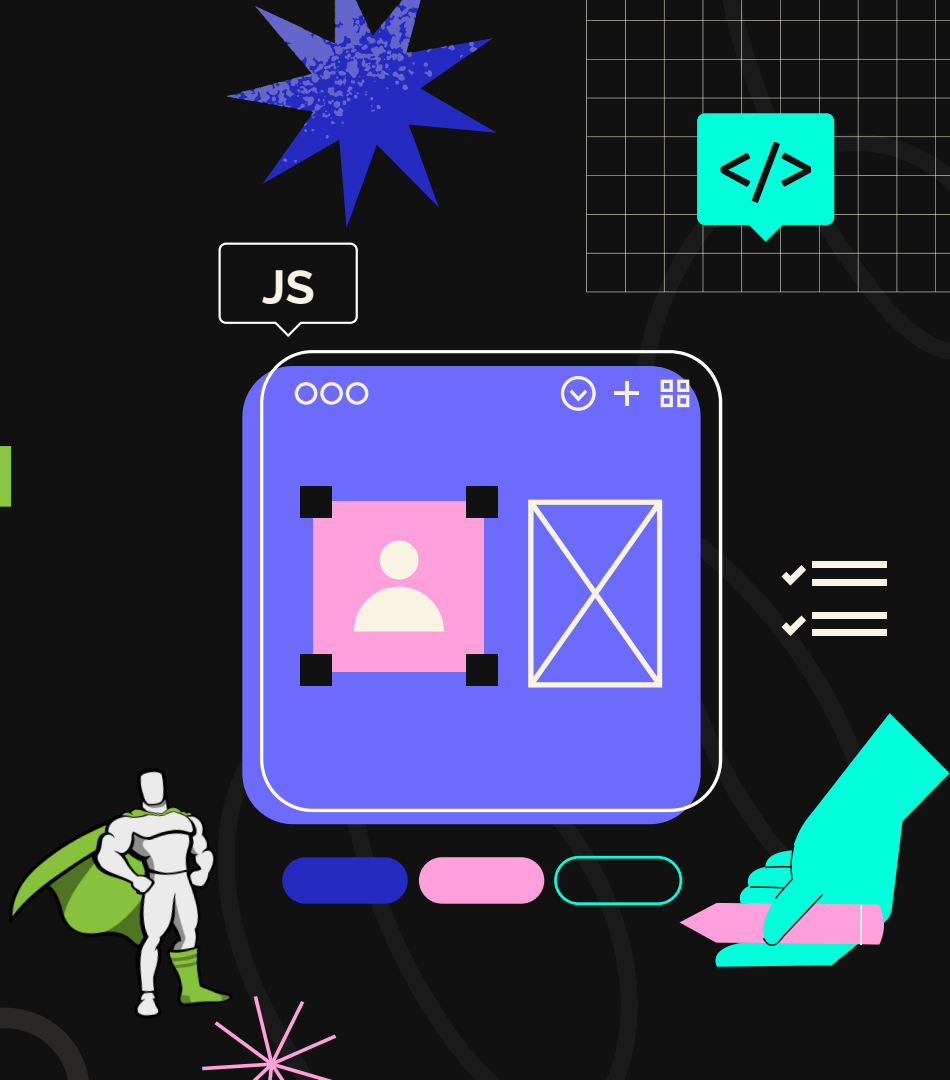
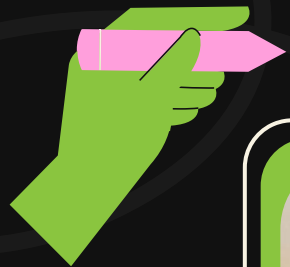


Как GSAP ukrал спектакль?



sports™



000



01

КТО Я?



СИРЕНА



Спецпроекты

DEMIX x sports.ru

7 ГЛОССОВ

#5

СТАРТОВЫЙ
5
НОМЕР

Эйс

2 года Возраст

2 года Беговой стаж

любимая дистанция

FONBET x sports.ru

РАУНД 3. ПАНОРАМА

6

Ваша подсказка

Ранжирование:

- 1 | Ранж. 1: Фонд
- 2 | Ранж. 2: Фонд
- 3 | Ранж. 3: Панорама

Забрать до 10000 р

Кибер

185 188

Очки за тур

Среднее у всех

Результаты за 8 тур

Регистрируйся в BetBoom

Фрибет 1000 р

Фантази PGL Major Copenhagen 2024 CS 2 в разгаре

Забрать

САМОКАТ | sports.ru

Геркулес
Наставник

Вы в Самокатии – городе, где проходят Самокатные игры.

Далее

FONBET x sports.ru

КАРЬЕРА

Сезааааа

3В

здоровая горна

«Байер», «Андерлехт», «Фортуна» и «Аякс» — кто из российских игроков выступал за эти клубы?

Сергей Юран

Дмитрий Селес

Сергей Кирьяков

Дмитрий Булыкин




GSAP

Menu 

Animate anything

W.
Site of the Day

{ GSAP - A wildly robust JavaScript animation library built for professionals }

Get GSAP 



Что рассмотрим?

01

Зачем GSAP?

