

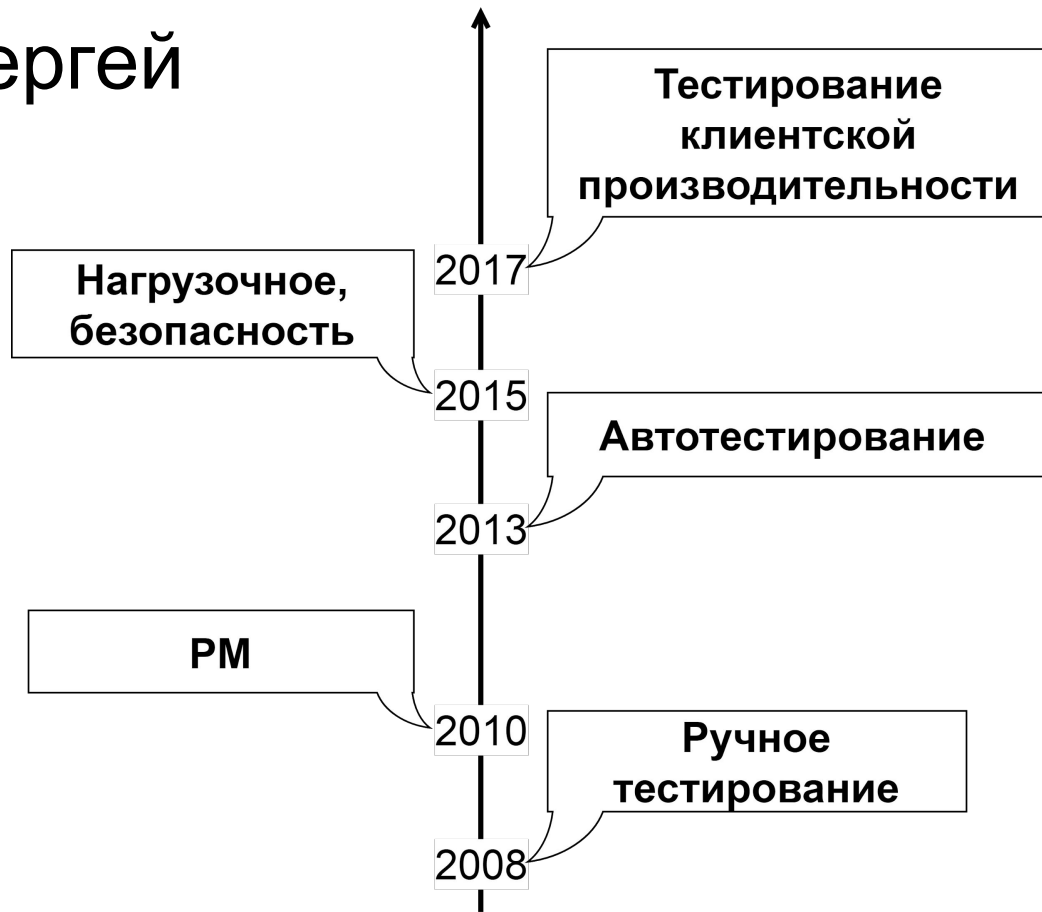
Как мы написали фреймворк
Sealant для поиска
утечек памяти
в JS

Докучаев Сергей
Тензор

Докучаев Сергей

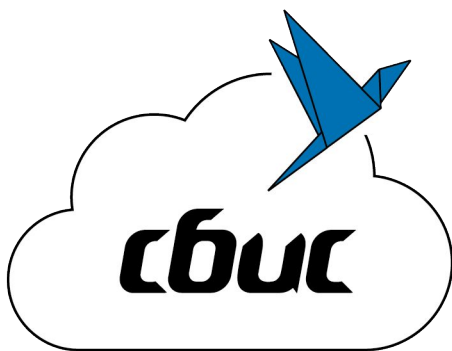
Тензор

t.me/m0rscan



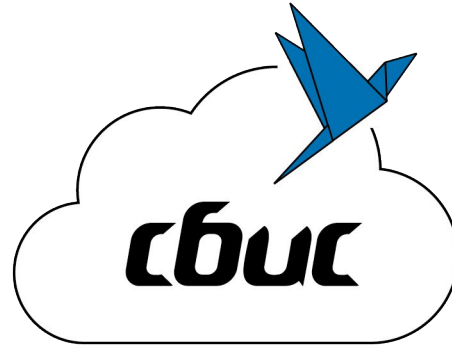
О чём будем говорить

- > Менеджмент памяти в JS
- > Инструменты
- > Автоматизация и фреймворк Sealant
- > Покрытие





Отчетность

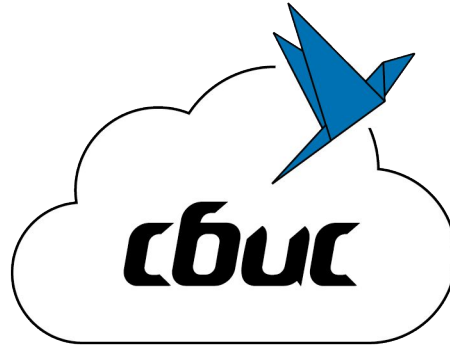




Отчетность



Электронный
документооборот





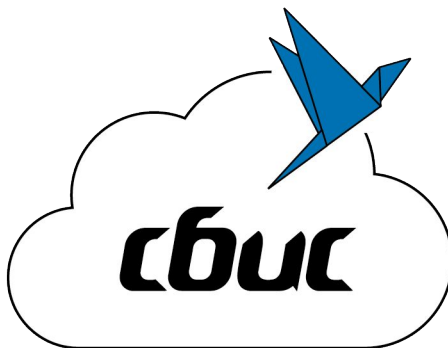
Бухгалтерия
и отчетность



Электронный
документооборот



Персонал и учет
рабочего времени



CRM и управление
call-центром



Задачи и
проекты



Бухгалтерия
и отчетность



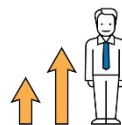
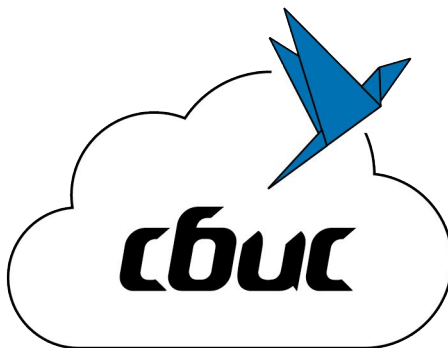
Электронный
документооборот



Заказы и
поставки (EDI)



Персонал и учет
рабочего времени



CRM и управление
call-центром



Все о компаниях,
закупках и торгах



Управление
бизнес-процессами



Задачи и
проекты



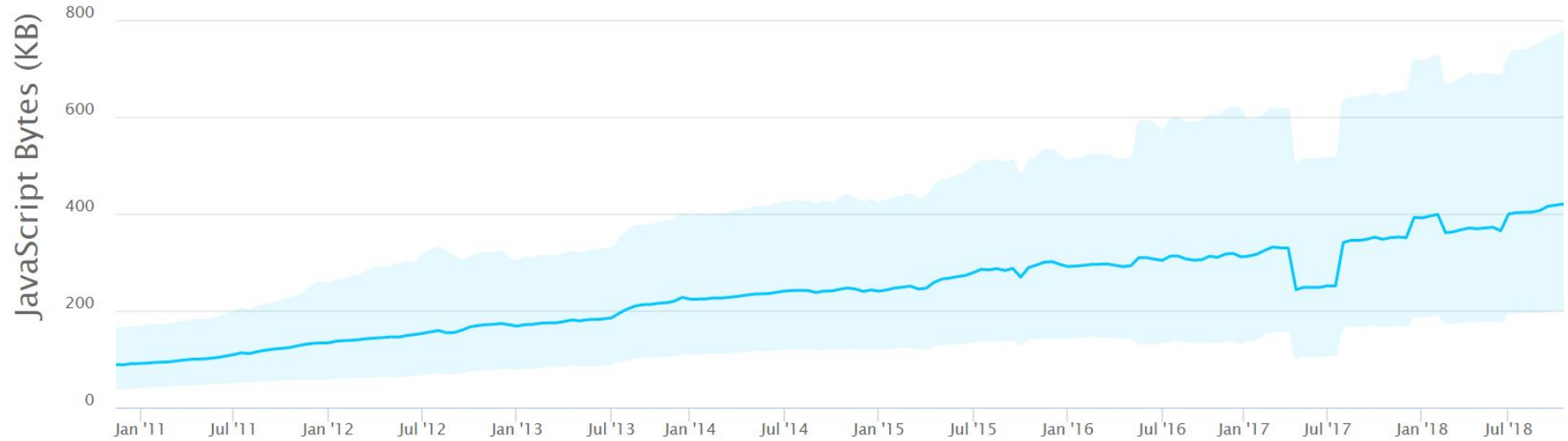
ОФД, автоматизация
розницы и общепита



Коммуникации и
корпоративная соцсеть



httparchive.org



88 KB B 2011 -> **420** KB B 2018

Steam Hardware & Software Survey: October 2018

< 4 ГБ памяти у ~20% пользователей



ОФД И ОНЛАЙН-КАССЫ

Оператор фискальных данных, анализ продаж, мониторинг торговых точек, поставка, настройка и обслуживание касс

[Подробнее →](#)

Начать работу





Тарифы | Контакты | Поддержка

ОФД И ОНЛАЙН-КАССЫ

Оператор фискальных данных, анализ продаж, мониторинг торговых точек, поставка, настройка и обслуживание касс

[Подробнее →](#)

Начать работу





СЕТЬ ДЕЛОВЫХ КОММУНИКАЦИЙ

Объединяет людей и компании, системы управления и учета, бизнес-процессы и документы в одной системе



Начать работу



Тарифы | Контакты | Поддержка | ☰

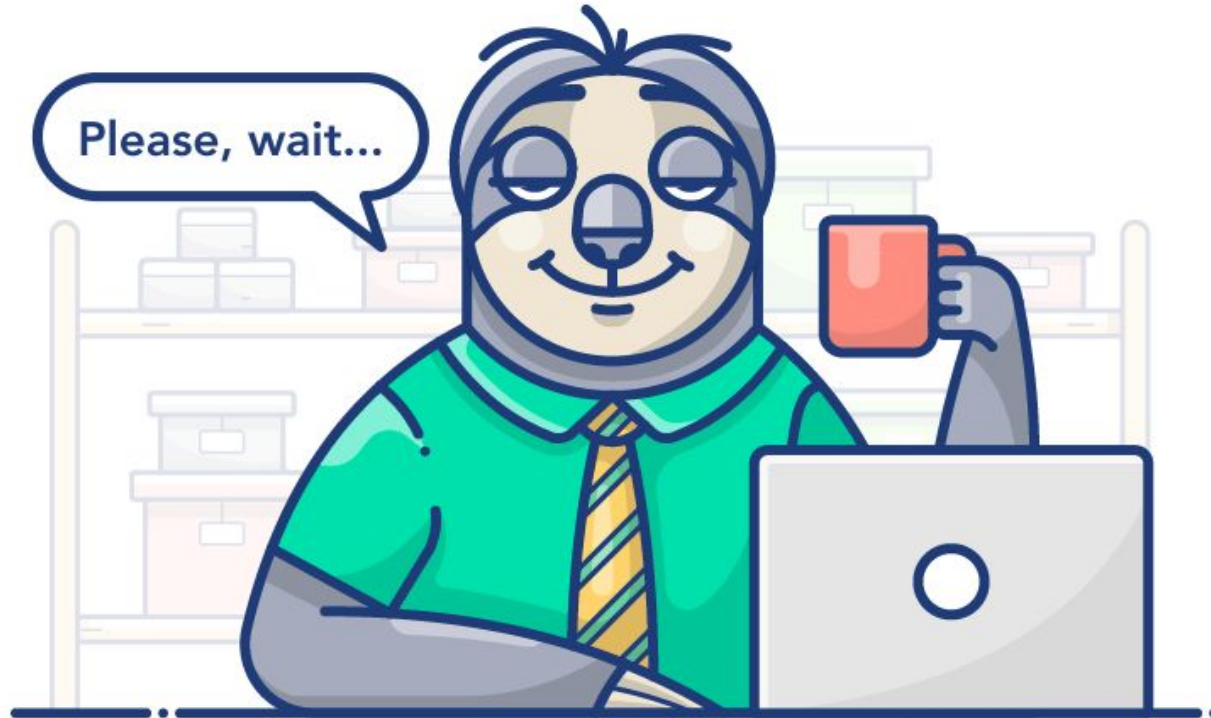
СЕТЬ ДЕЛОВЫХ КОММУНИКАЦИЙ

Объединяет людей и компании, системы
управления и учета, бизнес-процессы
и документы в одной системе

Начать работу



Страница работает медленно



Больше памяти - медленнее страница

BloatBusters:

Eliminating memory leaks in Gmail

Loreena Lee, Robert Hundt



Виснет вся система



Опаньки...

При загрузке этой страницы возникли неполадки.

[ПЕРЕЗАГРУЗИТЬ](#)

Single Page Application & F5

Задача: находить все критичные утечки
памяти в выпускаемой версии

Задача: находить все критичные утечки
памяти в выпускаемой версии **автоматически**

Задача: находить все критичные утечки
памяти в выпускаемой версии **автоматически**

В чем сложность

В чем сложность

- > Не тривиально анализировать

В чем сложность

- > Не тривиально анализировать
- > Мало информации

В чем сложность

- > Не тривиально анализировать
- > Мало информации
- > Мало инструментов

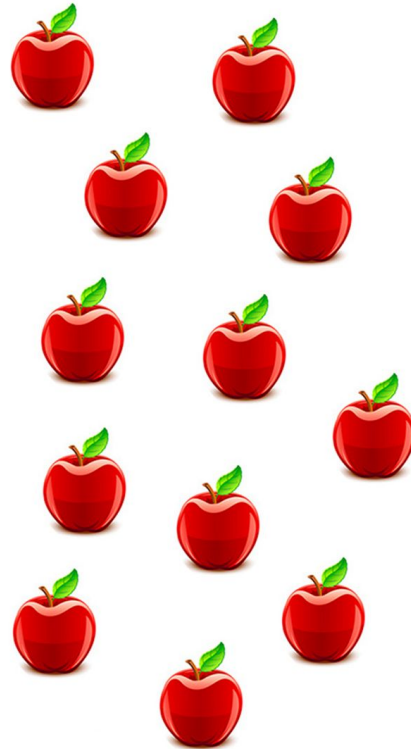
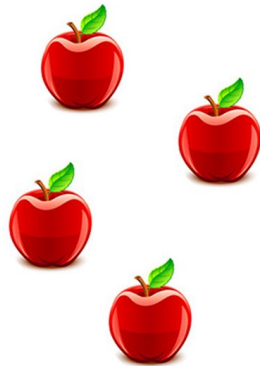
В чем сложность

- > Не тривиально анализировать
- > Мало информации
- > Мало инструментов
- > Сложно обеспечить покрытие



«Что такое утечка?»

