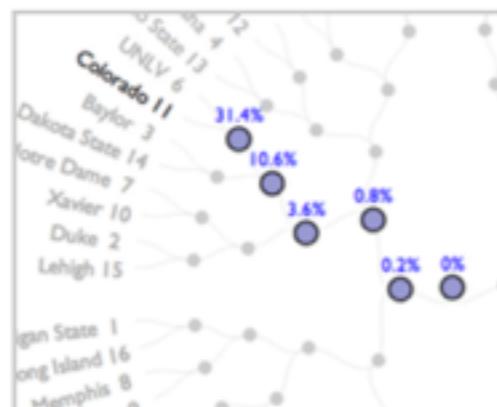
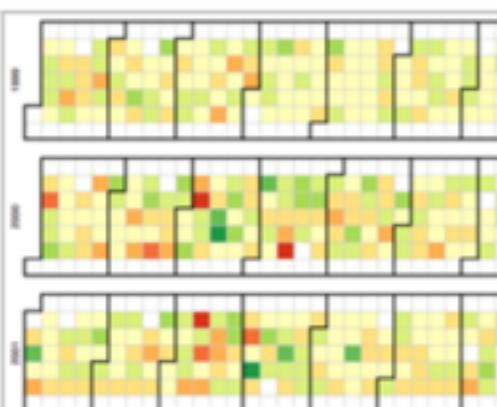
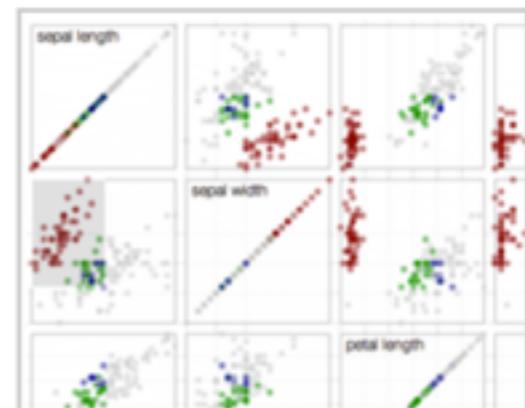
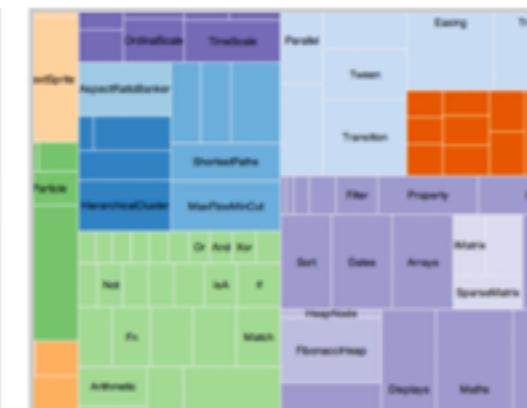
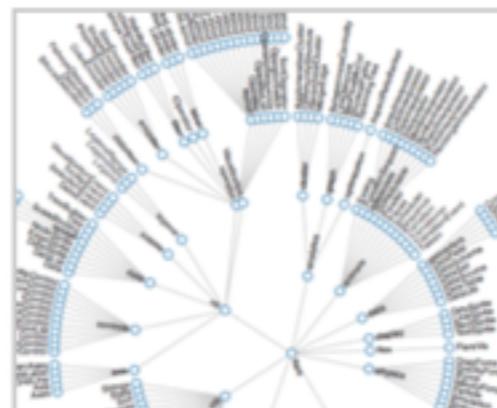
bem-js-t... An Introd... Тестовое... С.В.Савел... FrontTalk... Senior Java... Аудиозап... house md... D3.js – D...

Reader

Fork me on GitHub

[Overview](#) [Examples](#) [Documentation](#) [Source](#)

Data-Driven Documents



D3.js is a JavaScript library for manipulating documents based on data. **D3** helps you bring data to life using HTML, SVG and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

[See more examples.](#)

Download the latest version here:

- [d3.v3.zip](#)

За сколько строчек кода
можно создать вот это чудо?

Reingold–Tilford Tree

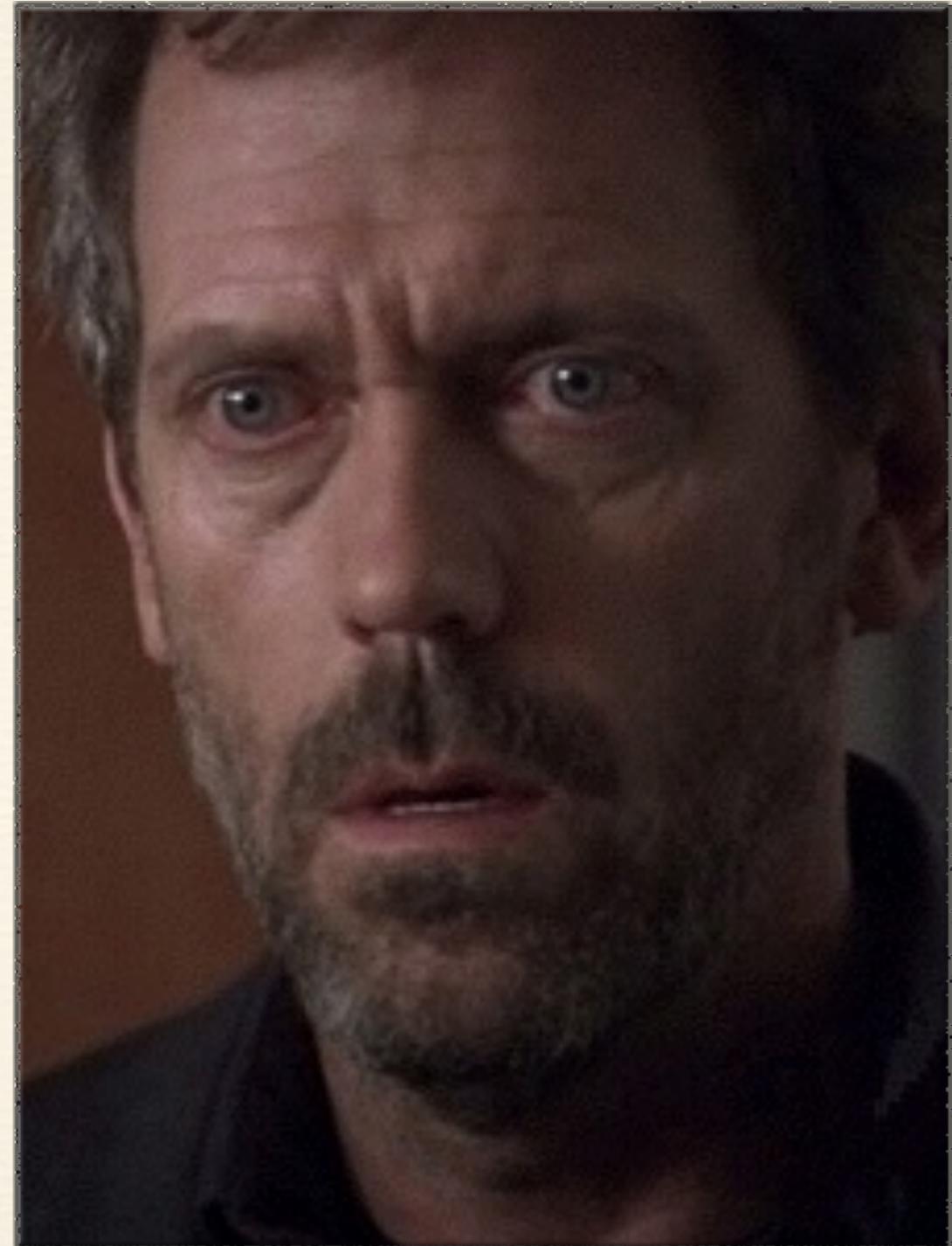
October 22, 2013



Всего 40
строчек кода!