02

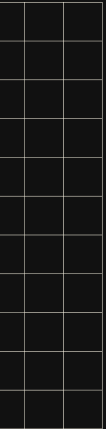
Немного кишков

03

Бенчмарки

04

Как GSAP помог
мне?



Зачем GSAP?

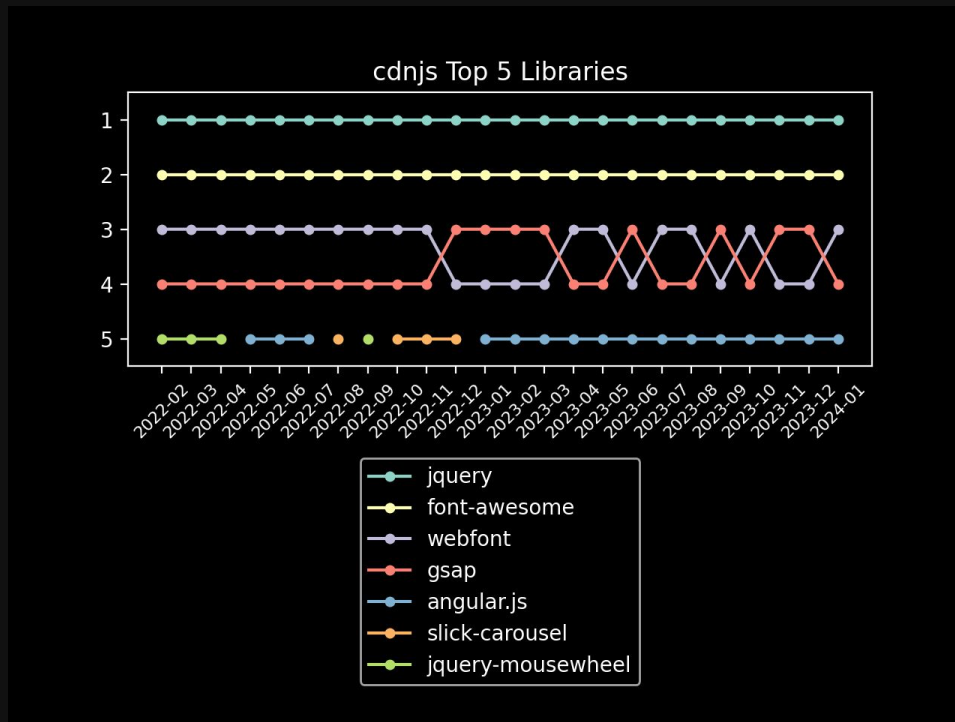
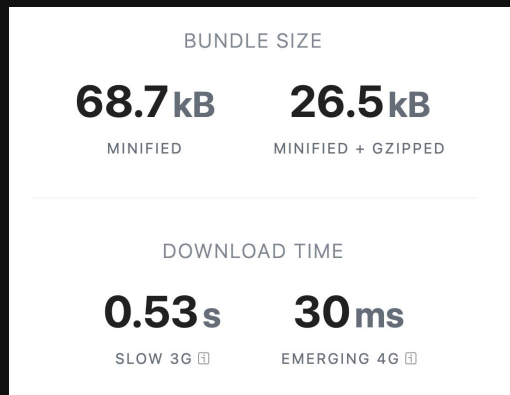
- Кроссплатформенность и совместимость
- Производительность
- Независимость от стилей
- Расширяемость и плагины
- Удобство разработки
- Упрощение будущих изменений
- Стандартизация API для анимации.



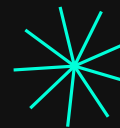
GSAP же тяжелый?



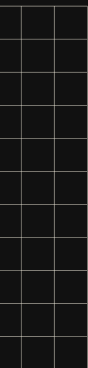
Ну да, но вообще-то нет



Альтернативы



Библиотека	Популярность	Размер (Кб)	Последнее обновление	Лицензия
GSAP	449к	26.5	2 months ago	CUSTOM
Animejs	196к	6.9	3 months ago	MIT
Framer-motion	355к	42.4	5 days ago	MIT
motion	265к	9.7	2 months ago	MIT
popmotion	1527к	9.9	2 years ago	MIT
react-motion	518к	4.8	6 years ago	MIT
react-spring	794к	19.4	5 months ago	MIT
velocity-js	526к	14.8	2 years ago	MIT