Было мало, стало много



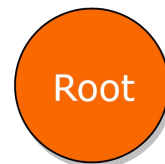
```
>var heisen = "bug";
```

```
var version = 2018;
```

```
var a = {b: [] };
```

```
a.b.push(heisen, version);
```

```
a.b = undefined;
```



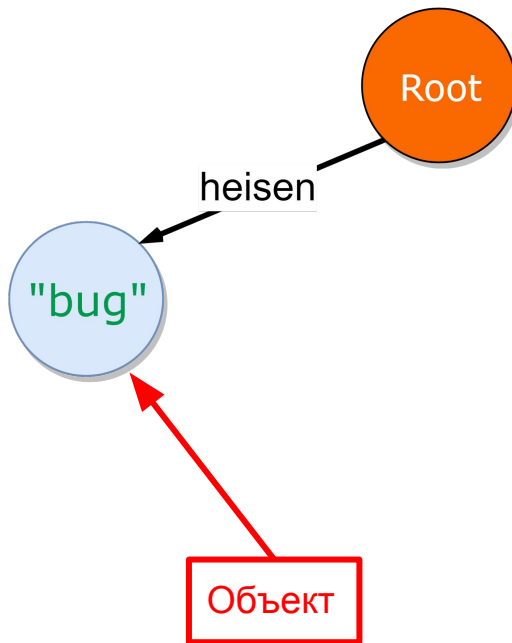
```
var heisen = "bug";
```

```
>var version = 2018;
```

```
var a = {b: [] };
```

```
a.b.push(heisen, version);
```

```
a.b = undefined;
```



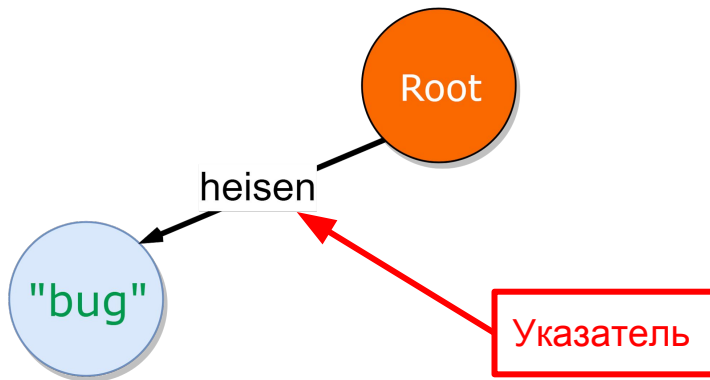

```
var heisen = "bug";
```

```
>var version = 2018;
```

```
var a = {b: [] };
```

```
a.b.push(heisen, version);
```

```
a.b = undefined;
```



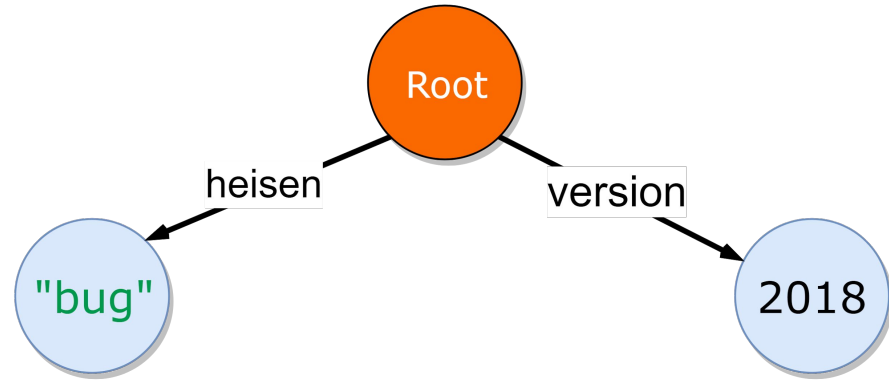
```
var heisen = "bug";
```

```
var version = 2018;
```

```
>var a = {b: [] };
```

```
a.b.push(heisen, version);
```

```
a.b = undefined;
```



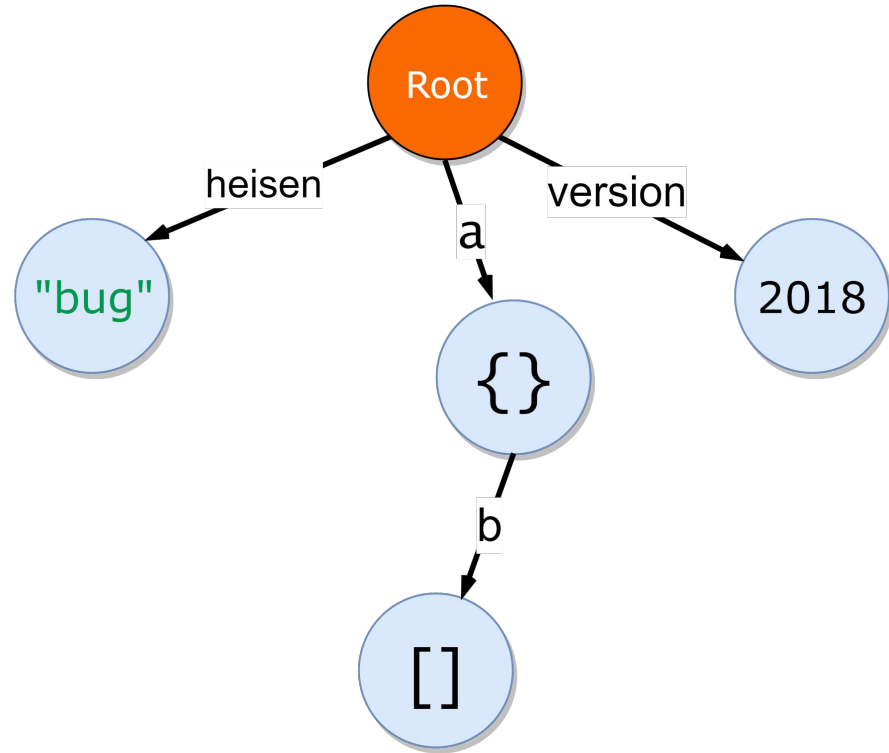
```
var heisen = "bug";
```

```
var version = 2018;
```

```
var a = {b: [] };
```

```
>a.b.push(heisen, version);
```

```
a.b = undefined;
```



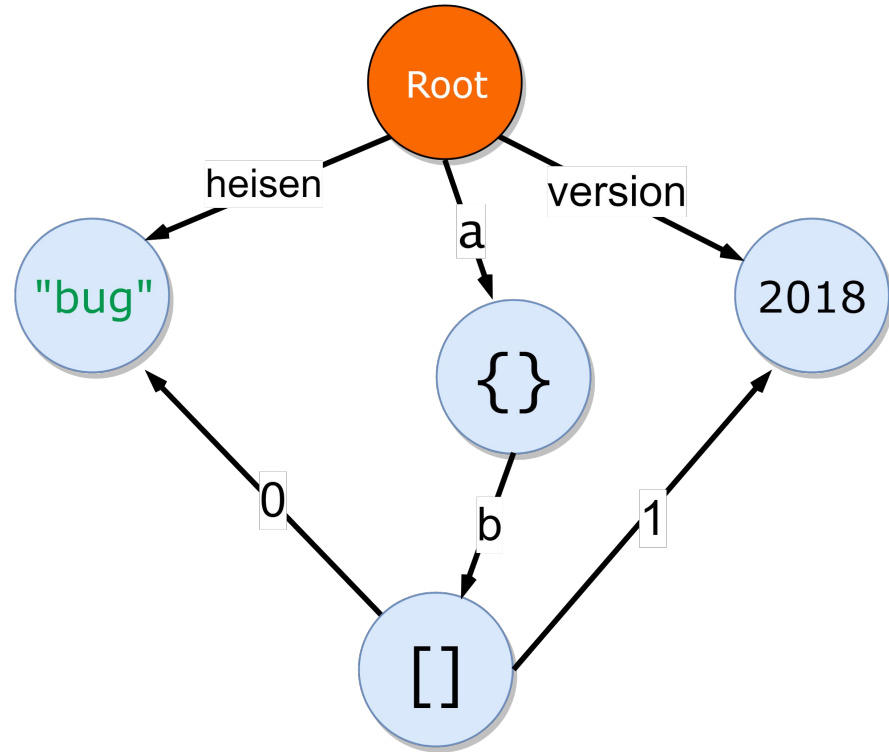
```
var heisen = "bug";
```

```
var version = 2018;
```

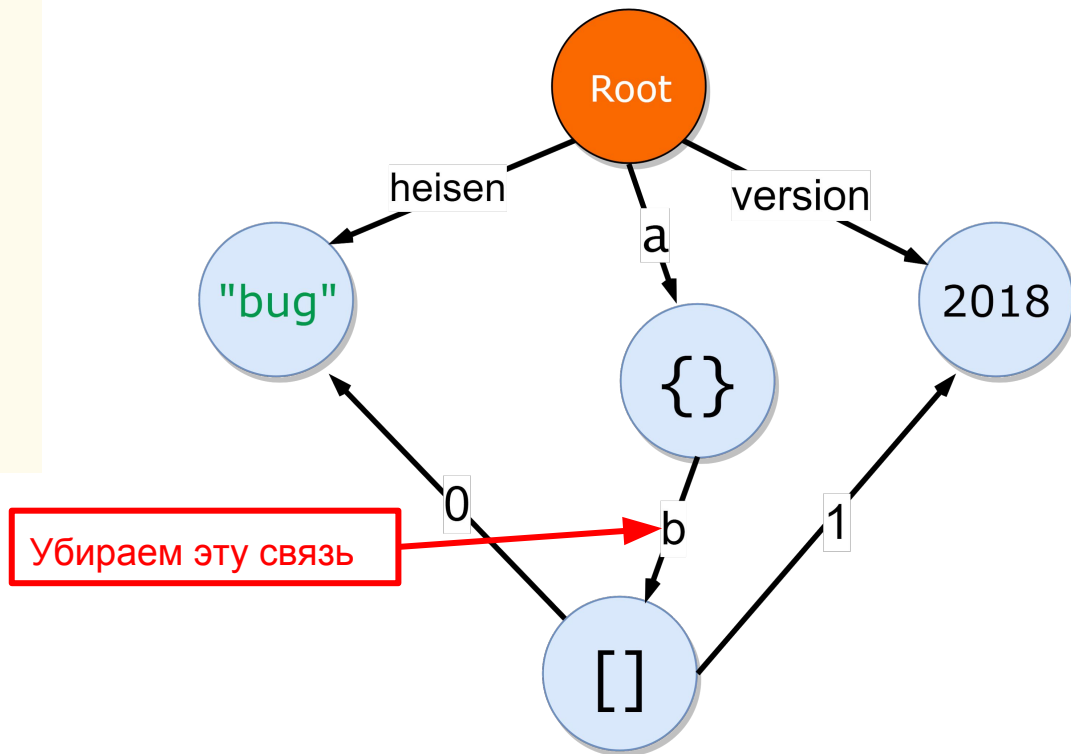
```
var a = {b: [] };
```

```
a.b.push(heisen, version);
```

```
>a.b = undefined;
```



```
var heisen = "bug";  
  
var version = 2018;  
  
var a = {b: [] };  
  
a.b.push(heisen, version);  
  
>a.b = undefined;
```



```
var heisen = "bug";
```

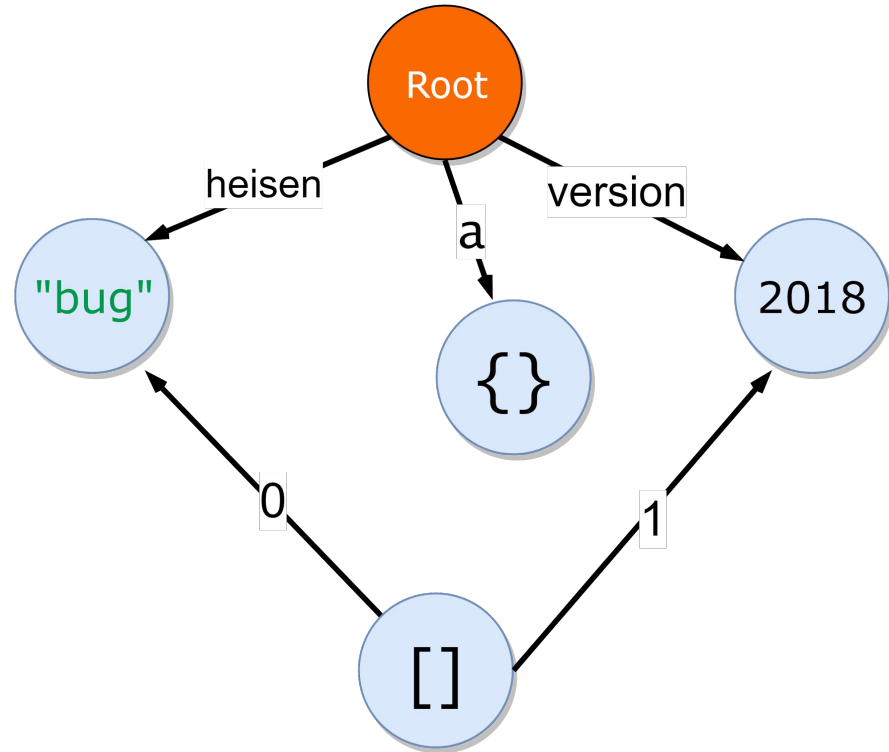
```
var version = 2018;
```

```
var a = {b: [] };
```

```
a.b.push(heisen, version);
```

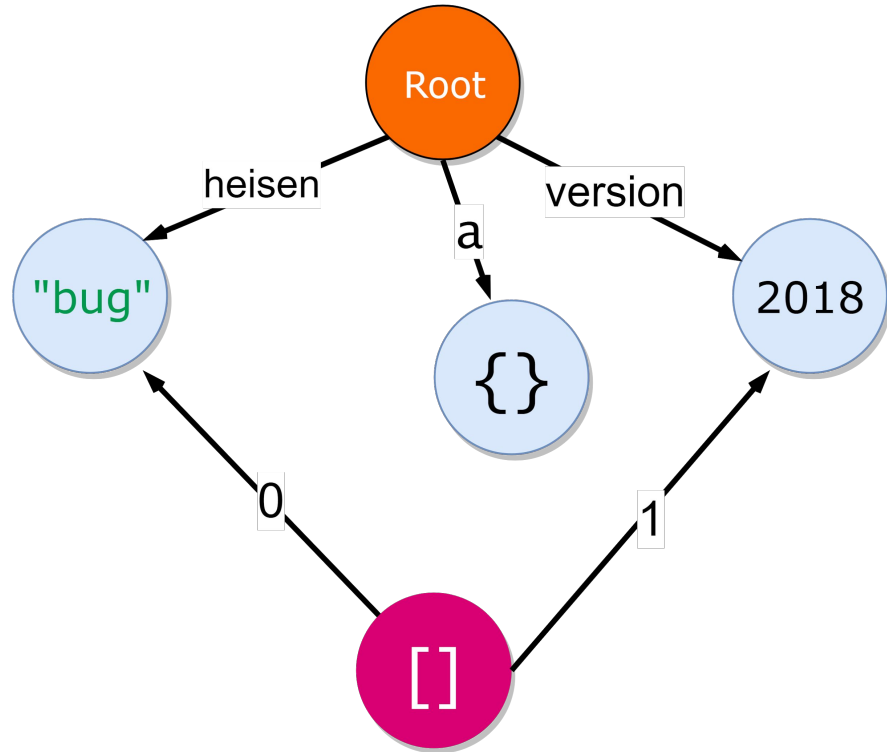
```
a.b = undefined;
```

>



```
var heisen = "bug";  
  
var version = 2018;  
  
var a = {b: [] };  
  
a.b.push(heisen, version);  
  
a.b = undefined;
```

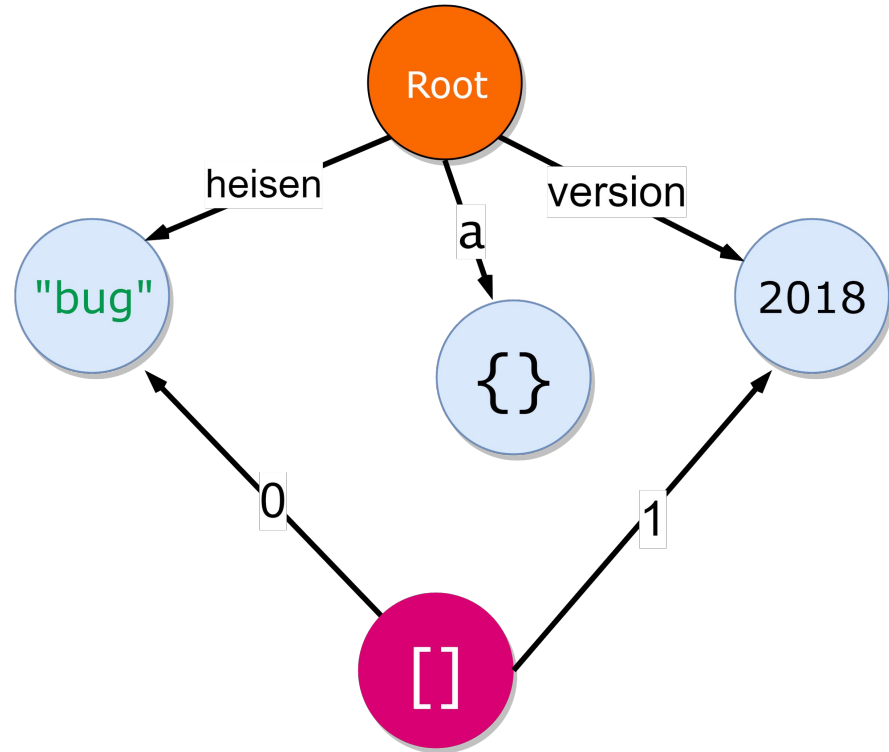
>



```
var heisen = "bug";  
  
var version = 2018;  
  
var a = {b: [] };  
  
a.b.push(heisen, version);  
  
a.b = undefined;
```