*Моя жизнь потеряла после этого
смысл*



```
1 <script>
2
3 var diameter = 960;
4
5 var tree = d3.layout.tree()
6   .size([360, diameter / 2 - 120])
7   .separation(function(a, b) { return (a.parent == b.parent ? 1 : 2) / a.depth; });
8
9 var diagonal = d3.svg.diagonal.radial()
10  .projection(function(d) { return [d.y, d.x / 180 * Math.PI]; });
11
12 var svg = d3.select("body").append("svg")
13   .attr("width", diameter)
14   .attr("height", diameter - 150)
15   .append("g")
16   .attr("transform", "translate(" + diameter / 2 + "," + diameter / 2 + ")");
17
18 d3.json("flare.json", function(error, root) {
19   var nodes = tree.nodes(root),
20       links = tree.links(nodes);
21
22   var link = svg.selectAll(".link")
23     .data(links)
24     .enter().append("path")
25     .attr("class", "link")
26     .attr("d", diagonal);
27
28   var node = svg.selectAll(".node")
29     .data(nodes)
30     .enter().append("g")
31     .attr("class", "node")
32     .attr("transform", function(d) { return "rotate(" + (d.x - 90) + ")translate(" + d.y + ")"; })
33
34   node.append("circle")
35     .attr("r", 4.5);
36
37   node.append("text")
38     .attr("dy", ".31em")
39     .attr("text-anchor", function(d) { return d.x < 180 ? "start" : "end"; })
40     .attr("transform", function(d) { return d.x < 180 ? "translate(8)" : "rotate(180)translate(-8)"; })
41     .text(function(d) { return d.name; });
42 });
43
44 d3.select(self.frameElement).style("height", diameter - 150 + "px");
45
46 </script>
```

D3 за 3 слайда

```
1  <!-- template file -->
2
3  <div id="comments">
4      {{#each comments}}
5          <h2><a href="/posts/{{id}}">{{title}}</a></h2>
6          <div>{{body}}</div>
7      {{/each}}
8  </div>
9
10 |
```

Handlebars

<http://handlebarsjs.com>

```
14
15  var mytemplate = function () {
16    header(
17      h1('Heading'),
18      h2('Subheading'));
19
20    nav(
21      ul({ 'class': 'breadcrumbs' },
22          li(a({ href: '/' }, 'Home')),
23          li(a({ href: '/section/' }, 'Section')),
24          li(a('Subject'))));
25
26    article(
27      p('Lorem ipsum...'));
28
29    footer('Footer stuff');
30  };
31
32
```

domjs

<https://github.com/medikoo/domjs>

```
1 var circle = svg.selectAll("circle")
2   .data([32, 57, 293], String);
3
4 ▼ circle.enter().append("circle")
5   .attr("cy", 90)
6   .attr("cx", String)
7   .attr("r", Math.sqrt);
8
9 circle.exit().remove();
```