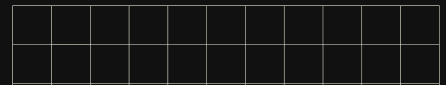
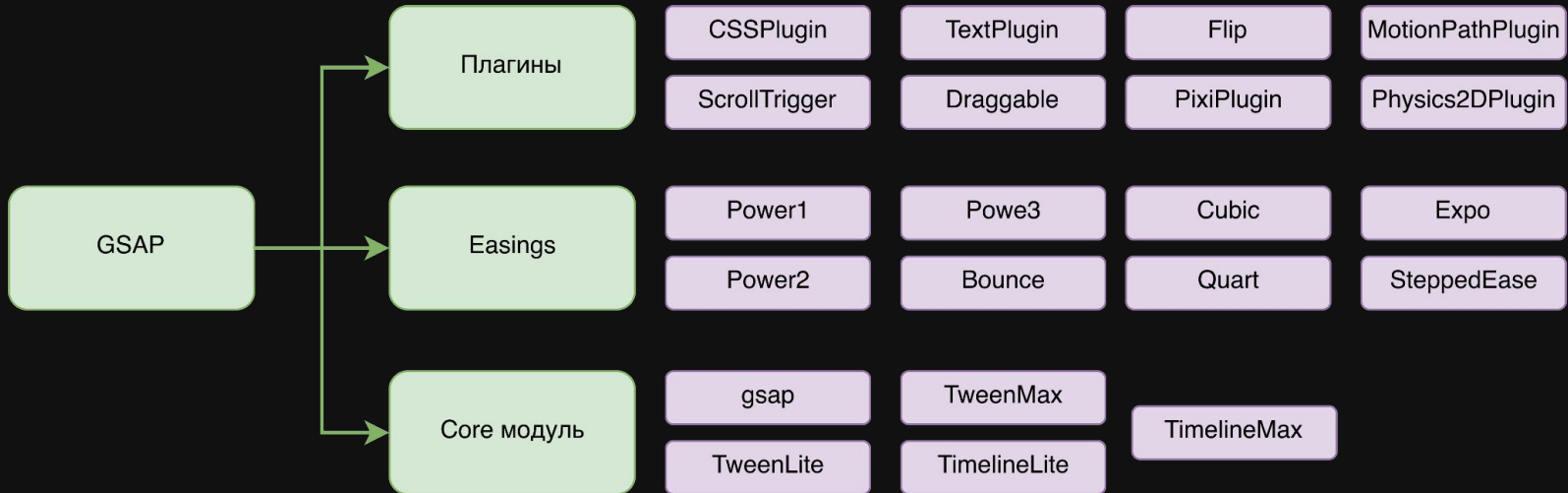
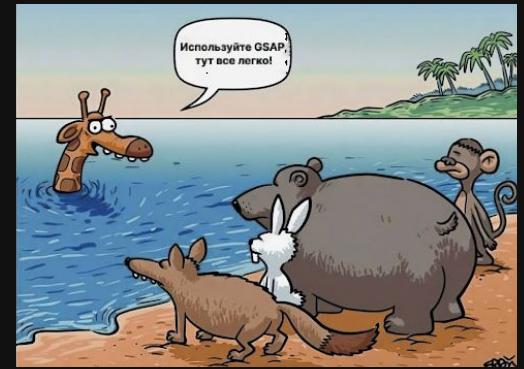


Возможности

Функция/Плагин	GSAP	Другие
Анимация по заданному пути (Motion Path)	✓	✓
Интеграция анимаций WebGL (Pixi.js)	✓	✗
Создание перетаскиваемых элементов (Draggable)	✓	✗
Flip анимации	✓	✗
Слушатель жестов пользователя (Observer)	✓	✗
Анимированная прокрутка до элемента (ScrollTo)	✓	✗
Анимация при скролле страницы (ScrollTrigger)	✓	✓
Манипуляции и анимация текста (Text)	✓	✗
Анимация рисования SVG (DrawSVG)	✓	✓
2D и свойства физики (Physics)	✓	✗
Анимация текста со смешиванием символов (ScrambleText)	✓	✗
Инструменты для разработчиков анимаций (GSDevTools)	✓	✗
Анимация с эффектом инерции (Inertia)	✓	✗
Морфинг SVG фигур (MorphSVG)	✓	✓
Вспомогательный инструмент для Motion Path (MotionPathHelper)	✓	✗
Плавная прокрутка с анимациями (ScrollSmoother)	✓	✗
Анимация разделения текста (SplitText)	✓	✗

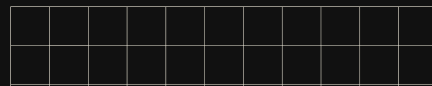
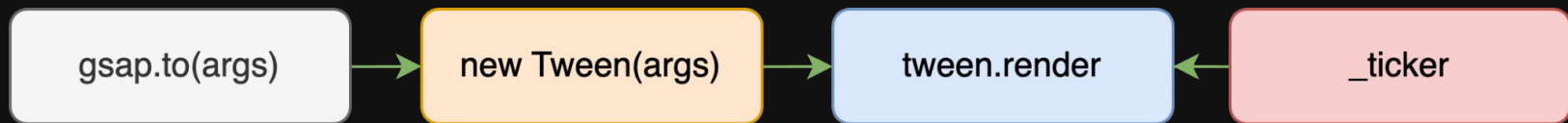


Структура GSAP



gsap.to?