>

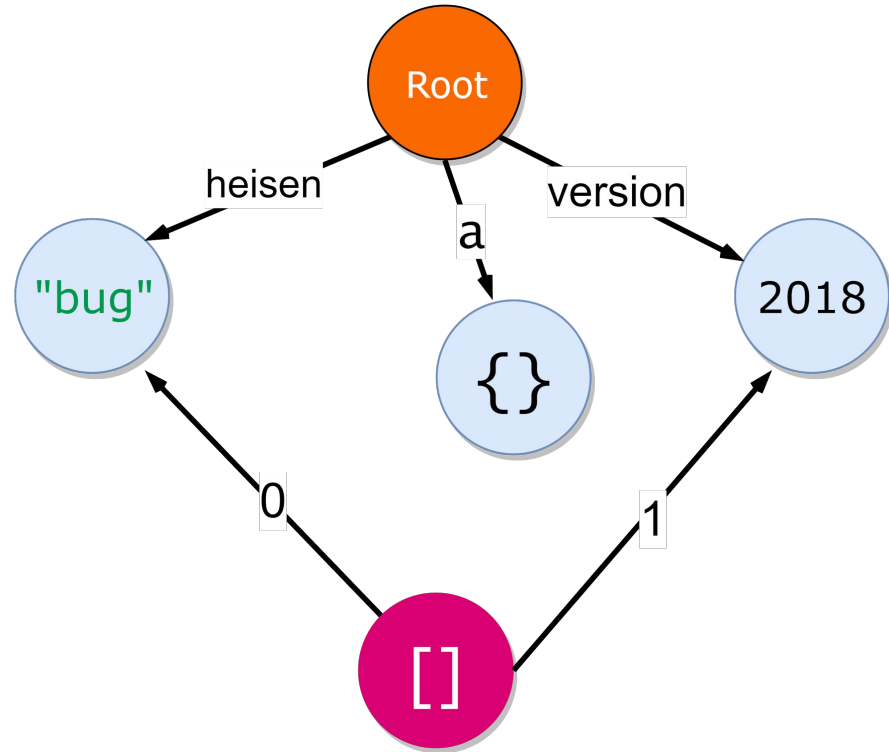
malloc & free




```
var heisen = "bug";  
  
var version = 2018;  
  
var a = {b: [] };  
  
a.b.push(heisen, version);  
  
a.b = undefined;
```

>

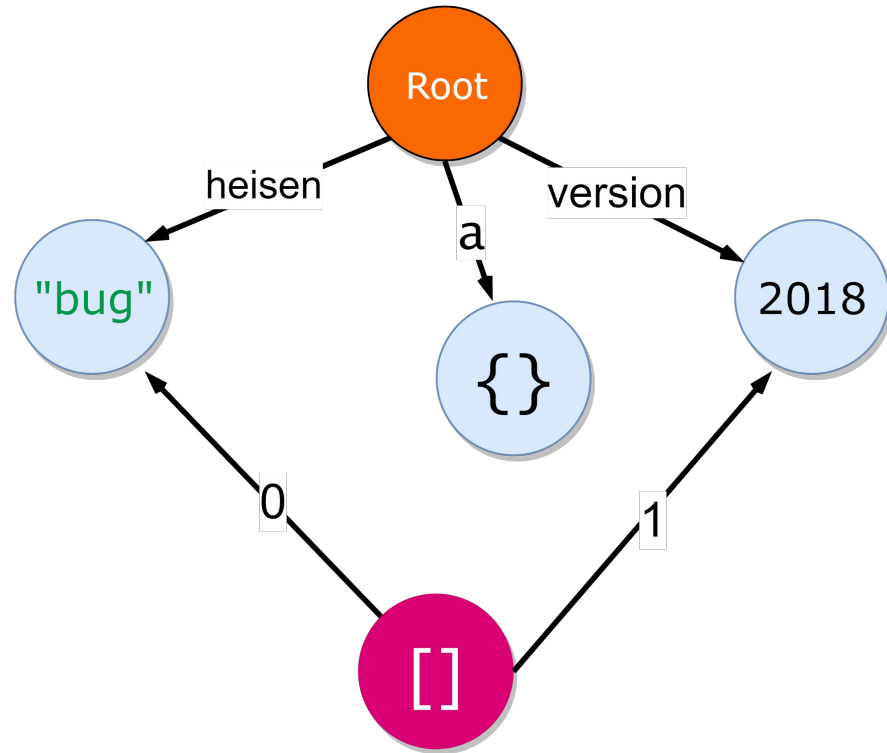
Garbage Collector (GC)



```
var heisen = "bug";  
  
var version = 2018;  
  
var a = {b: [] };  
  
a.b.push(heisen, version);  
  
a.b = undefined;
```

>

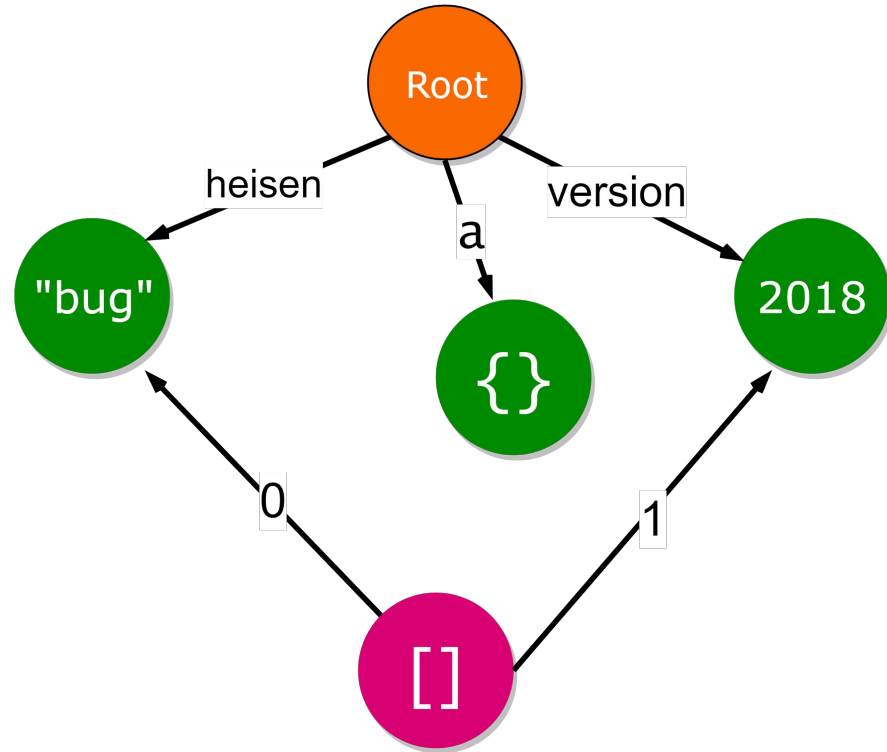
GC MARK & SWEEP



```
var heisen = "bug";  
  
var version = 2018;  
  
var a = {b: [] };  
  
a.b.push(heisen, version);  
  
a.b = undefined;
```

>

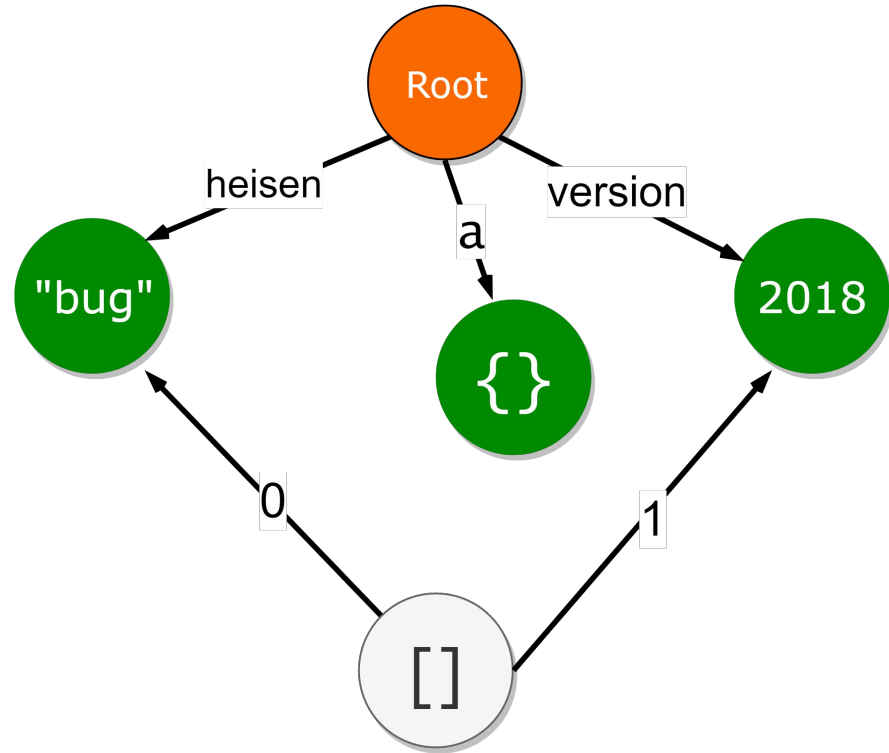
GC MARK & SWEEP



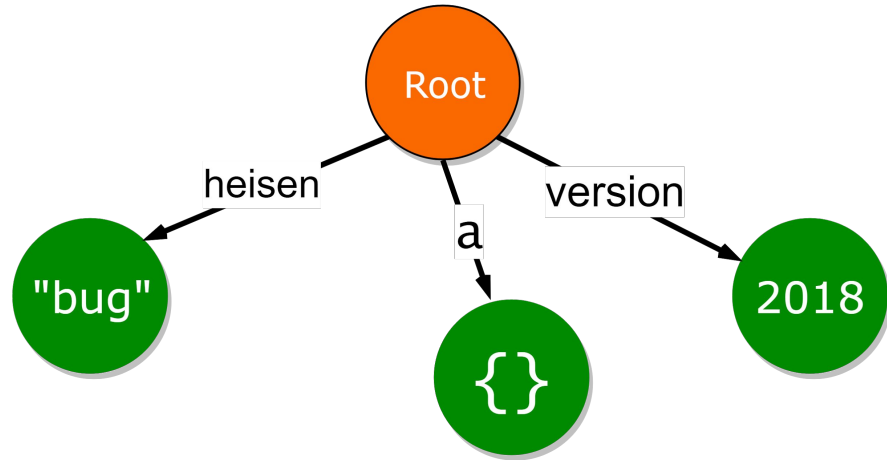
```
var heisen = "bug";  
  
var version = 2018;  
  
var a = {b: [] };  
  
a.b.push(heisen, version);  
  
a.b = undefined;
```

>

GC MARK & SWEEP



```
var heisen = "bug";  
  
var version = 2018;  
  
var a = {b: [] };  
  
a.b.push(heisen, version);  
  
a.b = undefined;
```



GC MARK & SWEEP

Утечка памяти - объекты, которые
доступны из кода, но нам больше
не нужны

Утечка памяти - объекты, которые
доступны из кода, но нам больше
не нужны

Утечка памяти - объекты, которые
доступны из кода, но нам больше
не нужны

Пример



Список заключенных

Фамилия Имя Отчество

Добавить

#	Фамилия	Имя	Отчество
1	Милославский	Юрий	Иванович
2	Люпен	Арсен	Петрович

```
<input id="name">  
<div id="canvas">  
  <button>  
    Посадить  
  </button>  
  <table id="prisoners">  
  </table>  
</div>
```

```
<input id="name">
<div id="canvas">
  <button>
    Посадить
  </button>
  <table id="prisoners">
  </table>
</div>
```

```
function add(){
  name = $('#name').val();
  tr = createRow(name);
  $('#prisoners').append(tr);
}
```

```
<input id="name">
<div id="canvas">
  <button onclick="add()">
    Посадить
  </button>
  <table id="prisoners">
  </table>
</div>
```

```
function add(){
  name = $('#name').val();
  tr = createRow(name);
  $('#prisoners').append(tr);
}
```

Список заключенных

Жан Вальжан Сидорович

Добавить

#	Фамилия	Имя	Отчество
1	Милославский	Юрий	Иванович
2	Люпен	Арсен	Петрович

Список заключенных

Жан Вальжан Сидорович

Добавить

#	Фамилия	Имя	Отчество
1	Милославский	Юрий	Иванович
2	Люпен	Арсен	Петрович

Список заключенных

Жан Вальжан Сидорович

Добавить

#	Фамилия	Имя	Отчество
1	Милославский	Юрий	Иванович
2	Люпен	Арсен	Петрович
3	Вальжан	Жан	Сидорович

Список заключенных

Фамилия Имя Отчество

Добавить

#	Фамилия	Имя	Отчество
1	Милославский	Юрий	Иванович
2	Люпен	Арсен	Петрович
3	Вальжан	Жан	Сидорович

Освободить ушлёпка

```
<input id="name">
<div id="canvas">
  <button onclick="add()">
    Посадить
  </button>
  <button onclick="free()">
    Освободить ушлёпка
  </button>
  <table id="prisoners">
  </table>
</div>
```

```
function free(){
  $(' .selected').empty();
}
```

Список заключенных

Фамилия Имя Отчество

Добавить

#	Фамилия	Имя	Отчество
1	Милославский	Юрий	Иванович
2	Люпен	Арсен	Петрович
3	Вальжан	Жан	Сидорович

Освободить ушлёпка

Список заключенных

Фамилия Имя Отчество

Добавить

#	Фамилия	Имя	Отчество
1	Милославский	Юрий	Иванович
2	Люпен	Арсен	Петрович

Освободить ушлёпка

Список заключенных

Фамилия Имя Отчество

Добавить

#	Фамилия	Имя	Отчество
1	Милославский	Юрий	Иванович
2	Люпен	Арсен	Петрович

Освободить всех ушлёпков

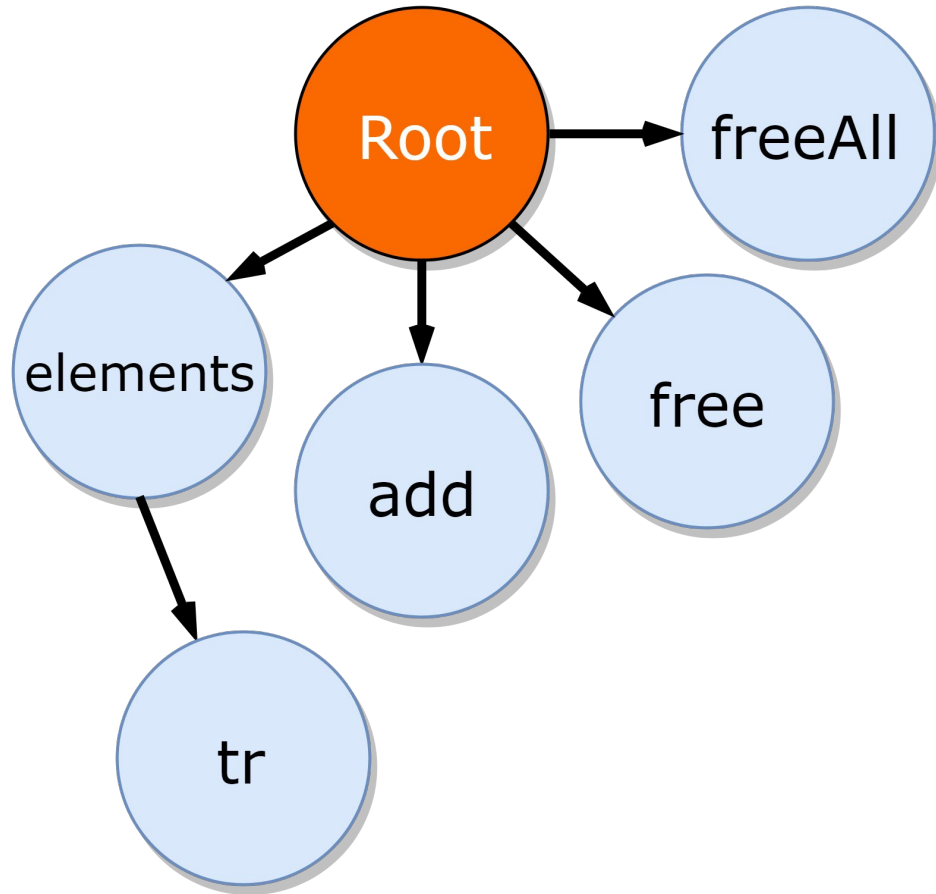
Освободить ушлёпка

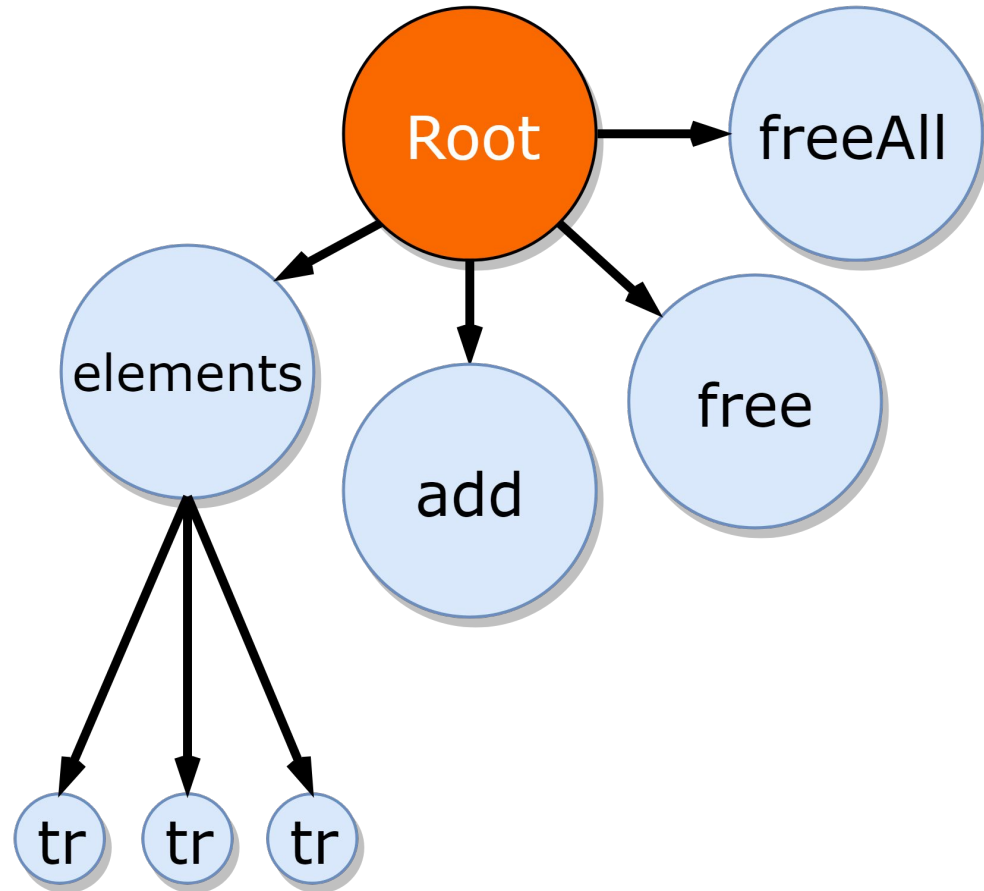
```
<input id="name">
<div id="canvas">
  <button onclick="add()">
    Посадить
  </button>
  <button onclick="free()">
    Освободить ушлёпка
  </button>
  <button onclick="freeAll()">
    Освободить всех ушлёпков
  </button>
  <table id="prisoners">
  </table>
</div>
```