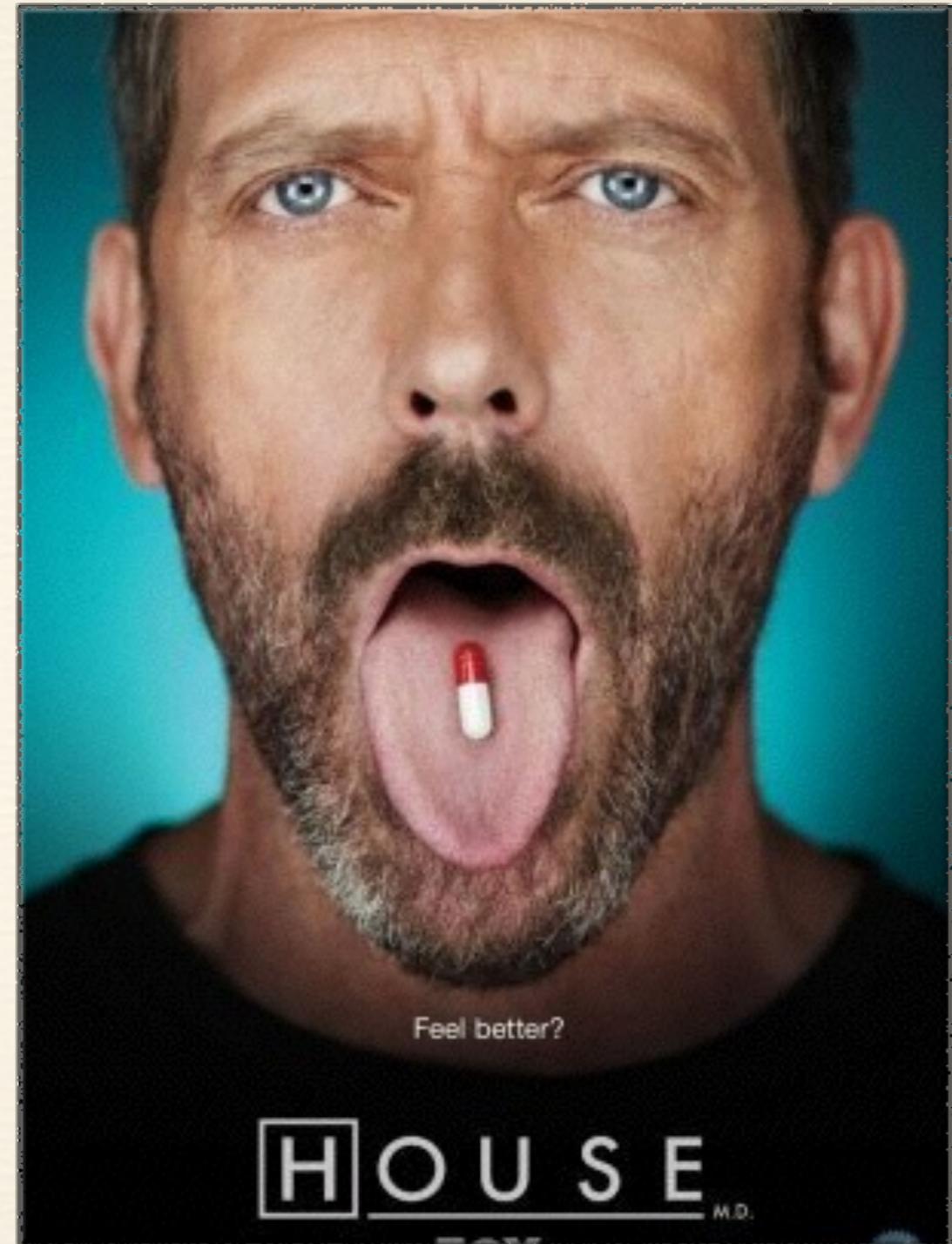
D3

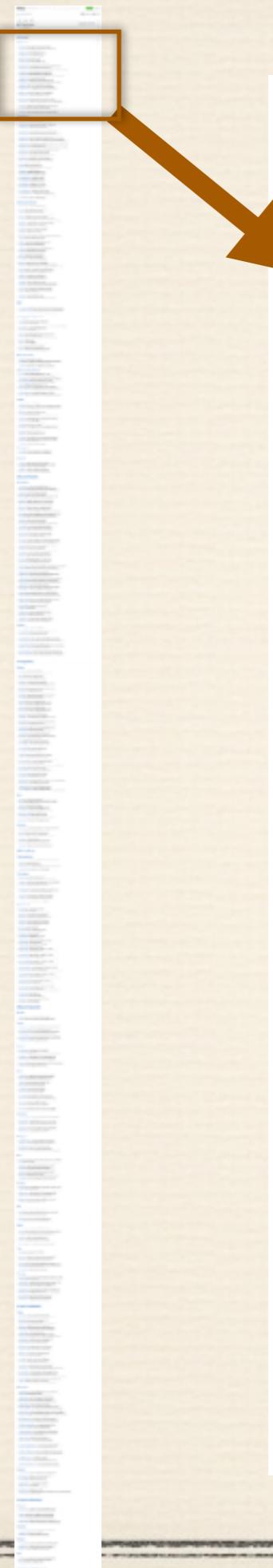
<http://d3js.org>

D3 - это
шаблонизатор
на стероидах



*я б даже сказал шаблонизатор на
викодин*





d3 (core)

Selections

- [d3.select](#) - select an element from the current document.
- [d3.selectAll](#) - select multiple elements from the current document.
- [selection.attr](#) - get or set attribute values.
- [selection.classed](#) - add or remove CSS classes.
- [selection.style](#) - get or set style properties.
- [selection.property](#) - get or set raw properties.
- [selection.text](#) - get or set text content.
- [selection.html](#) - get or set inner HTML content.
- [selection.append](#) - create and append new elements.
- [selection.insert](#) - create and insert new elements before existing elements.
- [selection.remove](#) - remove elements from the document.
- [selection.data](#) - get or set data for a group of elements, while computing a relational join.
- [selection.enter](#) - returns placeholders for missing elements.
- [selection.exit](#) - returns elements that are no longer needed.
- [selection.datum](#) - get or set data for individual elements, without computing a join.
- [selection.filter](#) - filter a selection based on data.
- [selection.sort](#) - sort elements in the document based on data.
- [selection.order](#) - reorders elements in the document to match the selection.
- [selection.on](#) - add or remove event listeners for interaction.
- [selection.transition](#) - start a transition on the selected elements.
- [selection.interrupt](#) - immediately interrupt the current transition, if any.
- [selection.each](#) - call a function for each selected element.
- [selection.call](#) - call a function passing in the current selection

Чтобы по-настоящему
понять D3 давайте
поиграем в режиссеров



*Вся наша жизнь театр, а люди
в нем свиньи!*



- 1) История
- 2) Сцена
- 3) Актеры и их реплики
- 4) Определенный сценарий
- 5) Много репетиций

1. Нам нужна классная история
(ориентированность на данные)

```
[  
  { name: 'Alex', value: 3 },  
  { name: 'Mary', value: 7 }  
]
```

Данные уже должны
существовать изначально

2. Концепция Сцены

Нам необходимо выбрать
сцену

```
var stage =  
d3.select('theater')
```

Но так как бюджет у нас ограничен
давайте выберем сцену попроще :)

```
var stage =  
d3.select(document.body)
```

d3.select(...)

==
==

document.querySelector(...)

==
==

\$(document.body)

Теперь нужно определиться с тем,
какие актеры у нас будут
выступать :)

```
var actors =  
d3.select(document.body)  
    .selectAll('div')
```

3. А теперь давайте каждому Актеру
назначим свой «кусочек» истории
(каждому выдадим по реплике)

```
var array = [
  { name: 'Alex', value: 3 },
  { name: 'Mary', value: 7 }
]
```

```
var actorsWithScript =
d3.select(document.body)
    .selectAll('div')
    .data(array)
```

4. Теперь нам нужен четкий сценарий

a) На сцене **остается**:

Вася

b) На сцену **выходят**:

Маша и Даша

c) Со сцены **уходит**:

Коля

a) На сцене **остается**:

Вася

```
actorsWithScript  
.style('color', 'red');
```

b) На сцену **выходят**:

Маша и Даша

```
actorsWithScript  
.enter()  
.append('div');
```

c) Со сцены **уходит**:

Коля

```
actorsWithScript  
.exit()  
.remove()
```

```
var actorsWithScript =
  d3.select(document.body)
    .selectAll('div')
    .data(array);

// для тех, кто уже на сцене
actorsWithScript
  .style('color', 'red');

// для тех, кто выходит на сцену
actorsWithScript
  .enter()
  .append('div');

// для тех, кто уходит со сцены
actorsWithScript
  .exit()
  .remove();
```

5. А теперь нужно
порепетировать

```
function playStage( array ) {  
    // весь предыдущий код  
    // ...  
}
```

```
function playStage( array ) {  
    var actorsWithScript =  
        d3.select(document.body)  
            .selectAll('div')  
            .data(array)  
  
    // для тех, кто уже на сцене  
    actorsWithScript  
        .style('color', 'red');  
  
    // для тех, кто выходит на сцену  
    actorsWithScript  
        .enter()  
        .append('div');  
  
    // для тех, кто уходит со сцены  
    actorsWithScript  
        .exit()  
        .remove();  
}
```

```
playStage([
    { name: 'Alex', value: 3 },
    { name: 'Mary', value: 7 }
])
```

JavaScript ▾

```
function playStage( array ) {
    var actorsWithScript = d3
        .select(document.body)
        .selectAll('div')
        .data(array);

    // для тех, кто уже на сцене
    actorsWithScript
        .style('color', 'red');

    // для тех, кто выходит на сцену
    actorsWithScript
        .enter()
        .append('div');

    // для тех, кто уходит со сцены
    actorsWithScript
        .exit()
        .remove();
}

playStage([
    { name: 'Alex', value: 3 },
    { name: 'Mary', value: 7 }
]);
```