Полный цикл создания, управления и воспроизведения анимаций в GSAP.
(вот он справа налево)



Класс GSCache 1/2

```
const boxesContainer = ref( value: null);

const iterateWithRead = (read) => {
  const startTime = Date.now();
  Array.from(boxesContainer.value.children).forEach((div) => {
    const computedStyle = window.getComputedStyle(div);
    if (read) {
      computedStyle.transform;
    }
  });
  timeTaken.value = Date.now() - startTime;
};
```

Mode: With reading transform
result - 129



Mode: Without reading transform
result - 24



Класс GSCache 2/2

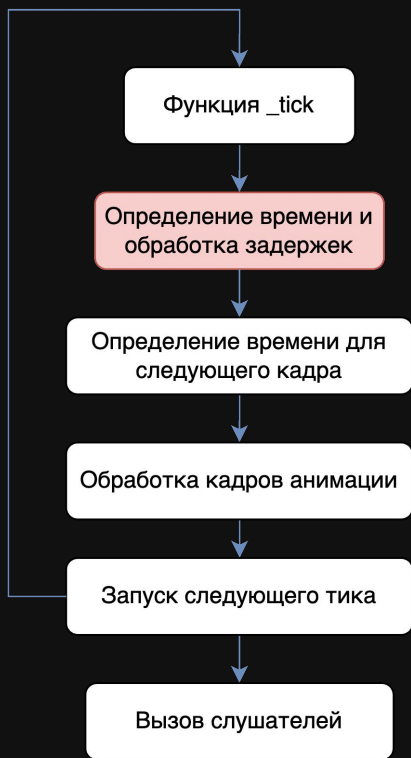
```
export class GSCache {  
  
  constructor(target, harness) {  
    this.id = _gsID++;  
    target._gsap = this;  
    this.target = target;  
    this.harness = harness;  
    this.get = harness ? harness.get : _getProperty;  
    this.set = harness ? harness.getSetter : _getSetter;  
  }  
  
}
```

```
gsap-core.js?t=1710276093426:1153  
▼ GSCache {id: 3, target: div.test, harness: f, get: f, set:  
  f, ...} i  
  force3D: "auto"  
  ▶ get: (target, property, unit, uncache) => {...}  
  ▼ harness: f ()  
    ▶ aliases: {autoAlpha: 'opacity,visibility', scale: 'scal  
    ▶ core: {_removeProperty: f, _getMatrix: f}  
    ▶ get: (target, property, unit, uncache) => {...}  
    ▶ getSetter: f getSetter(target, property, plugin)  
      prop: "css"  
    ▶ register: () => {...}  
    ▶ targetTest: f targetTest(target)  
      length: 0  
      name: ""  
    ▶ prototype: {init: f, render: f, add: f, kill: f, modifi  
      arguments: (...)  
      caller: (...)  
      [[FunctionLocation]]: gsap-core.js?t=1710276093426:669  
    ▶ [[Prototype]]: f ()  
    ▶ [[Scopes]]: Scopes[3]  
    id: 3  
  ▶ renderTransform: f (ratio, cache)  
    rotation: "45deg"  
    rotationX: "0deg"  
    rotationY: "0deg"  
    scaleX: 1  
    scaleY: 1  
  ▶ set: f getSetter(target, property, plugin)  
    skewX: "0deg"  
    skewY: "0deg"
```