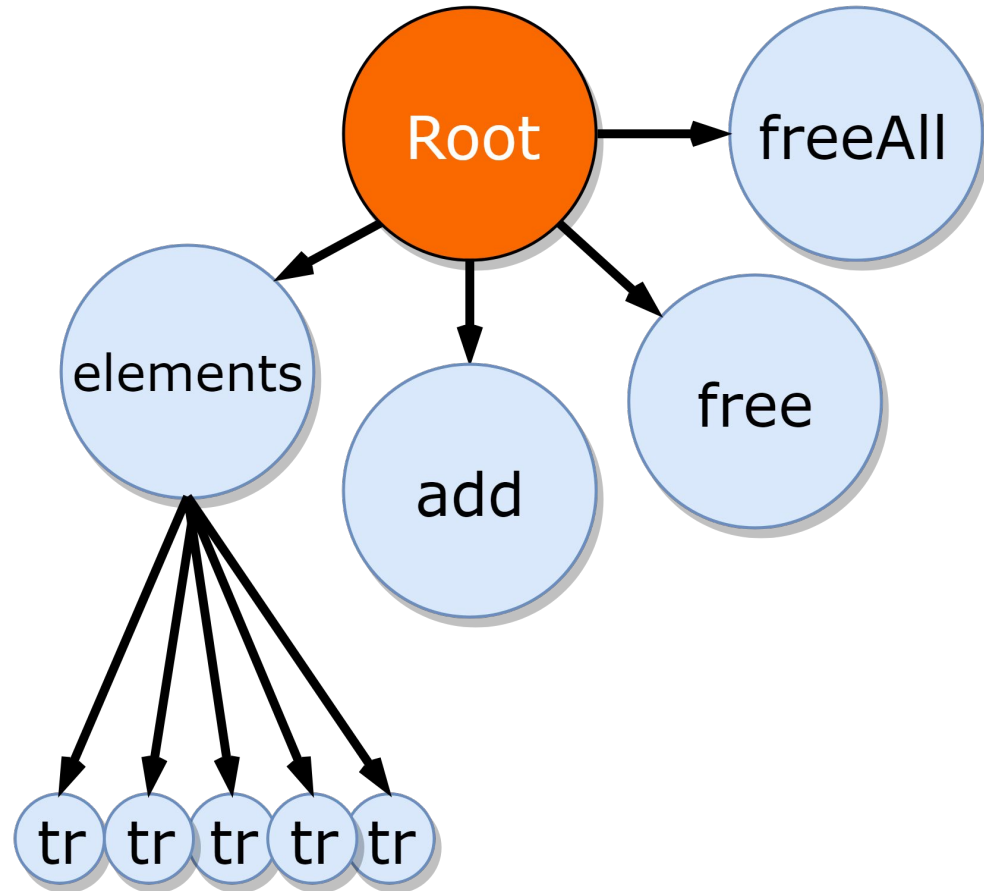
```
var elements = []

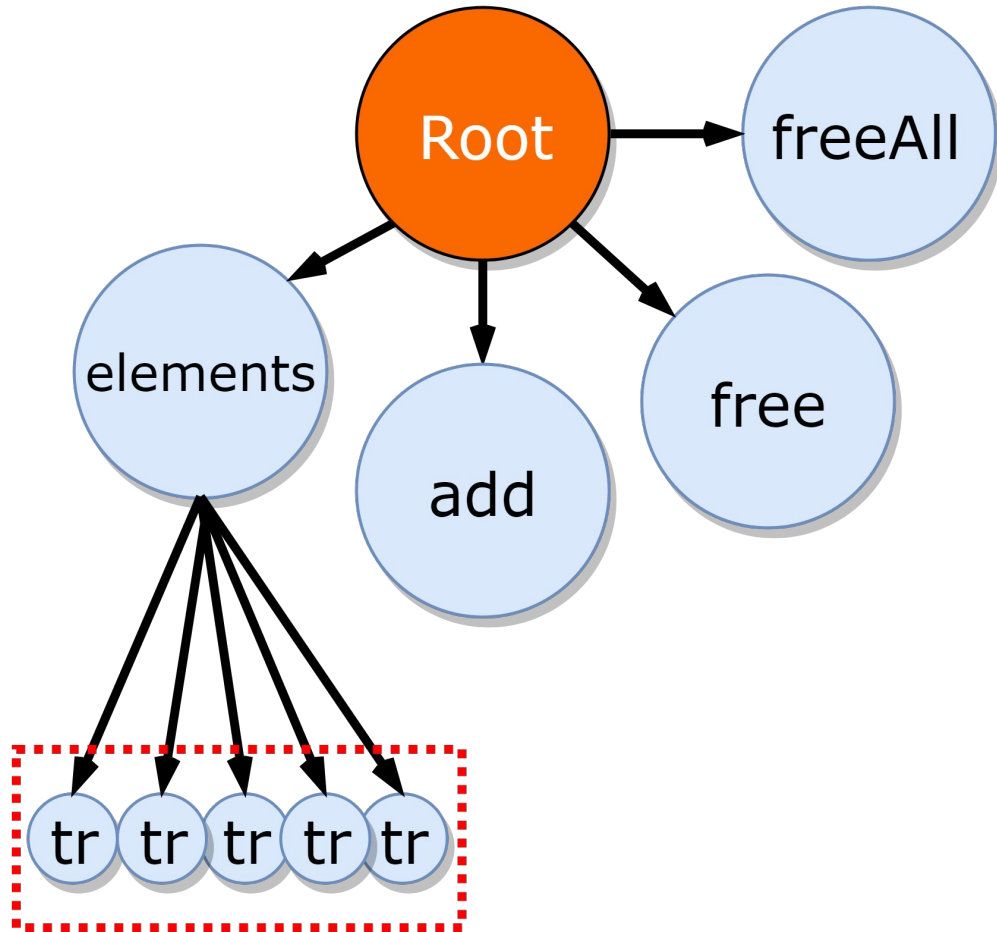
function add(){
  name = $('#name').val();
  tr = createRow(name);
  $('#prisoners').append(tr);
  elements.push(tr) ;
}

function freeAll(){
  $.each(elements,
    function (index, value){
      value.empty()
    });
}
```









```
<input id="name">
<div id="canvas">
  <button onclick="add()">
    Посадить
  </button>
  <button onclick="free()">
    Освободить ушлёпка
  </button>
  <button onclick="freeAll()">
    Освободить всех ушлёпков
  </button>
  <table id="prisoners">
  </table>
</div>
```

```
function freeAll(){
  $('#prisoners').empty(elements);
  elements.length = 0;
}
```



«Инструменты»

Диспетчер задач Windows

Файл Параметры Вид Справка

Приложения Процессы Службы Быстродействие Сеть Пользователи

Имя образа	Пользо...	ЦП	Память (частный рабочий набор)	Командная строка
chrome.exe	Serge	03	531 012 КБ	"C:\Program Files (x86)\Goog
chrome.exe	Serge	00	513 048 КБ	"C:\Program Files (x86)\Goog
chrome.exe	Serge	00	383 528 КБ	"C:\Program Files (x86)\Goog
chrome.exe	Serge	00	358 156 КБ	"C:\Program Files (x86)\Goog
chrome.exe	Serge	00	336 844 КБ	"C:\Program Files (x86)\Goog
chrome.exe	Serge	00	316 772 КБ	"C:\Program Files (x86)\Goog
chrome.exe	Serge	00	313 924 КБ	"C:\Program Files (x86)\Goog
chrome.exe	Serge	00	303 260 КБ	"C:\Program Files (x86)\Goog
chrome.exe	Serge	00	298 588 КБ	"C:\Program Files (x86)\Goog
chrome.exe	Serge	00	292 304 КБ	"C:\Program Files (x86)\Goog
chrome.exe	Serge	00	279 988 КБ	"C:\Program Files (x86)\Goog
chrome.exe	Serge	00	223 440 КБ	"C:\Program Files (x86)\Goog
ieexplore.exe *32	Serge	00	151 452 КБ	"C:\Program Files (x86)\Inter
Skype.exe *32	Serge	00	119 756 КБ	"C:\Program Files (x86)\Micr
chrome.exe	Serge	00	112 372 КБ	"C:\Program Files (x86)\Goog

Отображать процессы всех пользователей

Завершить процесс

Процессов: 123 Загрузка ЦП: 9% Физическая память: 87%

Диспетчер задач - Google Chrome

Задача	Объем потребляемой памяти	ЦПУ	Сеть	Идентификатор процесса	Память JavaScript
Расширение: Блокировщик...	490 864К	3.1	0	4288	378 844 КБ (363 619 К...
Приложение: Untitled presentation - Google Sli...	339 800К	0.0	23 Б/с	2816	180 436 КБ (154 294 К...
Вкладка: (3) Роман Поборчий: "Предвзвешенные ...	500 176К	0.0	0	1372	133 392 КБ (115 432 К...
Вкладка: (3) Как научить роботов играть в игр...					
Вкладка: (4) Настоящий Морской Волк - YouTu...					
Субфрейм: https://doubleclick.net/	251 936К	0.0	0	6972	113 412 КБ (75 459 К...
Субфрейм: https://doubleclick.net/					
Вкладка: Андрей Солнцев — Flaky tests - YouT...	304 176К	0.0	0	1108	96 016 КБ (84 485 КБ ...
Вкладка: (4) Gardener Ignores the Laws of Natur...					
Субфрейм: https://googlesyndication.com/	150 408К	0.0	0	5640	93 456 КБ (83 381 КБ ...
Вкладка: (3) Понятные слайды / Роман Побор...	351 980К	0.0	0	2992	92 432 КБ (75 443 КБ ...
Субфрейм: https://youtube.com/					
Приложение: Промышленный подход к поиск...	219 288К	1.6	1 851 Б/с	11116	91 572 КБ (77 603 КБ ...

Завершить процесс

Диспетчер задач - Google Chrome

Задача	Объем потребляемой памяти	ЦПУ	Сеть	Идентификатор процесса	Память JavaScript
Расширение: Блокировщик...	490 864К	3.1	0	4288	378 844 КБ (363 619 К...
Приложение: Untitled presentation - Google Sli...	339 800К	0.0	23 Б/с	2816	180 436 КБ (154 294 К...
Вкладка: (3) Роман Поборчий: "Предрассудки ...	500 176К	0.0	0	1372	133 392 КБ (115 432 К...
Вкладка: (3) Как научить роботов играть в игр...					
Вкладка: (4) Настоящий Морской Волк - YouTu...					
Субфрейм: https://doubleclick.net/	251 936К	0.0	0	6972	113 412 КБ (75 459 К...
Субфрейм: https://doubleclick.net/					
Вкладка: Андрей Солнцев — Flaky tests - YouT...	304 176К	0.0	0	1108	96 016 КБ (84 485 КБ ...
Вкладка: (4) Gardener Ignores the Laws of Natur...					
Субфрейм: https://googlesyndication.com/	150 408К	0.0	0	5640	93 456 КБ (83 381 КБ ...
Вкладка: (3) Понятные слайды / Роман Побор...	351 980К	0.0	0	2992	92 432 КБ (75 443 КБ ...
Субфрейм: https://youtube.com/					
Приложение: Промышленный подход к поиск...	219 288К	1.6	1 851 Б/с	11116	91 572 КБ (77 603 КБ ...

Завершить процесс

Задача	Объём потребляемой памяти	Память JavaScript
Презентаци..	339 МБ	180 МБ

Select profiling type

Heap snapshot

Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.

Allocation instrumentation on timeline

Allocation timelines show instrumented JavaScript memory allocations over time. Once profile is recorded you can select a time interval to see objects that were allocated within it and still alive by the end of recording. Use this profile type to isolate memory leaks.

Record allocation stacks (extra performance overhead)

Allocation sampling

Record memory allocations using sampling method. This profile type has minimal performance overhead and can be used for long running operations. It provides good approximation of allocations broken down by JavaScript execution stack.

Select profiling type

 Heap snapshot

Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.

 Allocation instrumentation on timeline

Allocation timelines show instrumented JavaScript memory allocations over time. Once profile is recorded you can select a time interval to see objects that were allocated within it and still alive by the end of recording. Use this profile type to isolate memory leaks.

Record allocation stacks (extra performance overhead)

 Allocation sampling

Record memory allocations using sampling method. This profile type has minimal performance overhead and can be used for long running operations. It provides good approximation of allocations broken down by JavaScript execution stack.

Record allocation stacks (extra performance overhead)

Allocation sampling

Record memory allocations using sampling method. This profile type has minimal performance overhead and can be used for long running operations. It provides good approximation of allocations broken down by JavaScript execution stack.

Select JavaScript VM instance

189 MB / 218 MB	online.sbis.ru
-----------------	----------------

Take snapshot

Load

Constructor	Distance	Shallow Size	Retained Size
▼ HTMLDivElement ×1953	4	23 280 0 %	855 584 1 %
▶ HTMLDivElement @38969	9	40 0 %	7 624 0 %
▶ HTMLDivElement @35837	9	40 0 %	7 616 0 %
▶ HTMLDivElement @38823	8	40 0 %	7 376 0 %
▶ HTMLDivElement @35853	11	40 0 %	7 328 0 %
Retainers			
Object	Distance	Shallow Size	Retained Size
▼ element in Object @839049	8	24 0 %	19 632 0 %
▼ _options in t @823325	7	24 0 %	32 528 0 %
▼ bound_this in native_bind() @838917	6	48 0 %	88 0 %
▼ handler in Object @1022501	5	88 0 %	88 0 %
▼ [2] in Array @1022495	4	32 0 %	3 720 0 %
▼ resize in Object @890843	3	56 0 %	7 728 0 %
▼ events in Object @890841	2	56 0 %	7 784 0 %
▶ jQuery331030469894556500331 in Wind	1	56 0 %	164 304 0 %

Constructor	Distance	Shallow Size	Retained Size
▼ HTMLDivElement ×1953	4	23 280 0 %	855 584 1 %
▶ HTMLDivElement @38969	9	40 0 %	7 624 0 %
▶ HTMLDivElement @35837	9	40 0 %	7 616 0 %
▶ HTMLDivElement @38823	8	40 0 %	7 376 0 %
▶ HTMLDivElement @35853	11	40 0 %	7 328 0 %
Retainers			
Object	Distance	Shallow Size	Retained Size
▼ element in Object @839049	8	24 0 %	19 632 0 %
▼ _options in t @823325	7	24 0 %	32 528 0 %
▼ bound_this in native_bind() @838917	6	48 0 %	88 0 %
▼ handler in Object @1022501	5	88 0 %	88 0 %
▼ [2] in Array @1022495	4	32 0 %	3 720 0 %
▼ resize in Object @890843	3	56 0 %	7 728 0 %
▼ events in Object @890841	2	56 0 %	7 784 0 %
▶ jQuery331030469894556500331 in Wind	1	56 0 %	164 304 0 %

Constructor	Distance	Shallow Size	Retained Size
▼ HTMLDivElement ×1953	4	23 280 0 %	855 584 1 %
▶ HTMLDivElement @38969	9	40 0 %	7 624 0 %
▶ HTMLDivElement @35837	9	40 0 %	7 616 0 %
▶ HTMLDivElement @38823	8	40 0 %	7 376 0 %
▶ HTMLDivElement @35853	11	40 0 %	7 328 0 %
Retainers			
Object	Distance	Shallow Size	Retained Size
▼ element in Object @839049	8	24 0 %	19 632 0 %
▼ _options in t @823325	7	24 0 %	32 528 0 %
▼ bound_this in native_bind() @838917	6	48 0 %	88 0 %
▼ handler in Object @1022501	5	88 0 %	88 0 %
▼ [2] in Array @1022495	4	32 0 %	3 720 0 %
▼ resize in Object @890843	3	56 0 %	7 728 0 %
▼ events in Object @890841	2	56 0 %	7 784 0 %
▶ jQuery331030469894556500331 in Wind	1	56 0 %	164 304 0 %