Output

```
<iframe sandbox="allow-forms allow-pointer-lock allow-popups allow-same-  
scripts" frameborder="0" id="1385117340433">  
  ▼#document  
    <!DOCTYPE html>  
    ▼<html>  
      ▶<head>...</head>  
      ▼<body>  
        ▶<script>...</script>  
        <div></div>  
        <div></div>  
      </body>  
    </html>  
  </iframe>  
/div>  
script src="http://static.jsbin.com/js/prod/runner-3.8.9.min.js"></script>  
body>  
nl>  
>  
ss="label">...</div>  
  
lswaiting"></div>  
eform" method="post" action="http://jsbin.com/AfoBuR0W/1/save">...</form>
```

Давайте внимательно вчитаемся
в то, что мы попросили сделать
наших актеров

```
// для тех, кто выходит на сцену
actorsWithScript
    .enter()
    .append('div');
```

Какие-то безвольные
актеры у нас :)

реплика

{ name: 'Alex', value: 3 },

```
// для тех, кто выходит на сцену
actorsWithScript
  .enter()
  .append('div')
  .text(function(d) { return d.name + d.value; });
```

JavaScript ▾

```
function playStage( array ) {
    var actorsWithScript = d3
        .select(document.body)
        .selectAll('div')
        .data(array);

    // для тех, кто уже на сцене
    actorsWithScript
        .style('color', 'red');

    // для тех, кто выходит на сцену
    actorsWithScript
        .enter()
        .append('div')
        .text(function(d) {
            return d.name + d.value;
        });

    // для тех, кто уходит со сцены
    actorsWithScript
        .exit()
        .remove();
}

playStage([
    { name: 'Alex', value: 3 },
    { name: 'Mary', value: 7 }
]);
```

Output

Alex3
Mary7

```
playStage([
    { name: 'Alex', value: 3 },
    { name: 'Mary', value: 7 }
])
playStage([
    { name: 'Mary', value: 9 },
    { name: 'Kolya', value: 6 },
    { name: 'Tolya', value: 1 }
])
```

JavaScript ▾

```
function playStage( array ) {
  var actorsWithScript = d3
    .select(document.body)
    .selectAll('div')
    .data(array);

  // для тех, кто уже на сцене
  actorsWithScript
    .style('color', 'red');

  // для тех, кто выходит на сцену
  actorsWithScript
    .enter()
    .append('div')
    .text(function(d) {
      return d.name + d.value;
    });

  // для тех, кто уходит со сцены
  actorsWithScript
    .exit()
    .remove();
}

playStage([
  { name: 'Alex', value: 3 },
  { name: 'Mary', value: 7 }
]);
playStage([
  { name: 'Mary', value: 9 },
  { name: 'Kolya', value: 6 },
  { name: 'Tolya', value: 1 }
]);
```

Output

Alex3

Mary7

Tolya1

По-умолчанию Актеры
очень наглье

```
playStage([
  { name: 'Alex', value: 3 },
  { name: 'Mary', value: 7 }])
```

```
playStage([
  { name: 'Mary', value: 9 },
  { name: 'Kolya', value: 6 },
  { name: 'Tolya', value: 1 }])
```

(каждый объект - это новая речь)

Output

Alex3

Mary7

Tolya1

Как проверить то что Актеры
у нас действительно наглые?

```
// для тех, кто уже на сцене
actorsWithScript
    .style('color', 'red')
    .text(function(d) { return d.name; });
```

JavaScript ▾

```
function playStage( array ) {
    var actorsWithScript = d3
        .select(document.body)
        .selectAll('div')
        .data(array);

    // для тех, кто уже на сцене
    actorsWithScript
        .style('color', 'red')
        .text(function(d) { return d.name; });

    // для тех, кто выходит на сцену
    actorsWithScript
        .enter()
        .append('div')
        .text(function(d) {
            return d.name + d.value;
        });

    // для тех, кто уходит со сцены
    actorsWithScript
        .exit()
        .remove();
}

playStage([
    { name: 'Alex', value: 3 },
    { name: 'Mary', value: 7 }
]);
playStage([
    { name: 'Mary', value: 9 },
    { name: 'Kolya', value: 6 },
    { name: 'Tolya', value: 1 }
]);

```

Output

Mary
Kolya
Tolya1

Как сделать так, чтобы Актеры
все-таки уходили со сцены

Кадому актеру нужно указать свой Index

```
var actorsWithScript =  
    d3.select(document.body)  
    .selectAll('div')  
    .data(array, function(d) { return d.name; };)
```

```
function playStage( array ) {
  var actorsWithScript = d3
    .select(document.body)
    .selectAll('div')
    .data(array, function(d) { return
d.name; });

  // для тех, кто уже на сцене
  actorsWithScript
    .style('color', 'red')
    .text(function(d) { return d.name; });

  // для тех, кто выходит на сцену
  actorsWithScript
    .enter()
    .append('div')
    .text(function(d) {
      return d.name + d.value;
    });

  // для тех, кто уходит со сцены
  actorsWithScript
    .exit()
    .remove();
}

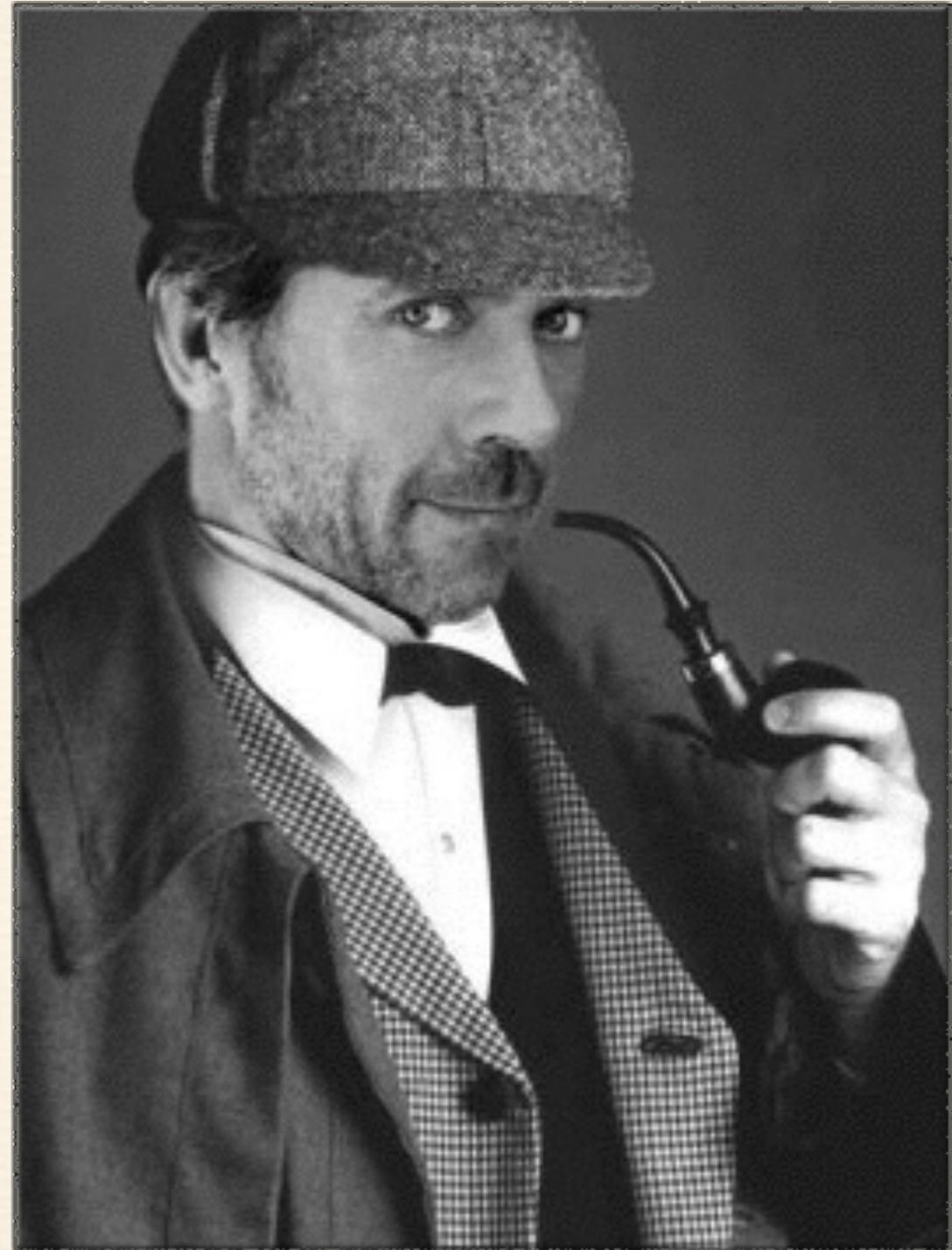
playStage([
  { name: 'Alex', value: 3 },
  { name: 'Mary', value: 7 }
]);
playStage([
  { name: 'Mary', value: 9 },
  { name: 'Kolya', value: 6 },
  { name: 'Tolya', value: 1 }
]);
```