Ленивый рендеринг

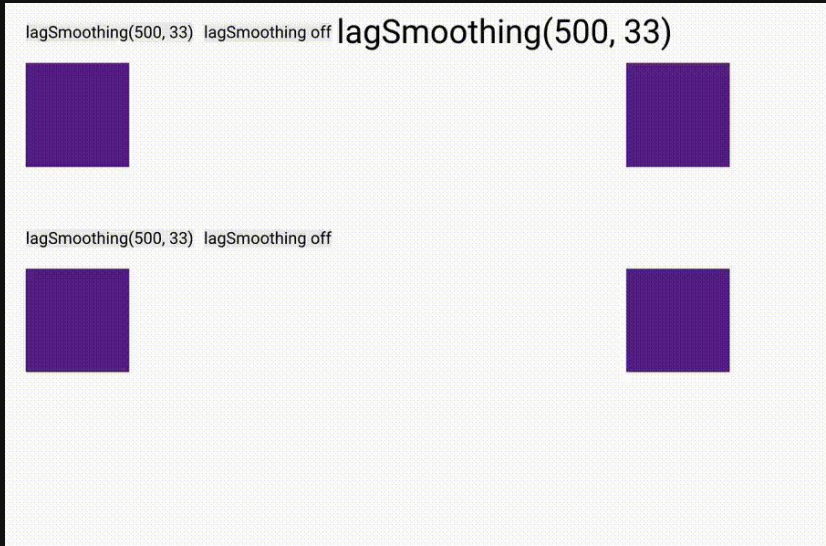
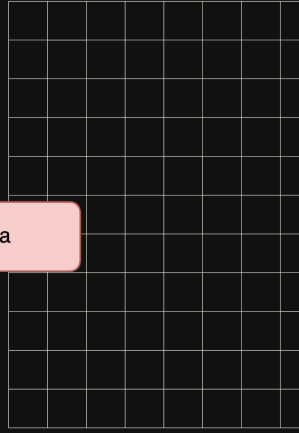
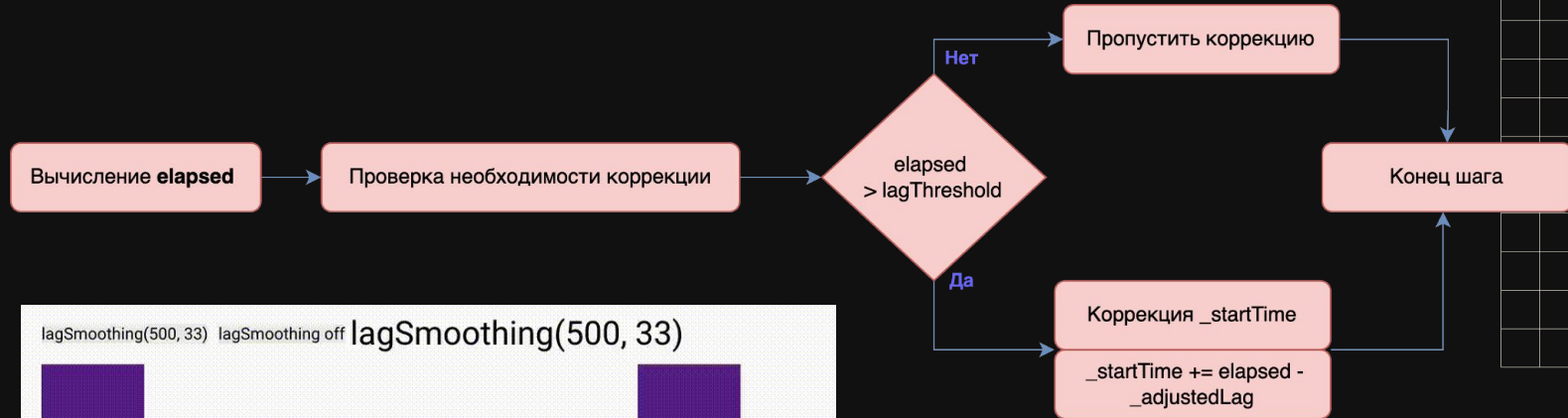
- Анимация с задержкой (delay)
- Нестандартная передача параметров
 - `gsap.to('.element', { x: () => Math.random() * 100, duration: 1 });`
 - `gsap.to('.element', { x: 'random(-100, 100, 5)' duration: 1 });`
 - `gsap.to('.element', { x: '+=10px' duration: 1 });`
- Использование stagger
- ScrollTrigger
- Анимация таймлайнов
- Ключевые кадры (keyframes)
- Параметр lazy





```
_tick = v => {  
  let elapsed : number = _getTime() - _lastUpdate,  
      manual : boolean = v === true,  
      overlap, dispatch, time, frame;  
  (elapsed > _lagThreshold || elapsed < 0) && (_startTime += elapsed - _adjustedLag);  
  _lastUpdate += elapsed;  
  time = _lastUpdate - _startTime;  
  overlap = time - _nextTime;  
  if (overlap > 0 || manual) {  
    frame = ++_self.frame;  
    _delta = time - _self.time * 1000;  
    _self.time = time = time / 1000;  
    _nextTime += overlap + (overlap >= _gap ? 4 : _gap - overlap);  
    dispatch = 1;  
  }  
  manual || (_id = _req(_tick));  
  if (dispatch) {  
    for (_i = 0; _i < _listeners.length; _i++) {  
      _listeners[_i](time, _delta, frame, v);  
    }  
  }  
};
```

lagSmoothing




```
_renderRoundedCSSProp = (ratio, data) : void => {  
  let value = data.s + data.c * ratio;  
  data.set(data.t, data.p, ~~(value + (value < 0 ? -.5 : .5)) + data.u, data);  
},
```

```
_round = value => Math.round( x: value * 100000) / 100000 || 0,  
_roundPrecise = value => Math.round( x: value * 10000000) / 10000000 || 0, //
```



Тип бенчмарка	FPS	Memory (mb)	CPU (ms)	Longest Frame Time (ms)
С округлением	39.2	141	10.4	360.5
Без округления	35.6	142	10.4	366.7

sports' Перевод в числовые значения

```
_numericIfPossible = value => {  
  let n : number = parseFloat(value);  
  return (n || n === 0)  
  && (value + "").match(_delimitedValueExp).length < 2 ?  
  n : _isString(value) ? value.trim() : value;  
},
```

```
const TEST_DATA = {  
  FIRST: 0,  
  SECOND: 1,  
};
```

```
const TEST_DATA = {  
  FIRST: 'FIRST',  
  SECOND: 'SECOND',  
};
```

```
let result = 0;  
for (let i = 0; i < REPETITIONS; i++) {  
  const value = i % 2 === 0  
    ? TEST_DATA.FIRST : TEST_DATA.SECOND;  
  if (value === TEST_DATA.FIRST) result += 1;  
}
```

В среднем на 20 процентов,
лучше :)



Force3d: "auto"

https://developer.mozilla.org/en-US/docs/Learn/Performance/CSS#animating_on_the_gpu

```
use3D : boolean = (force3D === "auto" && ratio && ratio !== 1) || force3D === true;
```



Honorable mentions

- Оптимизация сокращения чисел через побитовые операции
- Эффективность вычислений путём исключения операции извлечения корня
- Очистка памяти
- Оптимизация рендеринга: отображение только видимых элементов
- Группирование процессов обновления для повышения производительности
- Много условий (`_renderZeroDurationTween`)
- НЕ помешанность на оптимизациях



Совместимость. _tempDiv

```
_tempDiv = _createElement( type: "div" ) || {style:{}};  
_tempDiv.style.cssText = "border-width:0;line-height:0;position:absolute;padding:0";
```