Constructor	Distance	Shallow Size	Retained Size
▼ HTMLDivElement ×1953	4	23 280 0 %	855 584 1 %
▶ HTMLDivElement @38969	9	40 0 %	7 624 0 %
▶ HTMLDivElement @35837	9	40 0 %	7 616 0 %
▶ HTMLDivElement @38823	8	40 0 %	7 376 0 %
▶ HTMLDivElement @35853	11	40 0 %	7 328 0 %
Retainers			
Object	Distance	Shallow Size	Retained Size
▼ element in Object @839049	8	24 0 %	19 632 0 %
▼ _options in t @823325	7	24 0 %	32 528 0 %
▼ bound_this in native_bind() @838917	6	48 0 %	88 0 %
▼ handler in Object @1022501	5	88 0 %	88 0 %
▼ [2] in Array @1022495	4	32 0 %	3 720 0 %
▼ resize in Object @890843	3	56 0 %	7 728 0 %
▼ events in Object @890841	2	56 0 %	7 784 0 %
▶ jQuery331030469894556500331 in Wind	1	56 0 %	164 304 0 %

Constructor	Distance	Shallow Size	Retained Size
▼HTMLDivElement ×1953	4	23 280 0 %	855 584 1 %
▶HTMLDivElement @38969	9	40 0 %	7 624 0 %
▶HTMLDivElement @35837	9	40 0 %	7 616 0 %
▶HTMLDivElement @38823	8	40 0 %	7 376 0 %
▶HTMLDivElement @35853	11	40 0 %	7 328 0 %
Retainers			
Object	Distance	Shallow Size	Retained Size
▼element in Object @839049	8	24 0 %	19 632 0 %
▼_options in t @823325	7	24 0 %	32 528 0 %
▼bound_this in native_bind() @838917	6	48 0 %	88 0 %
▼handler in Object @1022501	5	88 0 %	88 0 %
▼[2] in Array @1022495	4	32 0 %	3 720 0 %
▼resize in Object @890843	3	56 0 %	7 728 0 %
▼events in Object @890841	2	56 0 %	7 784 0 %
▶jQuery331030469894556500331 in Wind	1	56 0 %	164 304 0 %

An iceberg floating in a blue ocean under a cloudy sky. The visible tip of the iceberg is on the right side of the frame, with the text 'DevTools' written on it. The much larger, submerged part of the iceberg is visible below the water line, extending across the bottom half of the image. The water is a deep blue, and the sky is a lighter blue with scattered white clouds.

DevTools

An iceberg floating in a blue ocean under a cloudy sky. The top part of the iceberg is above the water, and the much larger bottom part is submerged. The text 'DevTools' is written on the visible part, and 'chrome://tracing' is written on the submerged part.

DevTools

chrome://tracing

An iceberg floating in a blue ocean under a cloudy sky. The top part of the iceberg is above the water, and the much larger bottom part is submerged. The text 'DevTools' is written on the visible part, and 'chrome://tracing memory-infra' is written on the submerged part.

DevTools

chrome://tracing
memory-infra

Категории объектов

Process ▾	Private footprint ▾	blink_gc ▾	blink_objects ▾	cc ▾	discardable ▾	extensions ▾	font_caches ▾	gpu ▾
chrome.exe (pid 5484), uptime:101s	86,5 MiB							
chrome.exe (pid 6308), uptime:101s	101,8 MiB	3,8 MiB	0,0 MiB ▲		1,7 MiB		0,0 MiB	
chrome.exe (pid 9432), uptime:102s	413,1 MiB			8,8 MiB	0,0 MiB	0,5 MiB		4,1 MiB
chrome.exe (pid 9500), uptime:101s	82,8 MiB	1,6 MiB	0,0 MiB ▲				0,0 MiB	
chrome.exe (pid 9972), uptime:101s	160,5 MiB			0,0 MiB				3,1 MiB
chrome.exe (pid 10196): СБИС, uptime:101s	138,8 MiB	5,1 MiB	0,0 MiB ▲	11,5 MiB	0,0 MiB		0,1 MiB	1,2 MiB
chrome.exe (pid 11768), uptime:23s	76,4 MiB	0,1 MiB	0,0 MiB ▲				0,0 MiB	
chrome.exe (pid 15112): chrome://tracing, uptime:31s	112,7 MiB	3,6 MiB	0,0 MiB ▲	22,0 MiB			0,0 MiB	3,0 MiB
Total	1 172,5 MiB	14,3 MiB	0,0 MiB	42,3 MiB	1,7 MiB	0,5 MiB	0,1 MiB	11,3 MiB

Процессы

Техника 3 снимков

BloatBusters:

Eliminating memory leaks in Gmail

Loreena Lee, Robert Hundt

Finding Leaks with DevTools: The "3 snapshot" technique

● Take a heap snapshot (#1)

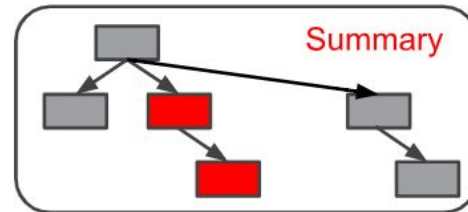
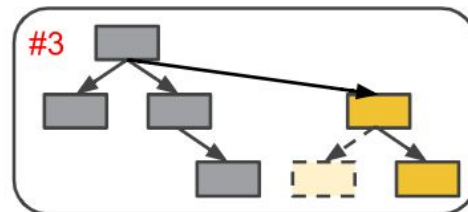
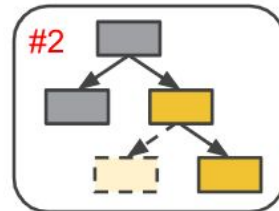
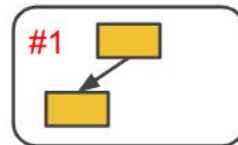
➡ Do stuff (i.e. compose/discard 7x)

● Take another heap snapshot (#2)

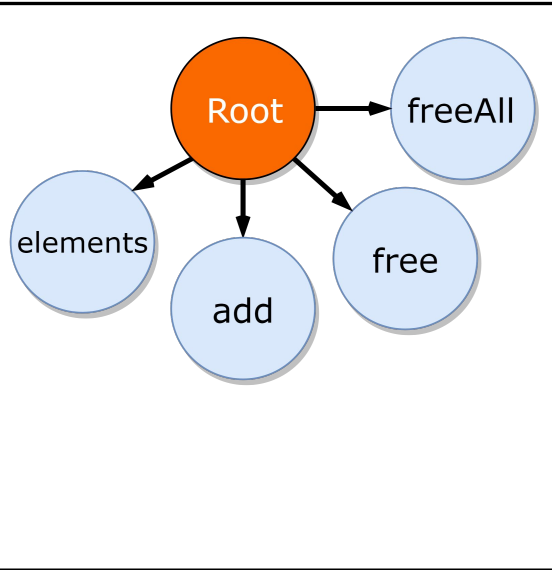
➡ Repeat the same stuff

● Take another heap snapshot (#3)

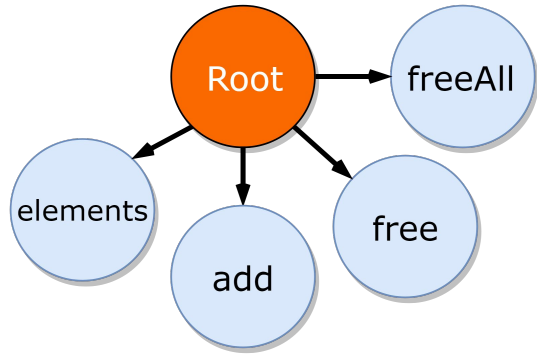
↔ Filter objects allocated between Snapshots 1 and 2 in Snapshot 3's "Summary" view



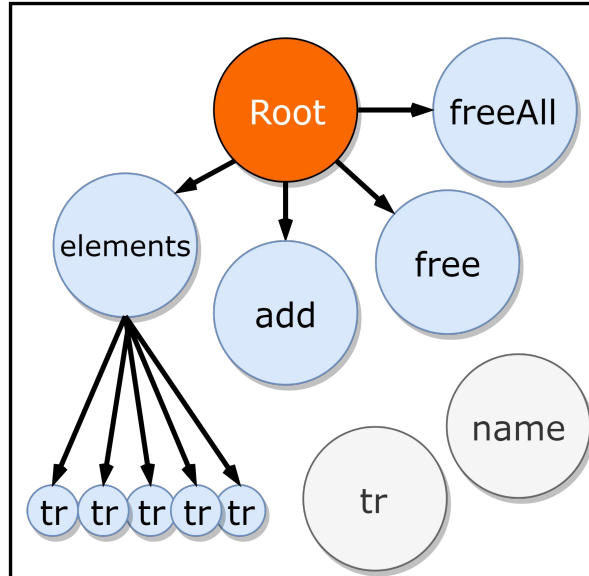
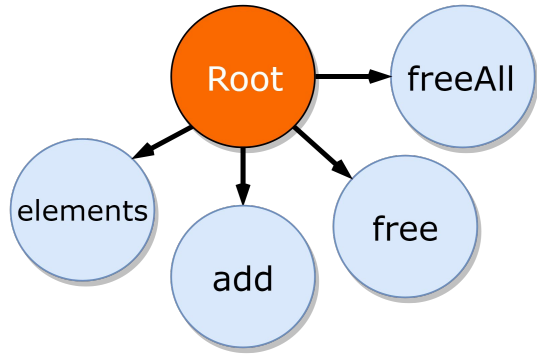
Делаем снимок памяти



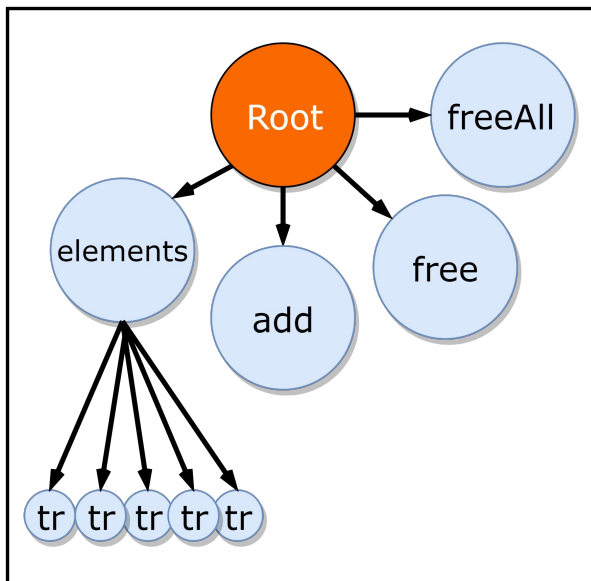
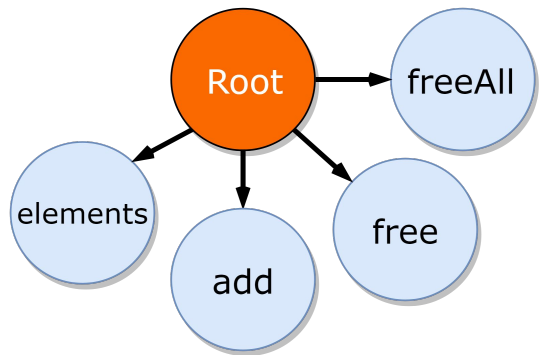
Выполняем действия (сценарий)



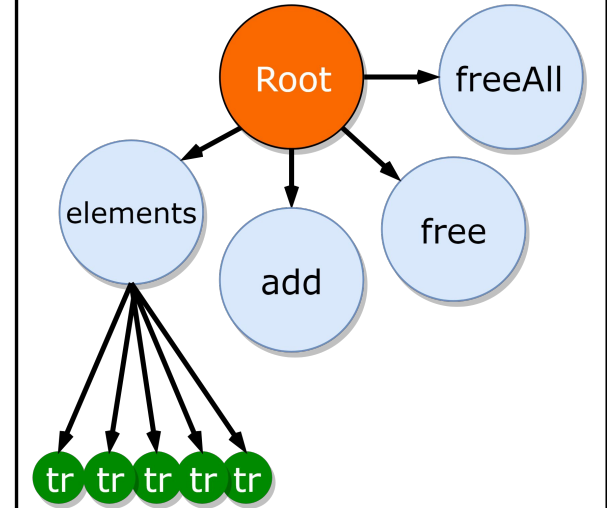
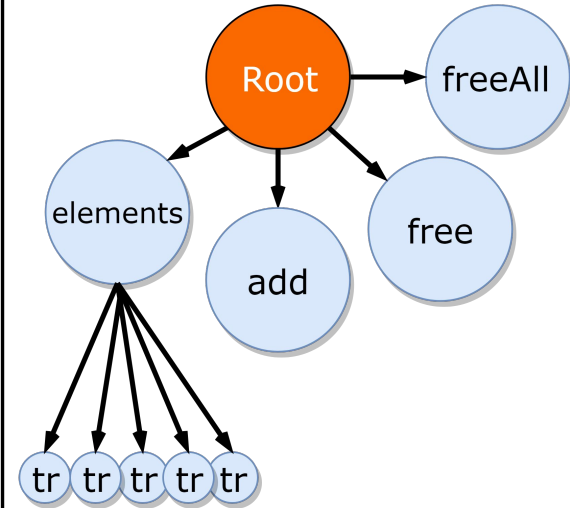
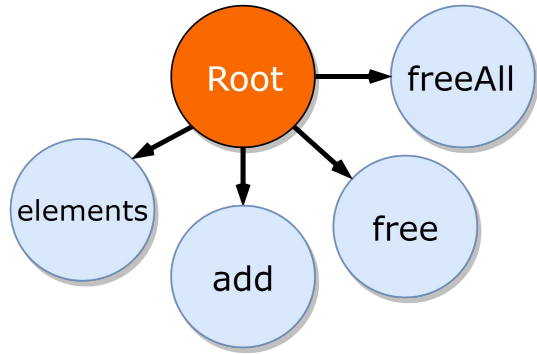
Делаем второй снимок памяти



Предварительно запускаем GC



Анализируем разницу



Elements		Console	Sources	Network	Performance	Memory	Application	»	17	⋮	✕
		Summary	Class filter	All objects							
Profiles HEAP SNAPSHOTS Snapshot 1 Save 8.1 MB	Constructor	Distance	Shallow Size		Retained Size						
	▶ (system) ×58437	2	3 452 472	41 %	4 560 192	54 %					
	▶ (array) ×11896	2	2 041 376	24 %	2 594 752	31 %					
	▶ (closure) ×10935	2	653 768	8 %	2 489 752	29 %					
	▶ (compiled code) ×6710	3	796 472	9 %	2 439 464	29 %					
	▶ Object ×4715	2	239 640	3 %	1 192 736	14 %					
	▶ Array ×2136	2	68 352	1 %	1 068 024	13 %					
	▶ system / Context ×1080	3	100 376	1 %	985 464	12 %					
	▶ Window / chrome-extension://lhdoppojpmngadmni...	1	56	0 %	935 784	11 %					
	▶ (string) ×17555	2	902 448	11 %	902 688	11 %					
▶ Window / chrome-extension://aiahmmpiobklfepioc...	1	56	0 %	867 000	10 %						
Retainers											
Object	Distance	Shallow Size		Retained Size							

Elements Console Sources Network Performance **Memory** Application » 17

Summary Class filter All objects

Profiles

HEAP SNAPSHOTS

Snapshot 1 8.1 MB

Snapshot 2 8.1 MB [Save](#)

Constructor

Class filter dropdown menu:

- All objects
- Objects allocated before Snapshot 1
- Objects allocated between Snapshot 1 and Snapshot 2

Object	Distance	Shallow Size	Retained Size
▶ (system) ×58815	2	2 021 568 24 %	2 776 54 %
▶ (array) ×11944	2	2 543 560 30 %	2 543 560 30 %
▶ (closure) ×10985	2	656 760 8 %	2 514 328 30 %
▶ (compiled code) ×6714	3	796 840 9 %	2 443 808 29 %
▶ Object ×4727	2	240 312 3 %	1 195 568 14 %
▶ Array ×2136	2	68 352 1 %	1 069 184 13 %
▶ system / Context ×1086	3	100 664 1 %	986 248 12 %
▶ Window / chrome-extension://lhdoppojpmngadmni...	1	56 0 %	936 840 11 %
▶ (string) ×17568	2	902 976 11 %	903 216 11 %
▶ Window / chrome-extension://aiahmmpiojklfepioc...	1	56 0 %	488 880 6 %