Mary
Kolya6
Tolya1

Все это
элементарно,
если помнить:



- 1) *D3 - это jQuery наоборот*
- 2) *D3 - это шаблонизатор на стереоидах*
- 3) *D3 - имеет концепцию «сцены»*



MALEVICH



новый паттерн программирования



This repository

Search or type a command



Explore Gist Blog Help

aiboy



aiboy / MALEVICH

[Unwatch](#) 1[Unstar](#) 1[Fork](#) 0

MaLEVICH - ModeL viEwmodel View Controller - new software architecture pattern for Web — Edit

1 commit

1 branch

0 releases

1 contributor



branch: master

MALEVICH /

Initial commit

aiboy authored 3 months ago

latest commit `67e22a686e`

LICENSE

Initial commit

3 months ago

README.md

Initial commit

3 months ago

README.md

MALEVICH

MaLEVICH - ModeL viEwmodel View Controller - new software architecture pattern for Web

Code

Issues

0

Pull Requests

0

Wiki

Pulse

Graphs

Network

Settings

HTTPS clone URL`https://github.com/` You can clone with [HTTPS](#), [SSH](#),
or [Subversion](#).

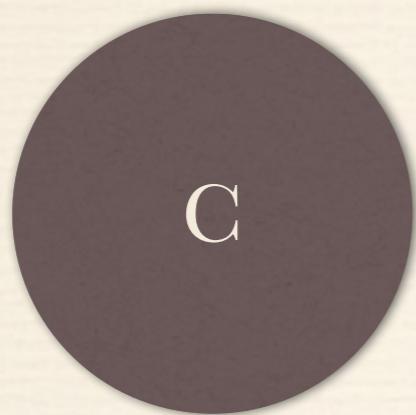
Clone in Desktop

Download ZIP

MALEVICH - это смесь
MVC и MVVM



MVC



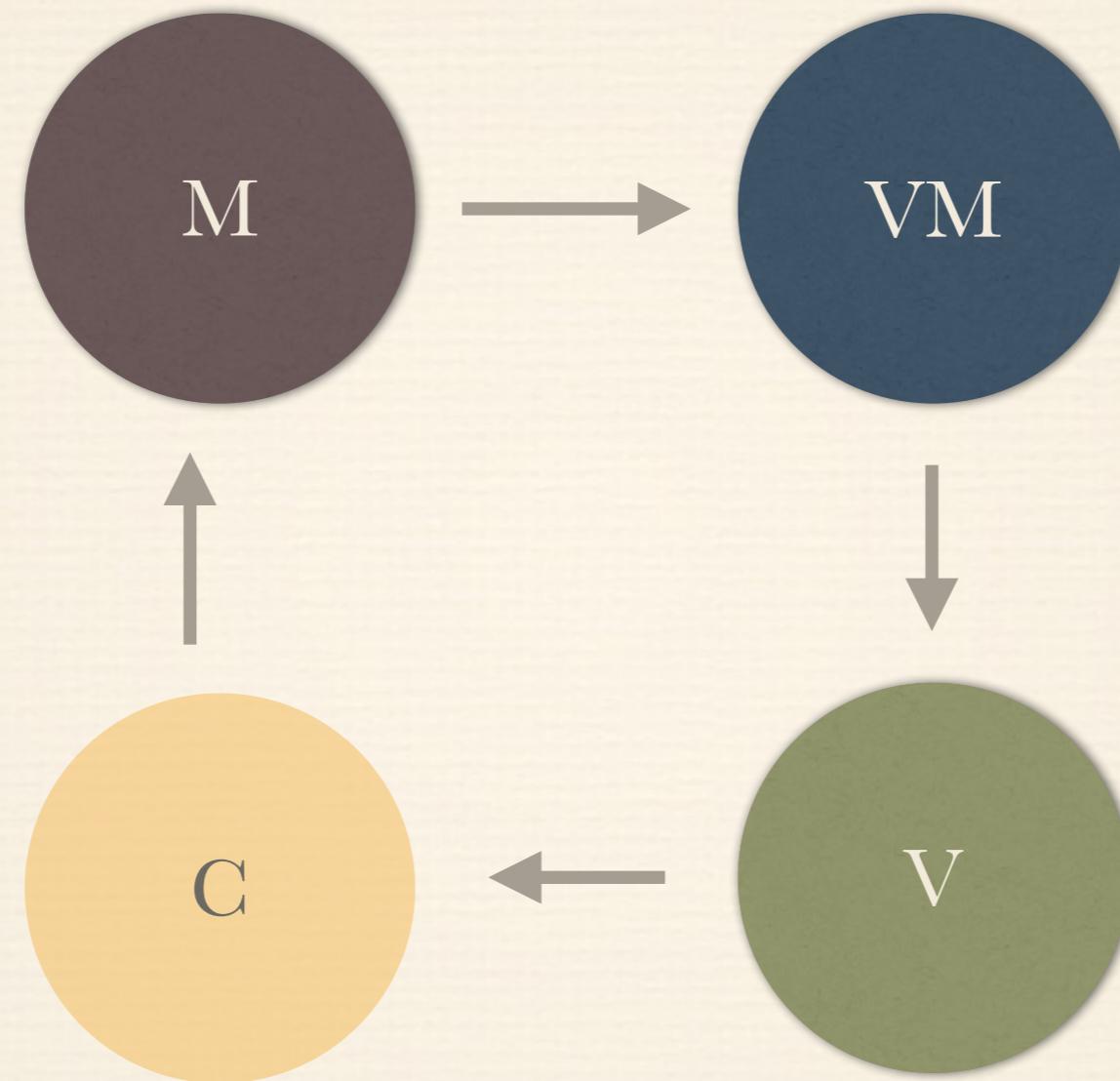
MVVM



MALEVICH

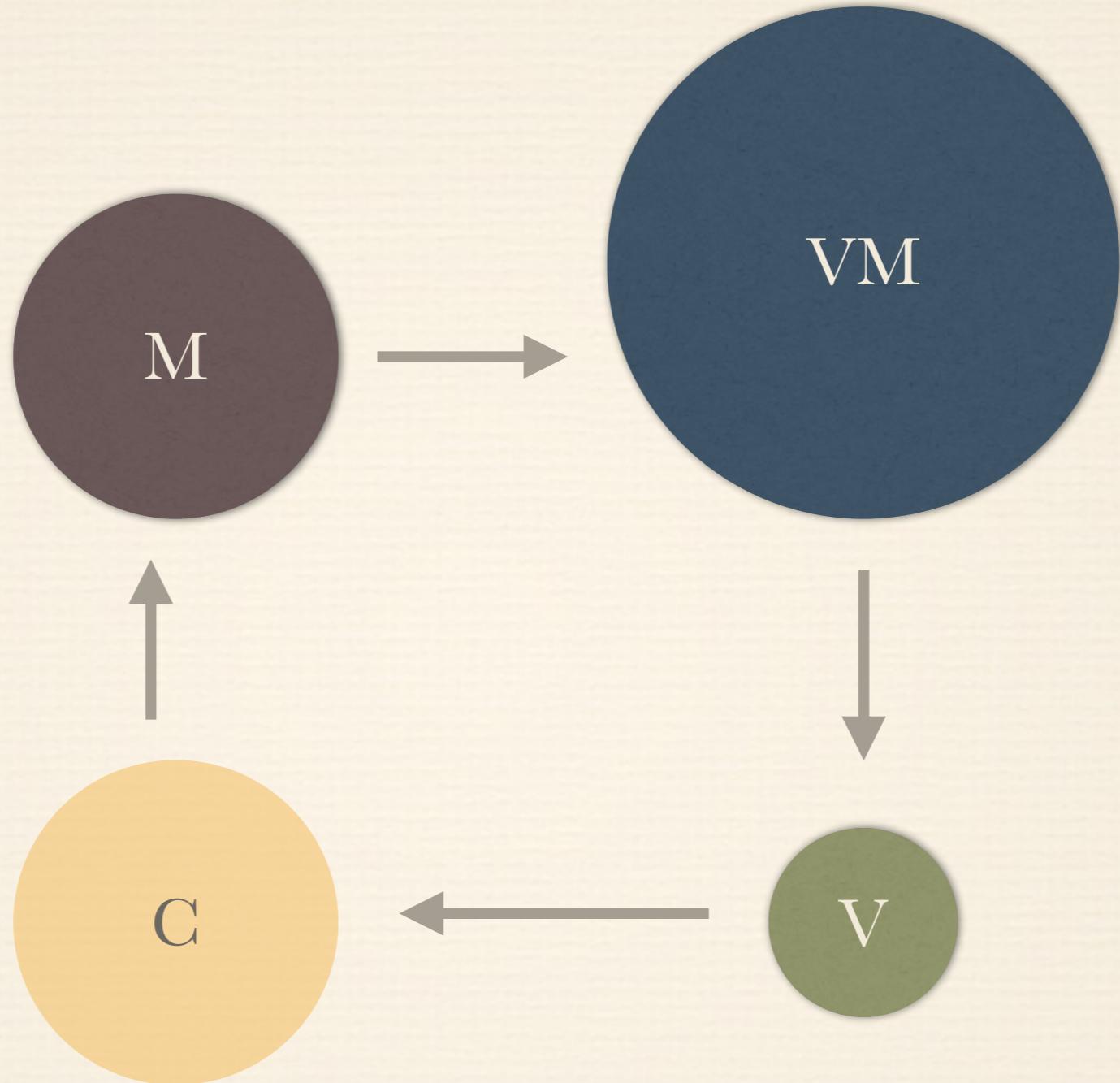