- Используется для проверки поддержки определенных CSS свойств браузером, включая те, что требуют вендорных префиксов.
- Помогает в определении возможностей браузера, таких как поддержка 3D-трансформаций (perspective).
- Конвертация и измерение значений в px (_convertToUnit)

```
_supports3D = !!_checkPropPrefix( property: "perspective");
```

```
style[horizontal ? "width" : "height"] = amount + (toPixels ? curUnit : unit);  
parent.appendChild(_tempDiv);  
px = _tempDiv[measureProperty];  
parent.removeChild(_tempDiv);
```

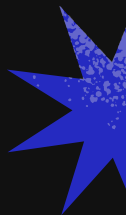
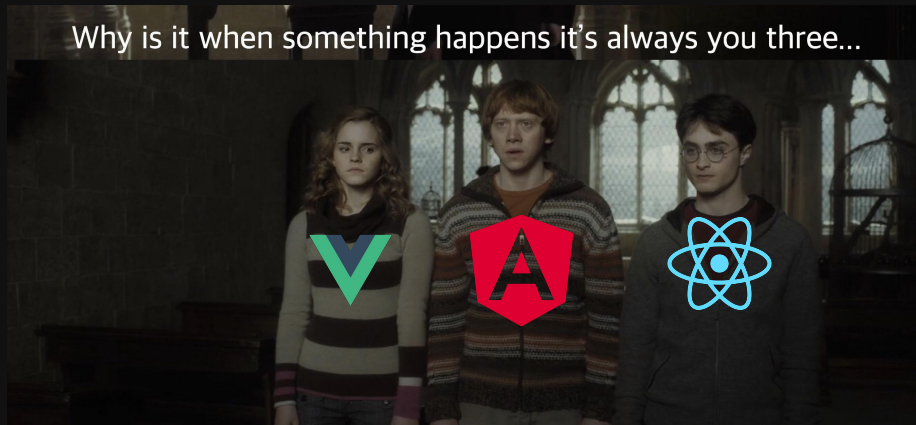
Магия совместимости, но какой ценой?

Количество вхождений по ключевым словам:

1. Firefox - 29
2. Safari - 22
3. IOS - 17
4. Android - 10
5. IE - 8
6. Chrome - 4
7. Vue - 4
8. React - 1
9. Angular - 1

```
_preventDefault = event => {  
  event.preventDefault && event.preventDefault();  
  event.preventManipulation && event.preventManipulation(); //for some Microsoft browsers  
},
```

Why is it when something happens it's always you three...



getBboxHack

1. Создание временного SVG-элемента
2. Перемещение исходного элемента
3. Изменение стиля для отображения
4. Попытка получить BBox
5. Восстановление исходного состояния
6. Возвращение BBox



```
_getBBoxHack = function(swapIfPossible) {  
  let svg = _createElement( type: "svg", ns: (this.ownerSVGElement &&  
    oldParent = this.parentNode,  
    oldSibling = this.nextSibling,  
    oldCSS = this.style.cssText,  
    bbox;  
  _docElement.appendChild(svg);  
  svg.appendChild(this);  
  this.style.display = "block";  
  if (swapIfPossible) {  
    try {  
      bbox = this.getBBox();  
      this._gsapBBox = this.getBBox(); //store the original  
      this.getBBox = _getBBoxHack;  
    } catch (e) { }  
  } else if (this._gsapBBox) {  
    bbox = this._gsapBBox();  
  }  
  if (oldParent) {  
    if (oldSibling) {  
      oldParent.insertBefore(this, oldSibling);  
    } else {  
      oldParent.appendChild(this);  
    }  
  }  
  _docElement.removeChild(svg);  
  this.style.cssText = oldCSS;  
  return bbox;  
},
```


sports'

Совместимость везде, даже в плагинах

```
$observer.create({  
  type: 'wheel,touch,pointer',  
  wheelSpeed: -1,  
  onDown: scrollToPrev,  
  onUp: scrollToNext,  
  tolerance: isMobile.value ? 10 : 150,  
  lockAxis: true,  
  axis: 'y'  
});
```

Observers:



Benchmark



<https://gsap.com/js/speed.html>



Benchmark

FPS

Кадры в секунду

Memory

Средний объем
памяти

CPU

Время в секундах,
которое ЦП тратит на
выполнение задачи

LFT

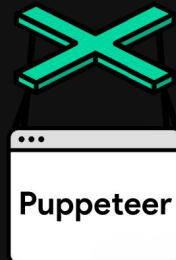
Максимальное время,
затраченное на
отрисовку одного кадра.

Divs

```
x: 200,  
ease: 'none',  
duration: 20,  
scale: 2,  
opacity: 0.5,  
stagger: 0.01,
```