Retainers

Elements		Console	Sources	Network	Performance	Memory	Application	»		17	⋮	✕
		Summary		Class filter		Objects allocated between Snapshot 1 and Snapshot 2						
Profiles HEAP SNAPSHOTS Snapshot 1 8.1 MB Snapshot 2 Save 8.1 MB	Constructor		Distance	Shallow Size		Retained Size						
	▼ Detached HTMLTableRowElement ×10		3	400	0 %	400	0 %					
	▶ Detached HTMLTableRowElement @517323		3	40	0 %	40	0 %					
	▶ Detached HTMLTableRowElement @517325		3	40	0 %	40	0 %					
	▶ Detached HTMLTableRowElement @517327		3	40	0 %	40	0 %					
	▶ Detached HTMLTableRowElement @517329		3	40	0 %	40	0 %					
	▶ Detached HTMLTableRowElement @517331		3	40	0 %	40	0 %					
	▶ Detached HTMLTableRowElement @517333		3	40	0 %	40	0 %					
	▶ Detached HTMLTableRowElement @517335		3	40	0 %	40	0 %					
	▶ Detached HTMLTableRowElement @517337		3	40	0 %	40	0 %					
▶ Detached HTMLTableRowElement @517339		3	40	0 %	40	0 %						
Retainers												
Object		Distance	Shallow Size		Retained Size							

Elements		Console	Sources	Network	Performance	Memory	Application	»	17	⋮	✕
		Summary ▾ <input type="text" value="Class filter"/> Objects allocated between Snapshot 1 and Snapshot 2 ▾									
Profiles HEAP SNAPSHOTS Snapshot 1 8.1 MB	Constructor	Distance	Shallow Size		Retained Size						
	▼ Detached HTMLTableRowElement ×10	3	400	0 %	400	0 %					
	▶ Detached HTMLTableRowElement @517323	3	40	0 %	40	0 %					
	▶ Detached HTMLTableRowElement @517325	3	40	0 %	40	0 %					
	Retainers										
Snapshot 2 Save 8.1 MB	Object	Distance	Shallow Size		Retained Size						
	▼ [27] in Array @435833	2	32	0 %	1 592	0 %					
	▶ elements in Window / fiddle.jshell.net @377	1	56	0 %	117 304	1 %					
	▶ value in system / PropertyCell @435831	3	40	0 %	40	0 %					
	▶ 27 in (object elements)[] @516987	3	360	0 %	360	0 %					
	▶ [5] in Detached HTMLTableRowElement @517323	3	40	0 %	40	0 %					
	▶ [1] in Detached Text @263133728	4	0	0 %	0	0 %					

Elements		Console	Sources	Network	Performance	Memory	Application	»		17	⋮	✕
Summary		Class filter		Objects allocated between Snapshot 1 and Snapshot 2								
Profiles	Constructor	Distance	Shallow Size		Retained Size							
	▼ Detached HTMLTableRowElement ×10	3	400	0 %	400	0 %						
	▶ Detached HTMLTableRowElement @517323	3	40	0 %	40	0 %						
HEAP SNAPSHOTS	▶ Detached HTMLTableRowElement @517325	3	40	0 %	40	0 %						
	Retainers											
Snapshot 1 8.1 MB	Object	Distance	Shallow Size		Retained Size							
Snapshot 2 <u>Save</u> 8.1 MB	▼ [27] in Array @435833	2	32	0 %	1 592	0 %						
	▶ elements in Window / fiddle.jshell.net @377	1	56	0 %	117 304	1 %						
	▶ value in system / PropertyCell @435831	3	40	0 %	40	0 %						
	▶ 27 in (object elements)[] @516987	3	360	0 %	360	0 %						
	▶ [5] in Detached HTMLTableRowElement @517323	3	40	0 %	40	0 %						
	▶ [1] in Detached Text @263133728	4	0	0 %	0	0 %						

DevTools Allocation Timeline

Select profiling type

Heap snapshot

Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.

Allocation instrumentation on timeline

Allocation timelines show instrumented JavaScript memory allocations over time. Once profile is recorded you can select a time interval to see objects that were allocated within it and still alive by the end of recording. Use this profile type to isolate memory leaks.

Record allocation stacks (extra performance overhead)

Console

Sources

Network

Performance

Memory

Application

Security



71



be used for long running operations. It provides good approximation of allocations broken down by JavaScript execution stack.

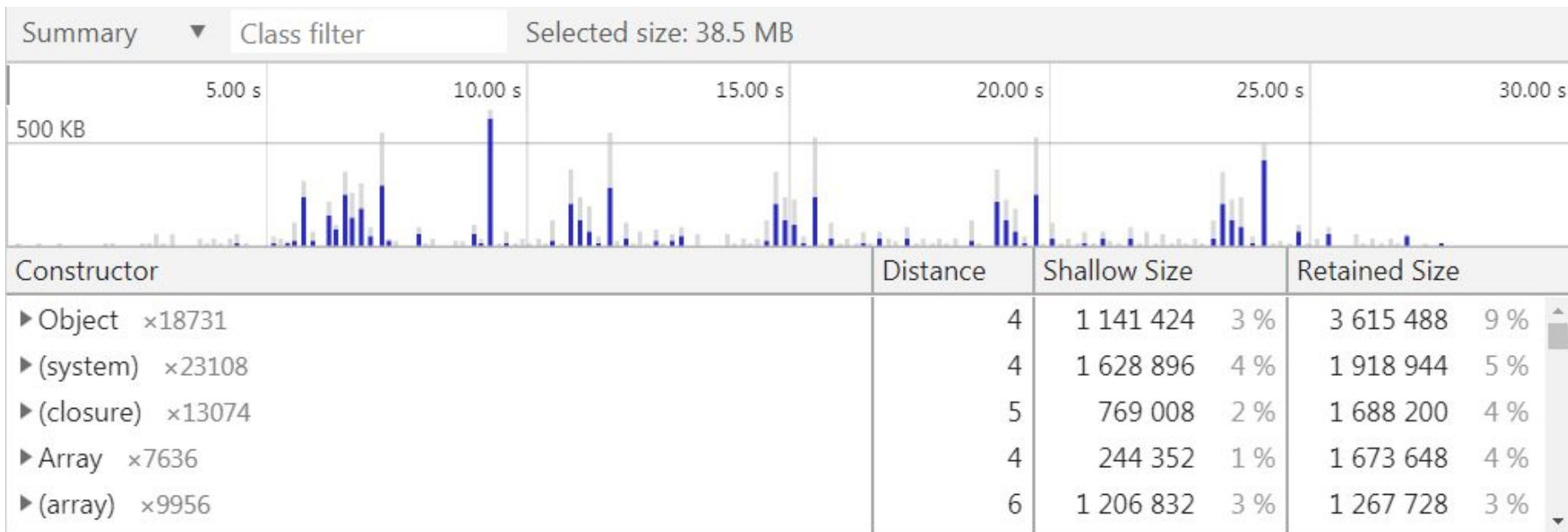
Select JavaScript VM instance

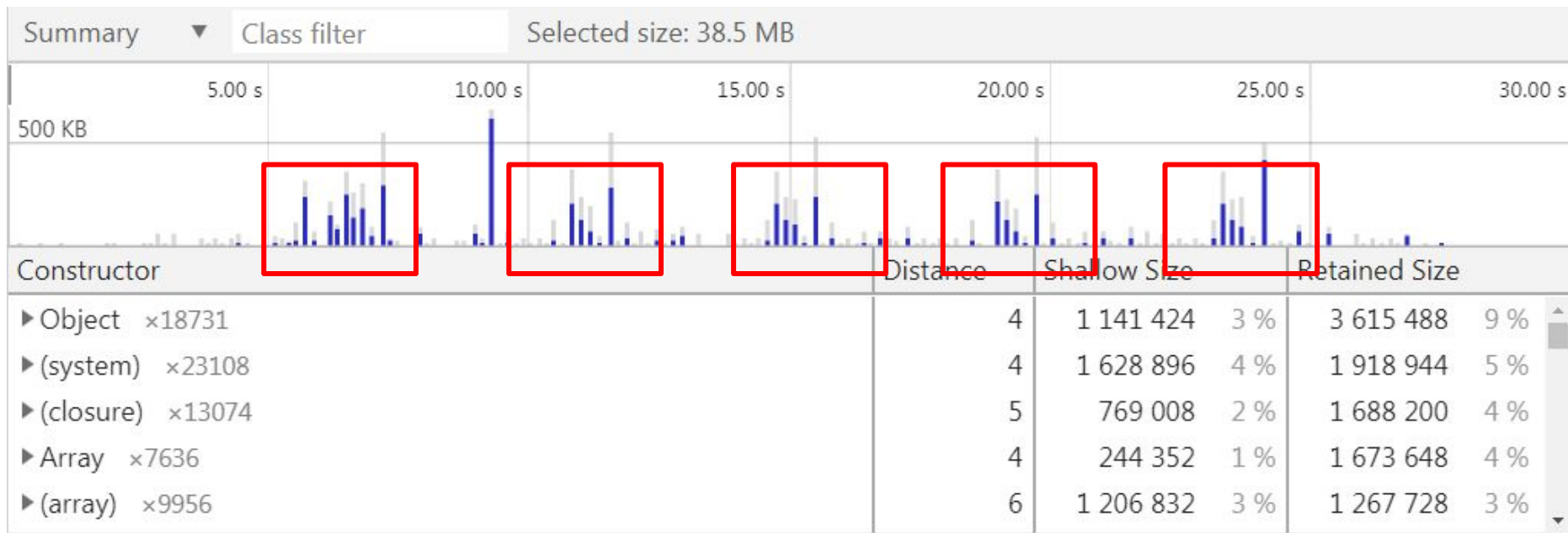
67.8 MB / 98.8 MB

online.sbis.ru

Start

Load





Summary

▼ Class filter

Selected size: 734 KB



Constructor	Distance	Shallow Size	Retained Size
▶ (system) ×2811	5	193 872 0 %	216 360 1 %
▶ r.constructor ×207	14	14 136 0 %	196 824 0 %
▶ (closure) ×2011	12	118 256 0 %	190 120 0 %
▶ Array ×955	13	30 560 0 %	182 120 0 %
▶ (array) ×1107	9	131 856 0 %	138 608 0 %
▶ system / Context ×396	13	26 400 0 %	59 184 0 %

Summary

▼ Class filter

Selected size: 734 KB



Constructor	Distance	Shallow Size	Retained Size
▶ (system) ×2811	5	193 872 0 %	216 360 1 %
▶ r.constructor ×207	14	14 136 0 %	196 824 0 %
▶ (closure) ×2011	12	118 256 0 %	190 120 0 %
▶ Array ×955	13	30 560 0 %	182 120 0 %
▶ (array) ×1107	9	131 856 0 %	138 608 0 %
▶ system / Context ×396	13	26 400 0 %	59 184 0 %

Summary

▼ Class filter

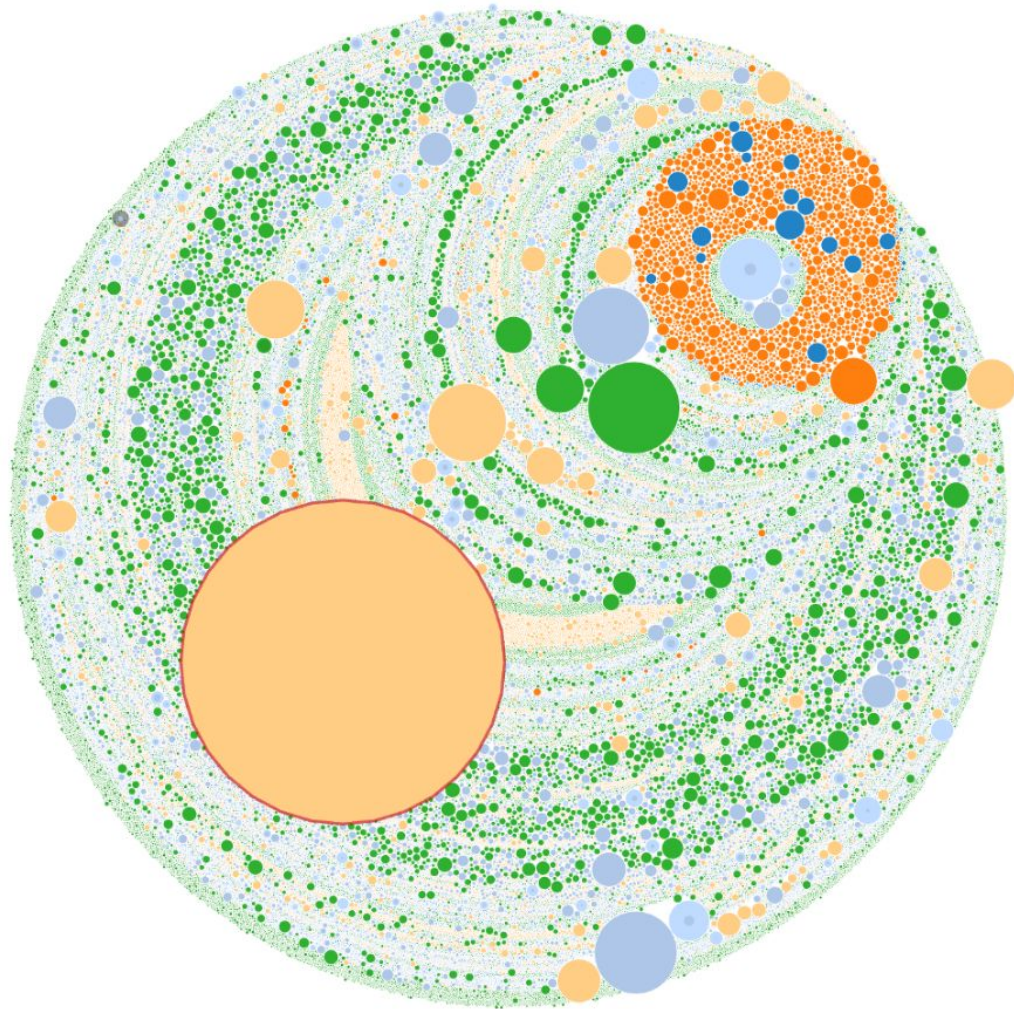
Selected size: 734 KB



Constructor	Distance	Shallow Size	Retained Size
▶ (system) ×2811	5	193 872 0 %	216 360 1 %
▶ r.constructor ×207	14	14 136 0 %	196 824 0 %
▶ (closure) ×2011	12	118 256 0 %	190 120 0 %
▶ Array ×955	13	30 560 0 %	182 120 0 %
▶ (array) ×1107	9	131 856 0 %	138 608 0 %
▶ system / Context ×396	13	26 400 0 %	59 184 0 %



<https://heapviz.com/> Tom Lagier



Инструменты

- > Диспетчер процессов Chrome
- > DevTools **Memory Allocation** и **Heap Snapshot**
- > **chrome://tracing + memory-infra**



«Автоматизация»

Сценарий

Анализ памяти

Фреймворк для автотестов

Анализ памяти

Фреймворк для автотестов

???