кэш, валидация,
синхронизация

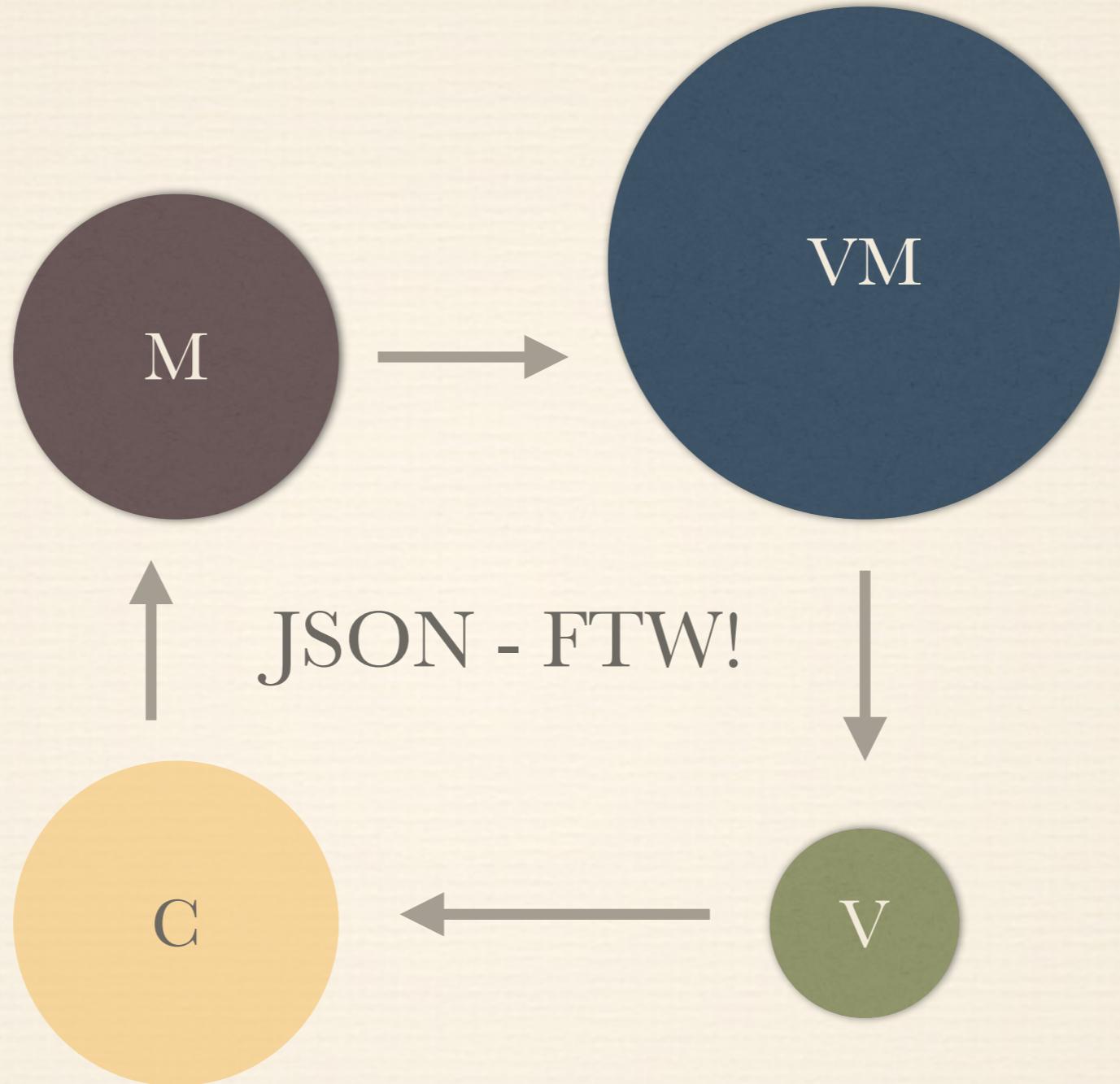
бизнес-логика
отображение данных



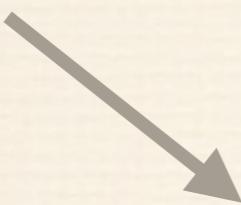
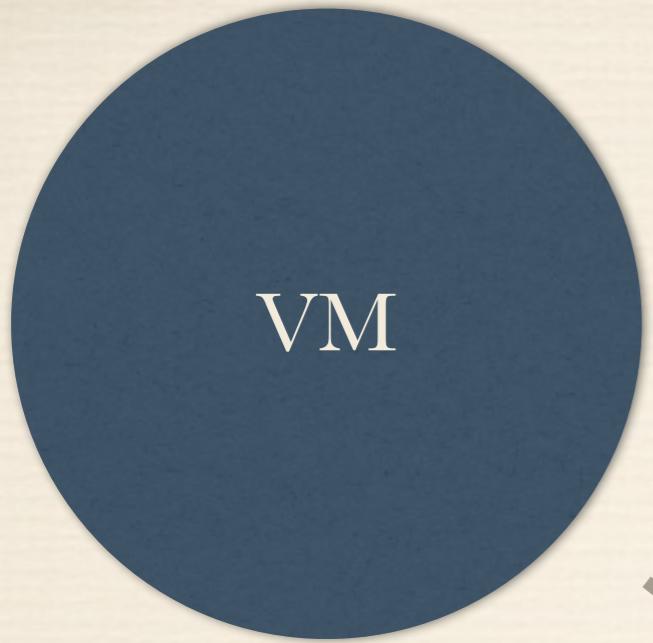
бизнес-логика
реакция на Юзверя

тупое отображение





- 1) вызов метода с передачей JSON
- 2) PubSub паттерн - с JSON совместимыми аргументами



[

```
{ type: 'button', value: 'Done', w: 100 },  
{ type: 'button', value: 'Cancel', w: 200 },
```

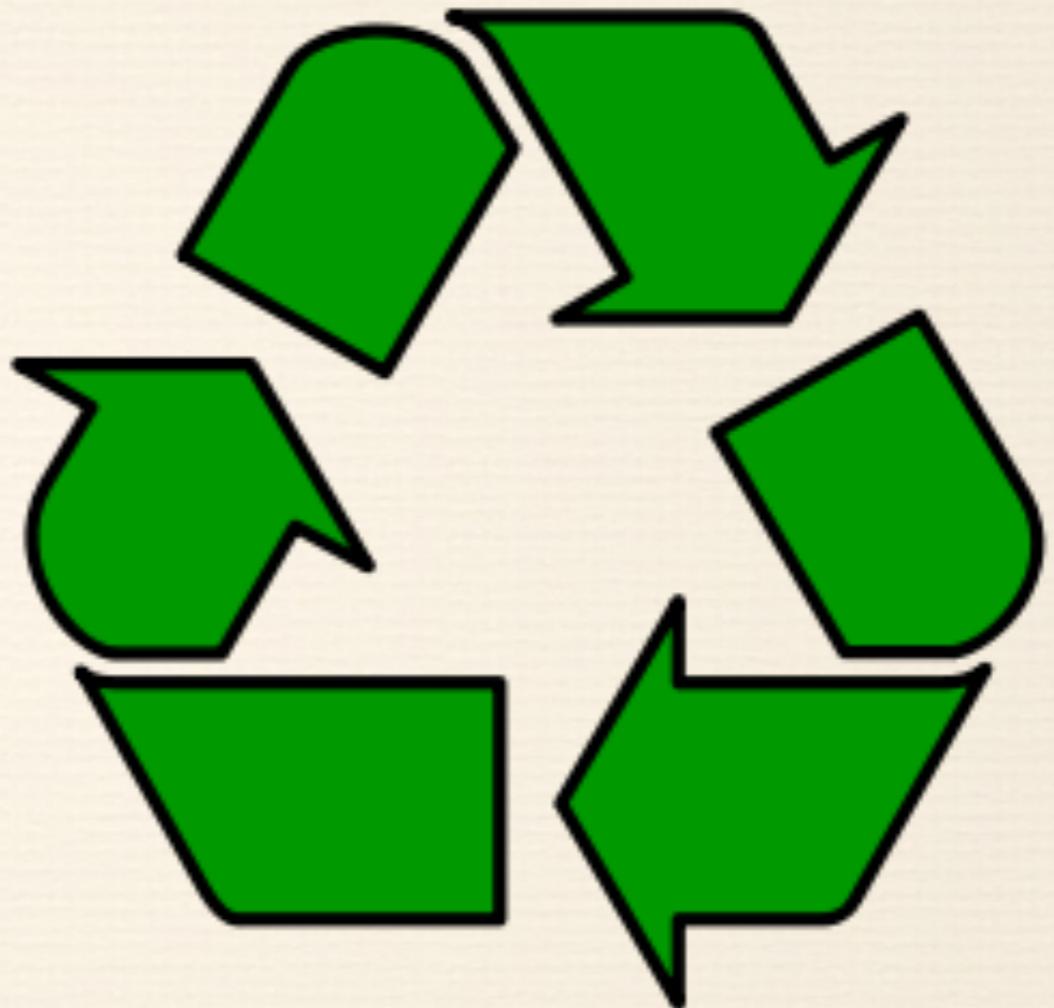
]

А теперь давайте вспомним про
жадное поведение Актеров :)

DOM recycling!