SVG

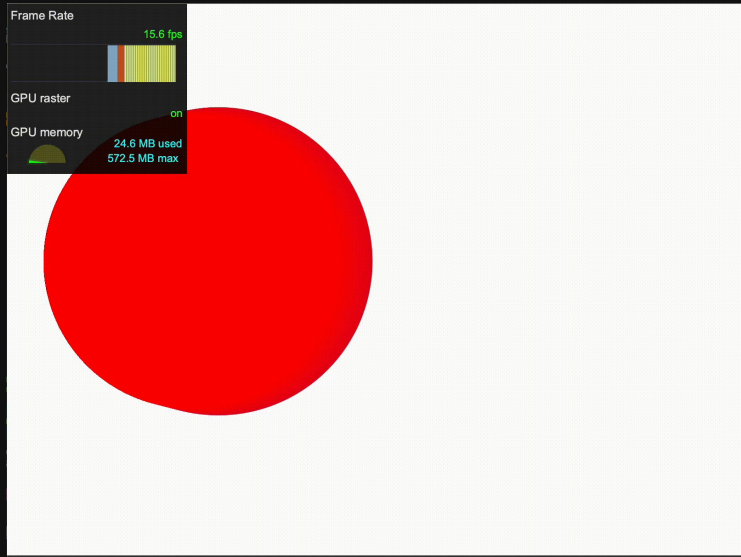
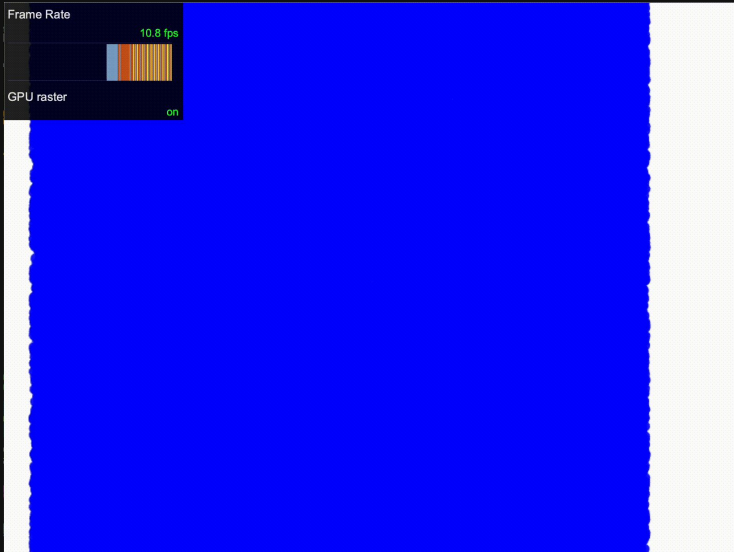
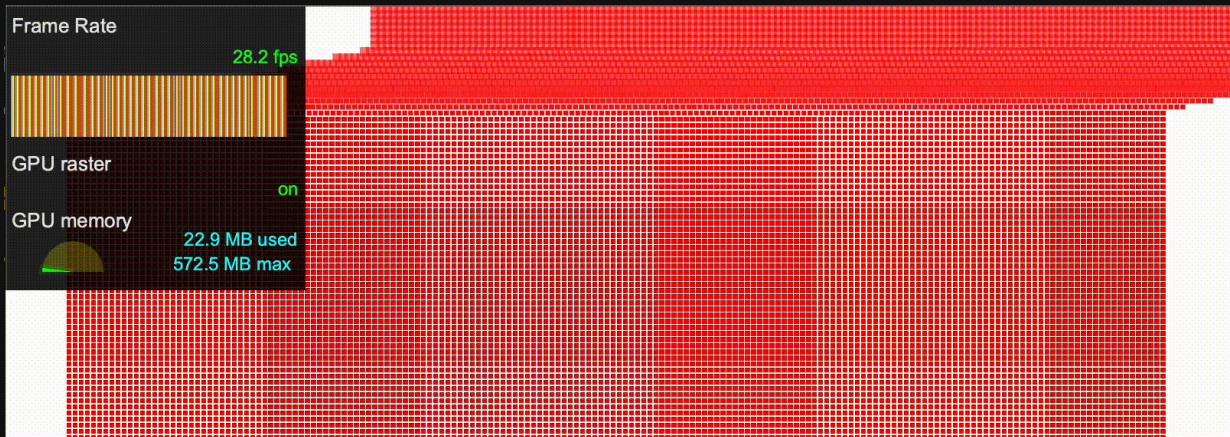
```
attr: { cx: PARAMS.to.cx, r: PARAMS.to.r },  
duration: 5,  
fill: 'blue',  
ease: 'none',  
stagger: 0.01,
```



Canvas

```
x: 200,  
scale: 2,  
opacity: 0.5,  
duration: 5,  
ease: 'none',
```





Benchmark. Данные

Library	PARTICLES	FPS	Avg Memory Usage (b)	CPU usage (ms)	Longest Frame Time (ms)	Jank/Stutter Score	Load time (ms)
gsap	10000	60.1	98549490	7.12	20	1.106	996
animejs	10000	49.17	107389674.7	10.11	51.4	-108.276	1708
mojs	10000	29.92	191051194.4	10.2	46.8	-301.452	1942
popmotion	10000	39.81	101749000.2	10.14	45.9	-202.206	1603
velocity-js	10000	59.97	102369537.4	6.68	28.7	-0.162	1102
react-spring	10000	59.76	116003476.8	7.52	53.8	-2.324	1152
framer-motion	10000	9.1	273900307.4	7.62	151	-508.988	8718
gsap	15000	58.51	133557711.6	9.25	46.7	-24.858	1923
animejs	15000	30.63	175765188.2	10.19	66.9	-294.32	3393
mojs	15000	18.86	420426931.4	10.28	106.9	-412.158	3611
popmotion	15000	23.75	145017660.2	10.23	73.2	-363.05	2991
velocity-js	15000	56.08	146402434	8.44	48.4	0.88	1954
react-spring	15000	56.15	166283913.2	9.45	61.9	-38.378	199
framer-motion	15000	5.29	483378763.6	7.34	202.7	-547.522	
gsap	30000	34.41	253021077	10.18	51.3	-256	
animejs	30000	15.53	339208268.6	10.37	92.9		
mojs	30000	9.42	725334173.4	10.63	149.2		
popmotion	30000	9.15	281656571.2	10.63	117		
velocity-js	30000	38.82	273749105.4	10.02	79.1		
react-spring	30000	28.55	302105270.2	10.19	42.7		
framer-motion	30000	2.63	856986829	7.03	424.3		
gsap	45000	23.02	371789264.8	10.26	63.3		
animejs	45000						
mojs	45000	6.32	1071446478	10.95	181.3		
popmotion	45000	4.9	488422551.8	11.3	223.9		
velocity-js	45000	25.07	409575118.2	10.21	47.1		
react-spring	45000						
framer-motion	45000						