Фреймворк для автотестов



SeaLant*

- * **sea lantern** - морской фонарь
- sealant** - герметик

<https://github.com/saby/SeaLant>


```
from selene.driver import SeleneDriver
from selenium.webdriver import Chrome
from selene.support.conditions import have

driver = SeleneDriver.wrap(Chrome())
driver.get('http://todomvc4tasj.herokuapp.com')

def test_selene_demo():
    tasks = driver.all("#todo-list>li")
    for task_text in ["1", "2", "3"]:
        driver.element("#new-todo").set_value(task_text).press_enter()
    tasks.should(have.texts("1", "2", "3"))
```

```
from selene.driver import SeleneDriver
from selenium.webdriver import Chrome
from selene.support.conditions import have
from sealant import Sealant
```

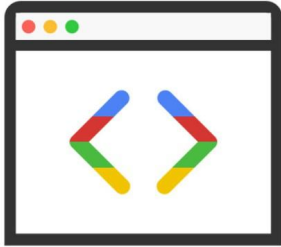
```
driver = SeleneDriver.wrap(Chrome())
driver.get("http://todomvc4tasj.herokuapp.com")
sl = Sealant("localhost", 9221)
```

```
@sl.memory_leak()
```

```
def test_selene_demo():
    tasks = driver.all("#todo-list>li")
    for task_text in ["1", "2", "3"]:
        driver.element("#new-todo").set_value(task_text).press_enter()
    tasks.should(have.texts("1", "2", "3"))
```

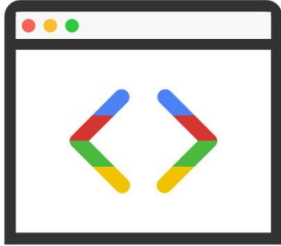
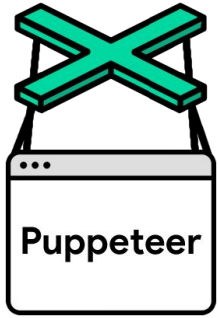
Утечка памяти: +673 КБ

- > DOM nodes: +1033
- > event listeners: -2
- > local.storage: 0
- > http cache: +4303
- > window.config: +10



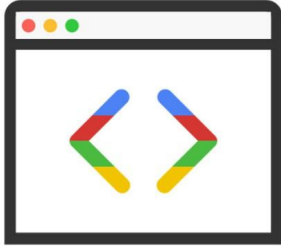
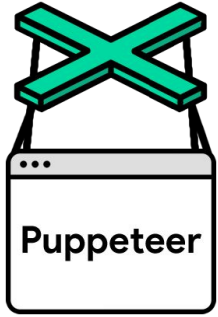
Chrome Devtools
Protocol



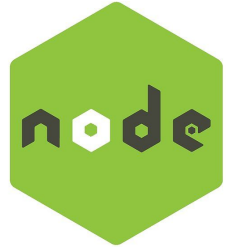


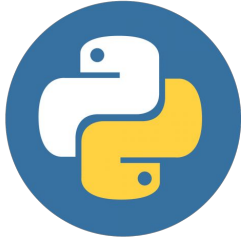
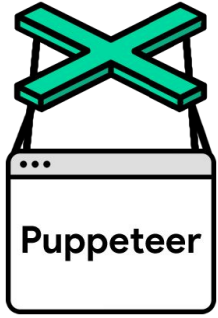
Chrome Devtools
Protocol



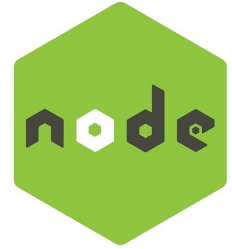


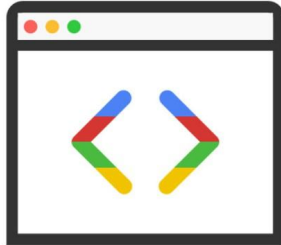
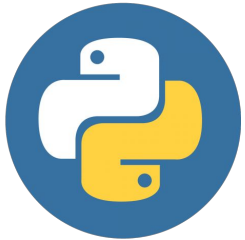
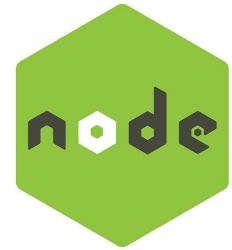
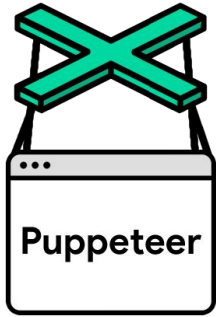
Chrome Devtools
Protocol





Chrome Devtools
Protocol



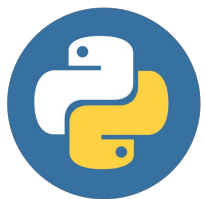


Chrome Devtools
Protocol

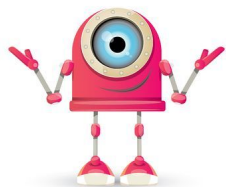


<https://chromedevtools.github.io/devtools-protocol/1-3/>





Тест

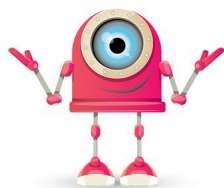


Фреймворк





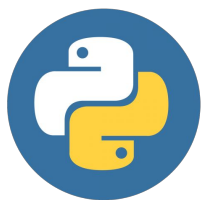
Тест



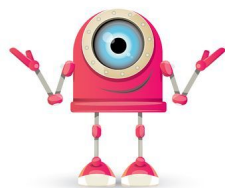
Фреймворк



Chrome 63 - много клиентов при отладке



Тест



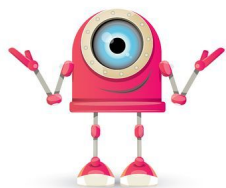
Фреймворк



SeaLant



Тест



Фреймворк



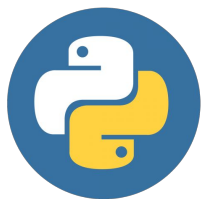
CDP



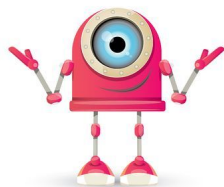
хост:порт



SeaLant



Тест



Фреймворк



CDP



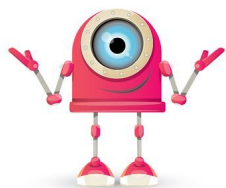
SeaLant

`HeapProfiler.enable`





Тест



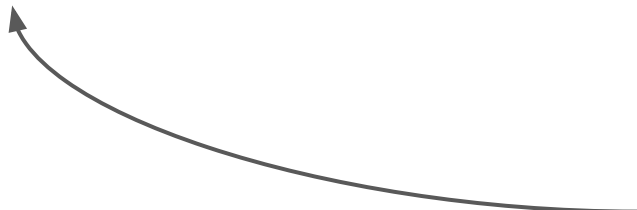
Фреймворк



CDP

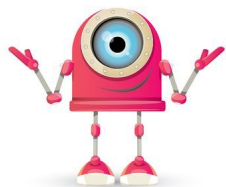


SeaLant





Тест



Фреймворк



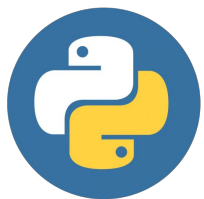
CDP



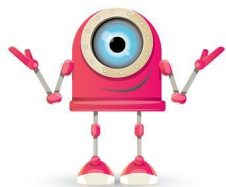
SeaLant

`HeapProfiler.collectGarbage`





Тест



Фреймворк



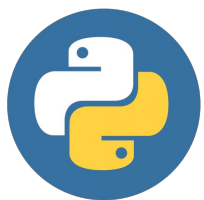
CDP



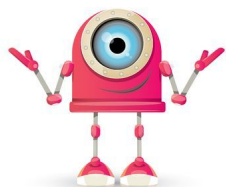
SeaLant

`HeapProfiler.takeHeapSnapshot`





Тест



Фреймворк

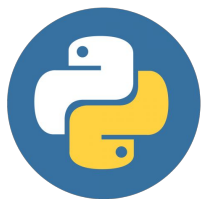


CDP

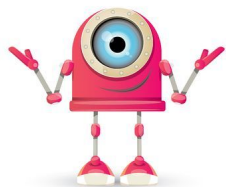


chunks

SeaLant



Тест



Фреймворк



CDP

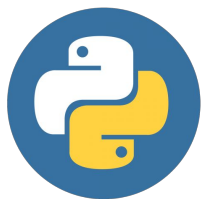


chunks

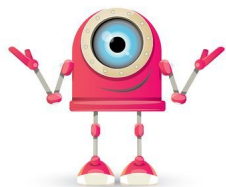
SeaLant



Snapshot



Тест



Фреймворк



Объём памяти

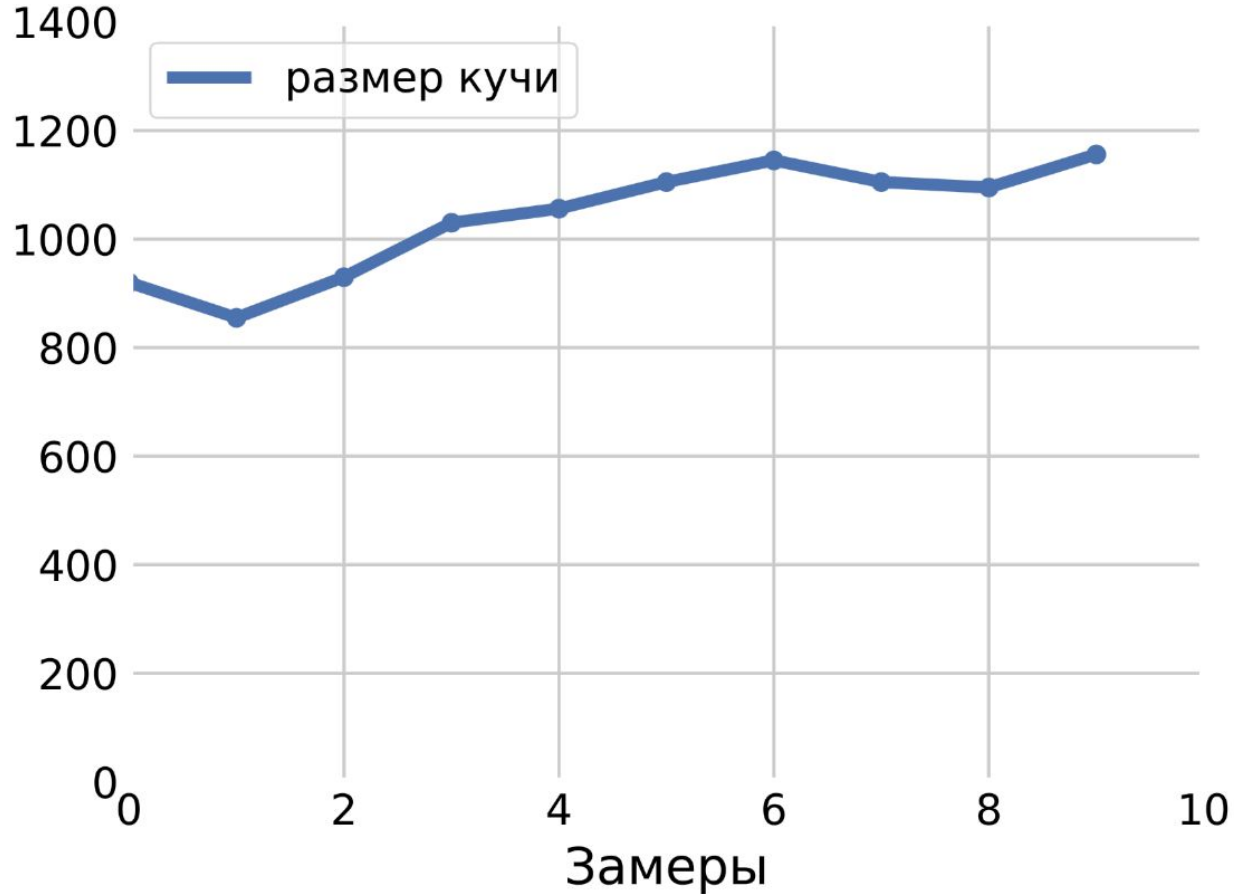


SeaLant

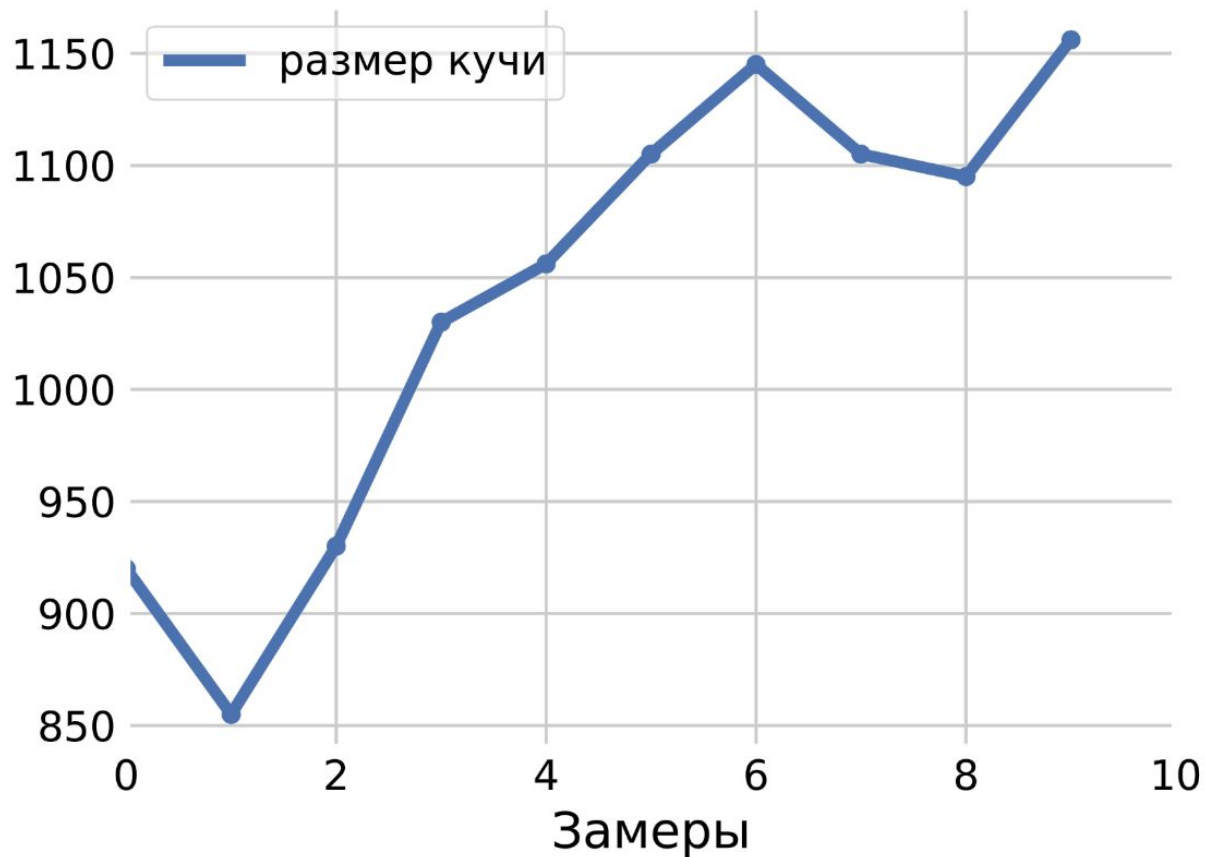


Snapshot

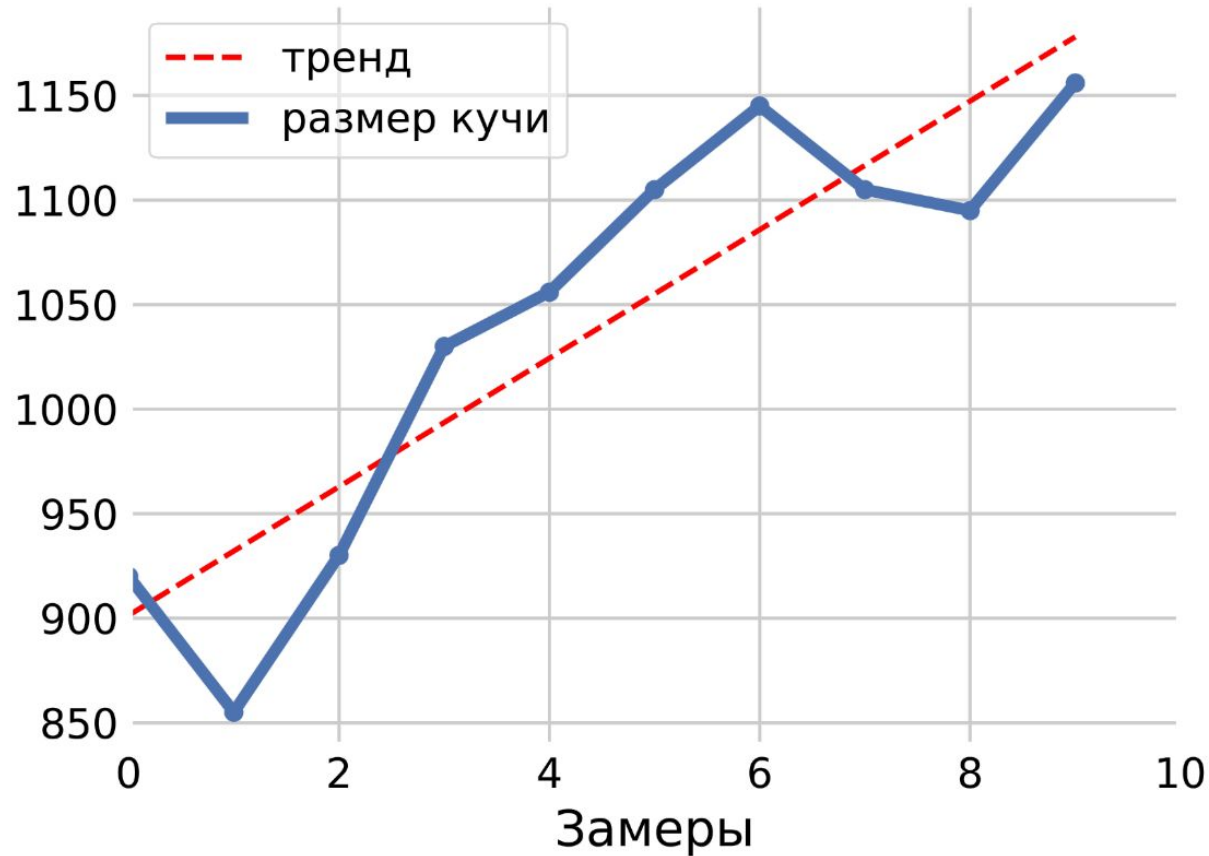
Размер кучи от замера к замеру, КБ



Размер кучи от замера к замеру, КБ



Размер кучи от замера к замеру, КБ



```
from selene.driver import SeleneDriver
from selenium.webdriver import Chrome
from selene.support.conditions import have
from sealant import Sealant
```

```
driver = SeleneDriver.wrap(Chrome())
driver.get("http://todomvc4tasj.herokuapp.com")
sl = Sealant("localhost", 9221)
```

```
@sl.memory_leak()
def test_selene_demo():
    tasks = driver.all("#todo-list>li")
    for task_text in ["1", "2", "3"]:
        driver.element("#new-todo").set_value(task_text).press_enter()
    tasks.should(have.texts("1", "2", "3"))
```

Утечка памяти: +673 КБ

- > DOM nodes: +1033
- > event listeners: -2
- > local.storage: 0
- > http cache: +4303
- > window.config: +10



Теперь мы вооружены!