*придерживаемся магического числа
1000 DOM элементов*



A close-up, black and white promotional photograph of Dr. House. He has intense blue eyes and a serious expression. His mouth is sealed shut by two pieces of yellowish-brown medical tape, which are crossed over each other. He is wearing a dark suit jacket over a light-colored shirt.

EVERYBODY LIES.

HOUSE

К сожалению в WebWorker нельзя работать с
DOM
(по крайней мере нам так говорит спецификация)



This repository

Search or type a command



Explore Gist Blog Help

aiboy



aiboy / jsdom4browser

Unwatch 1

Unstar 1

Fork 1

A javascript implementation of the DOM, for use with WebWorkers and browsers — Edit

2 commits

1 branch

0 releases

1 contributor



branch: master

jsdom4browser /

jsdom4browsers - initial commit

Oleksii Okhrymenko authored 18 days ago

latest commit 88a49d278e

example-d3

jsdom4browsers - initial commit

18 days ago

LICENSE

Initial commit

18 days ago

README.md

jsdom4browsers - initial commit

18 days ago

README.md

jsdom4browser

A javascript implementation of the DOM, for use with WebWorkers and browsers

Warning

Code

Issues 0

Pull Requests 0

Wiki

Pulse

Graphs

Network

Settings

HTTPS clone URL You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Clone in Desktop

Download ZIP

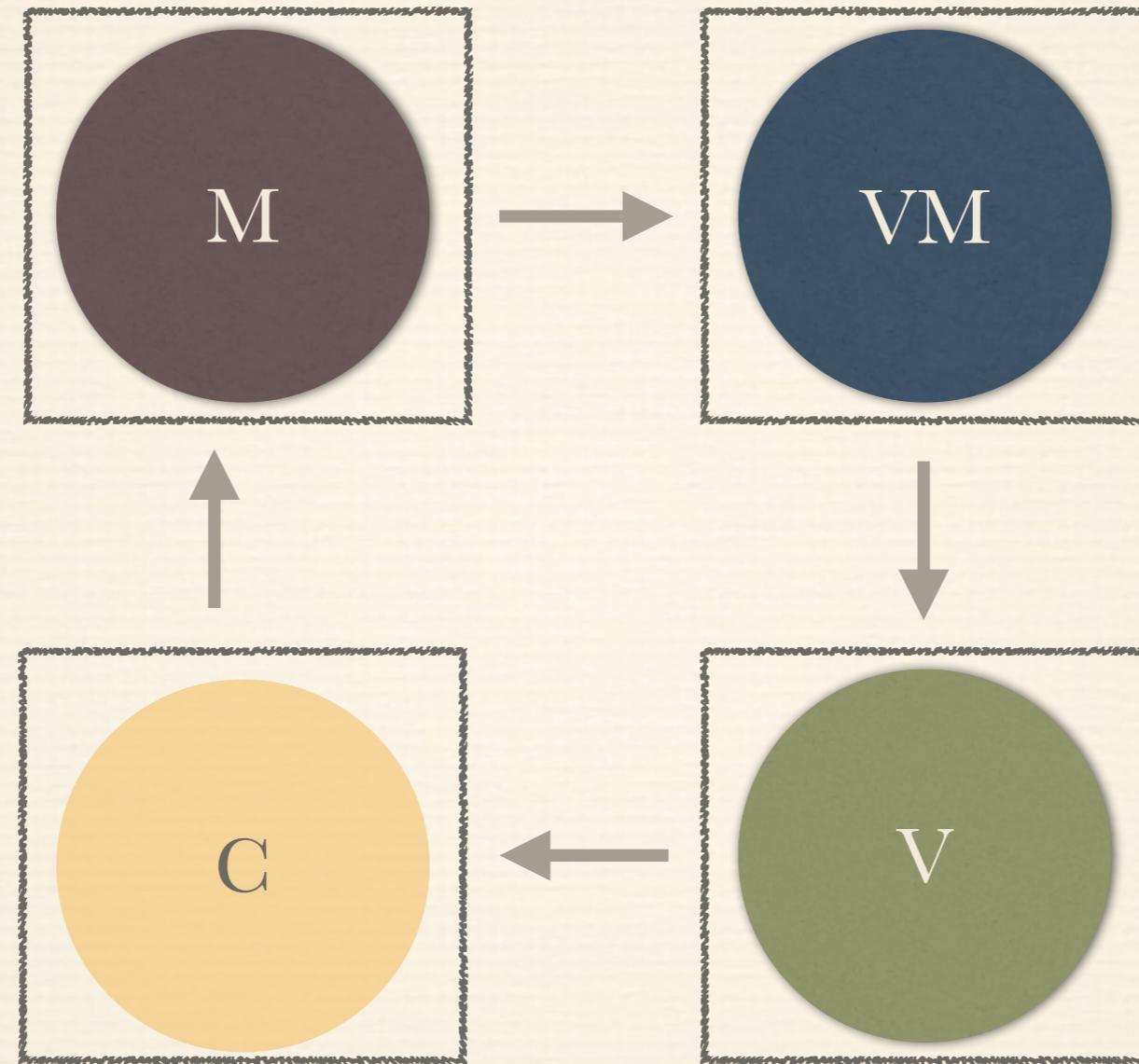
```
d3.select(window.document.body)
  .selectAll('div')
  .data([1, 2, 3])
  .enter()
  .append('div')
  .classed('love', true)
  .text(String);
```

```
postMessage(window.document.body.innerHTML);
```

К сожалению jsdom это лишь эмуляция API.

`offsetWidth === 0`

WebWorkers - FTW!



Плюсы MALEVICH

- ❖ DOM recycling!
- ❖ Неимоверная скорость и отзывчивость UI
(даже на мобильных девайсах)
- ❖ Безграничный простор для диких оптимизаций
- ❖ Почти бесплатный Drag&Drop

Минусы MALEVICH

- ❖ Семантика теряется по дороге (No SEO)
- ❖ Это паттерн - это не фреймворк
(многое приходиться делать с нуля)
- ❖ Осутствие какой либо документации на
данный момент

Мой совет



H SERIES
FINALE
EVENT
MON MAY 21
8/7c FOX

Алексей Охрименко

Все ссылки с доклада вы сможете найти
у меня в

Twitter: @Ai_boy