«Покрытие»

«Течь» может любой сценарий

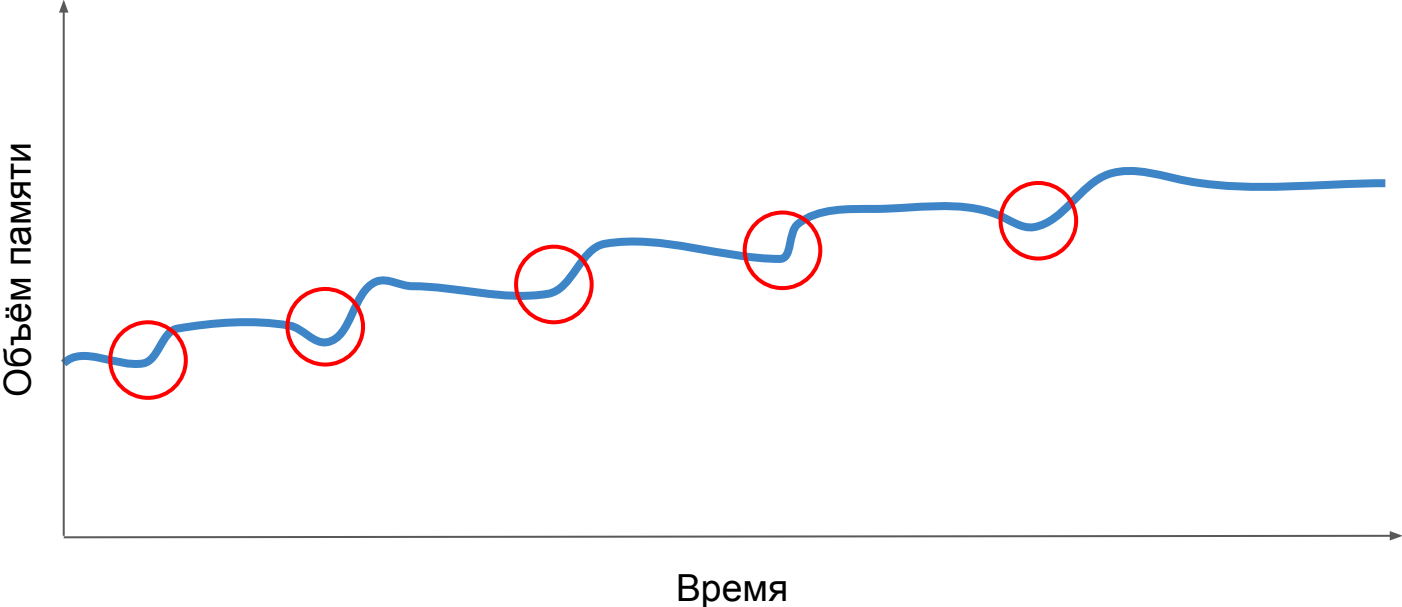


Сценарии утечек

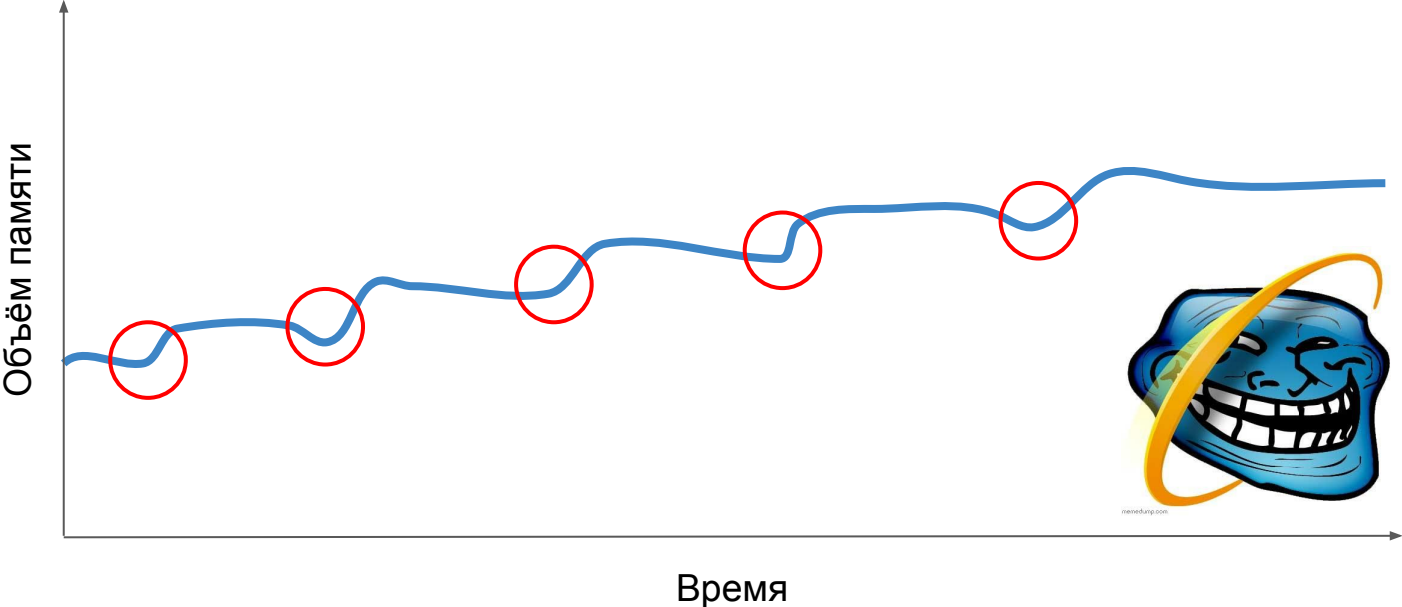
Сценарии утечек

> “Ничего не поделаешь”

microsoft.com, IE 10



microsoft.com, IE 10









Целевое действие + обратное действие

Сценарии утечек

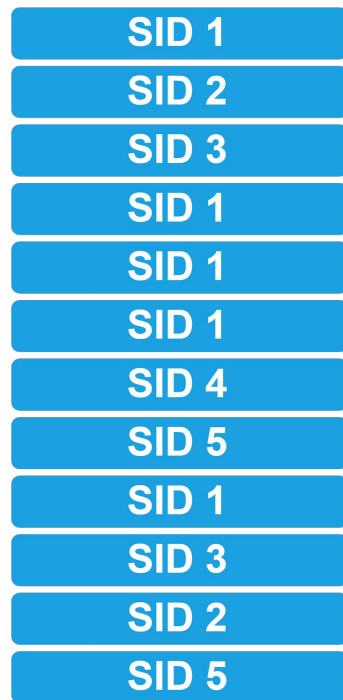
- > “Ничего не поделаешь”
- > Таймеры и события

Сценарии утечек

- > “Ничего не поделаешь”
- > Таймеры и события
- > Выполняемые действия

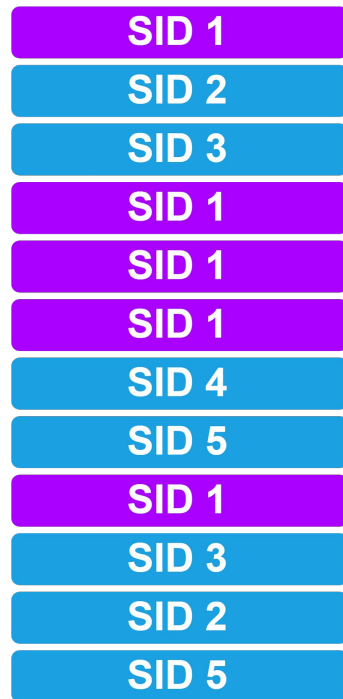
Циклический сценарий - последовательность действий, которая много раз повторяется без перезагрузки страницы

Фильтруем по сессии

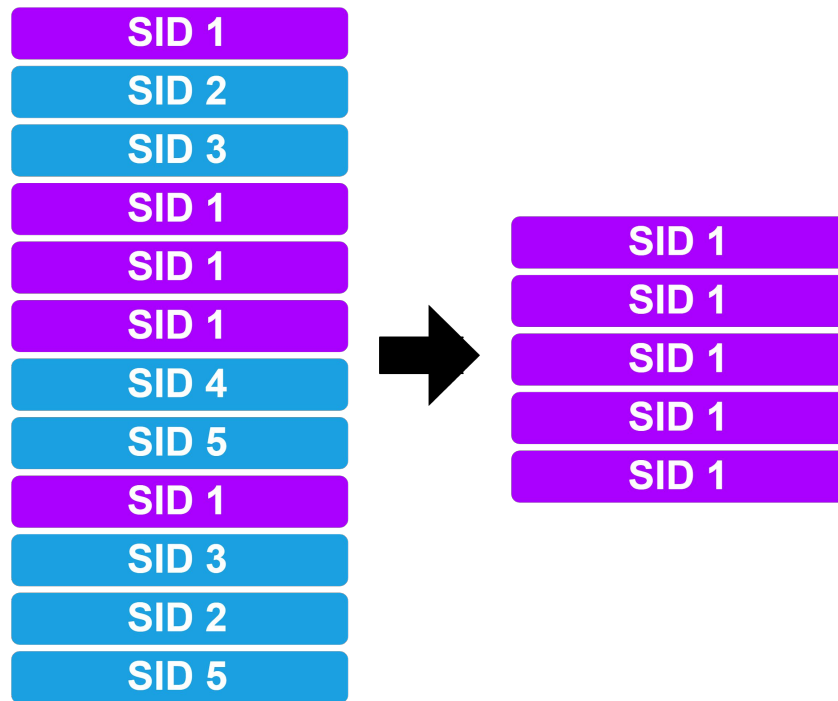


Запросы из
access.log

Фильтруем по сессии



Фильтруем по сессии



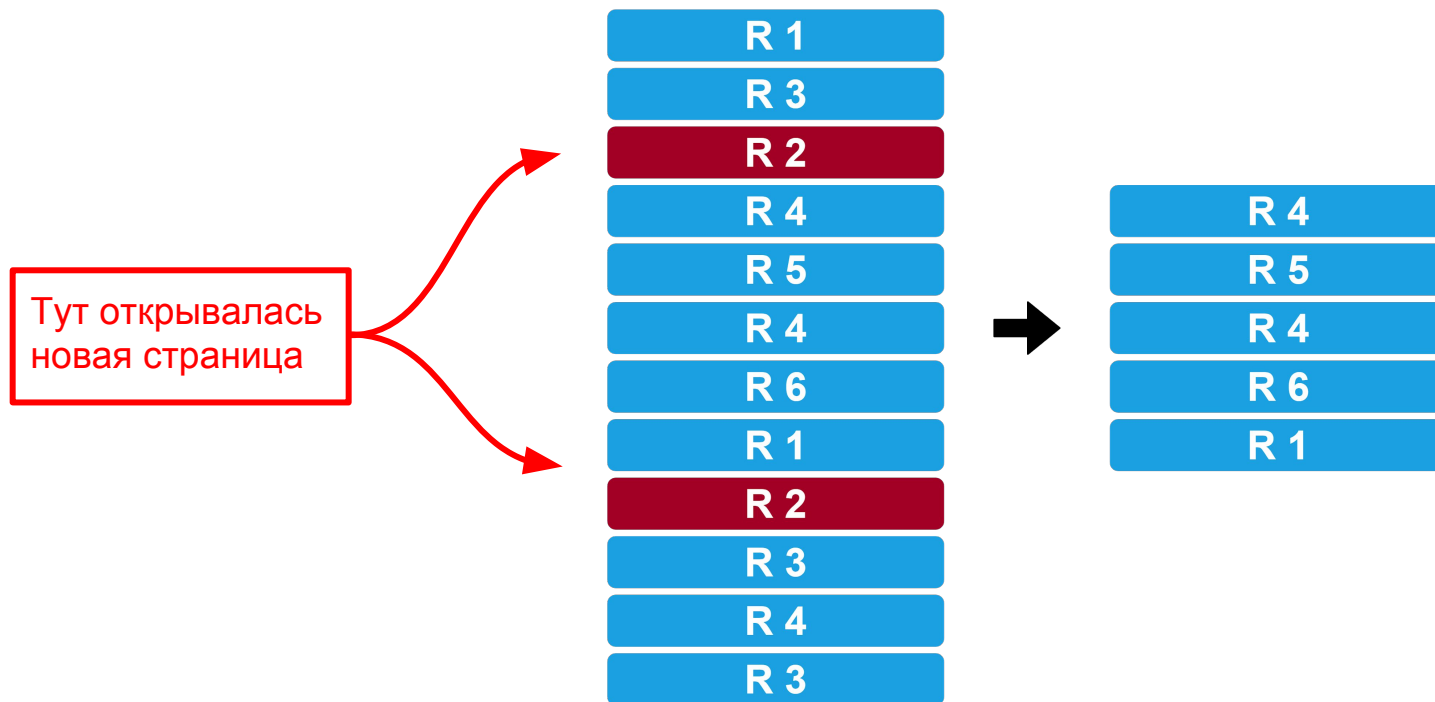
Находим группу запросов с одной страницы



Находим группу запросов с одной страницы



Находим группу запросов с одной страницы



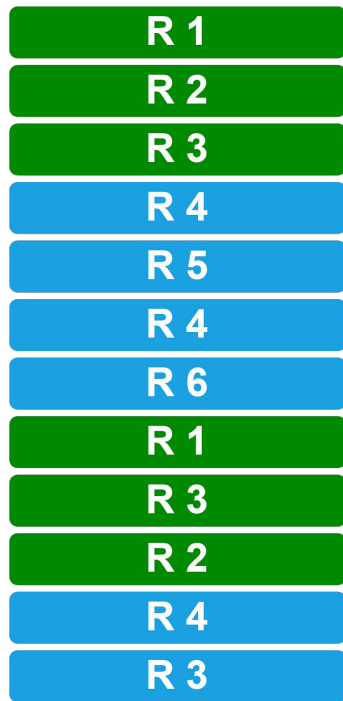
Выделяем повторяющиеся группы запросов



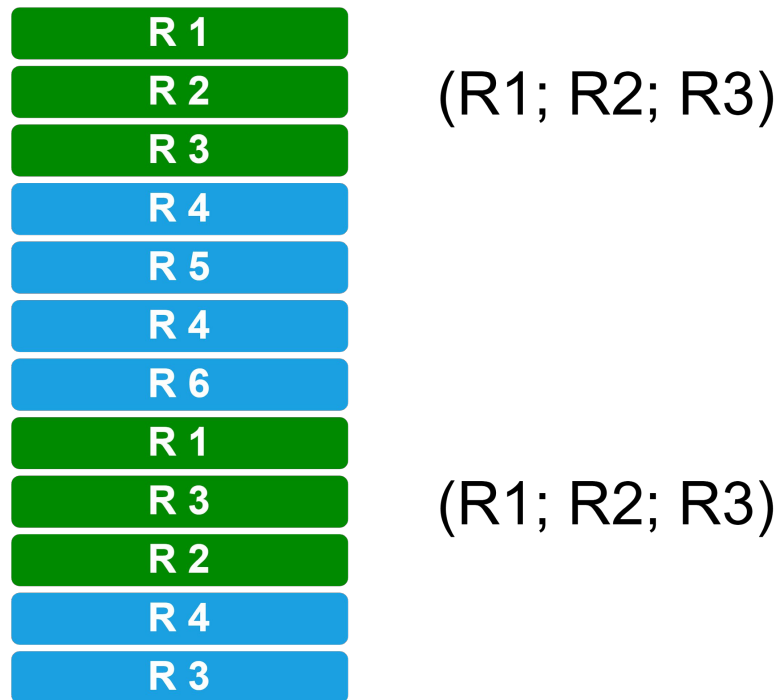
**Аннотированные суффиксные
деревья: особенности
реализации**

Михаил Дубов, Екатерина Черняк

Выделяем повторяющиеся группы запросов



Выделяем повторяющиеся группы запросов



Реверсинжиниринг

<https://online.sbis.ru/carry.html>

R1 = Payment.GetLastError

R2 = Sale.Pay

R3 = Payment.Carry

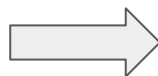
Реверсинжиниринг

<https://online.sbis.ru/carry.html>

R1 = Payment.GetLastError

R2 = Sale.Pay

R3 = Payment.Carry



«Оплата товара в кассе»

Итого

Итого

> Сотни тестов в СИ

Итого

- > Сотни тестов в CI
- > Находимые ошибки с 0 до десятков за версию

Итого

- > Сотни тестов в CI
- > Находимые ошибки с 0 до десятков за версию
- > Автоматизация анализа серверного JS за 2 дня



**Проверил своё
приложение на утечки
памяти?**

> Забираем **SeaLant**:

<http://github.com/saby/sealant/>

> Находим **циклические сценарии**

> Пишем **автотесты**



Докучаев Сергей
Тензор
t.me/m0rasan