Benchmark. Расчеты

Нормализация для метрик, где больше = лучше

$$\text{Normalized Value} = \frac{\text{Value} - \text{Min Value}}{\text{Max Value} - \text{Min Value}}$$

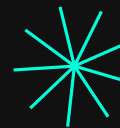
Нормализация для метрик, где меньше = лучше

$$\text{Normalized Value} = 1 - \frac{\text{Value} - \text{Min Value}}{\text{Max Value} - \text{Min Value}}$$

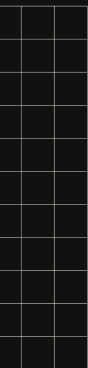
- FPS - 0.35
- Memory - 0.15
- CPU - 0.15
- Longest Frame - 0.2
- Load Time - 0.15

```
= IF(
  AND(ISBLANK(D2), ISBLANK(E2), ISBLANK(F2),
  ISBLANK(G2), ISBLANK(I2)), 0,
  ROUND(
    IF(MAX(FILTER($D$2:$D, $D$2:$D <> "")) =
  MIN(FILTER($D$2:$D, $D$2:$D <> "")), 0,
    (D2 - MIN(FILTER($D$2:$D, $D$2:$D <>
  ""))) / (MAX(FILTER($D$2:$D, $D$2:$D <> "")) -
  MIN(FILTER($D$2:$D, $D$2:$D <> ""))) *
  coeffs!$A$2)
  + MAX(0, 1 - (E2 - MIN(FILTER($E$2:$E,
  $E$2:$E <> ""))) / (MAX(FILTER($E$2:$E, $E$2:$E <>
  "")) - MIN(FILTER($E$2:$E, $E$2:$E <> "")))) *
  coeffs!$B$2)
  + MAX(0, 1 - (F2 - MIN(FILTER($F$2:$F,
  $F$2:$F <> ""))) / (MAX(FILTER($F$2:$F, $F$2:$F <>
  "")) - MIN(FILTER($F$2:$F, $F$2:$F <> "")))) *
  coeffs!$C$2)
  + MAX(0, 1 - (G2 - MIN(FILTER($G$2:$G,
  $G$2:$G <> ""))) / (MAX(FILTER($G$2:$G, $G$2:$G <>
  "")) - MIN(FILTER($G$2:$G, $G$2:$G <> "")))) *
  coeffs!$D$2)
  + MAX(0, 1 - (I2 - MIN(FILTER($I$2:$I,
  $I$2:$I <> ""))) / (MAX(FILTER($I$2:$I, $I$2:$I <>
  "")) - MIN(FILTER($I$2:$I, $I$2:$I <> "")))) *
  coeffs!$E$2,
  2)
)
```

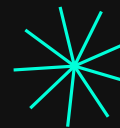
Benchmark. DIVS.



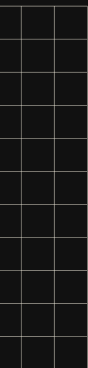
Library	Particles				AVG
	10000	15000	30000	45000	
velocity-js	0.98	0.92	0.85	0.88	0.91
gsap	0.98	0.9	0.82	0.84	0.89
react-spring	0.89	0.84	0.75	0	0.62
popmotion	0.66	0.58	0.52	0.41	0.54
animejs	0.71	0.62	0.58	0	0.48
motion	0.45	0.43	0.45	0.4	0.43
mojs	0.5	0.38	0.39	0.22	0.37
framer-motion	0.11	0.15	0.15	0	0.1



Benchmark. SVGS.



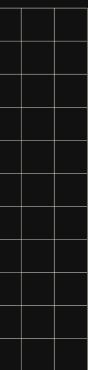
	Particles				
Library	2000	5000	10000	15000	AVG
gsap	0.92	0.61	0.7	0.72	0.74
motion	0.65	0.71	0.67	0.74	0.69
mojs	0.56	0.45	0.75	0.83	0.65
animejs	0.84	0.58	0.52	0.51	0.61
popmotion	0.63	0.51	0.62	0.6	0.59
velocity-js	0.49	0.52	0.69	0.64	0.59
react-spring	0.62	0.31	0.57	0.23	0.43
framer-motion	0.43	0.12	0.33	0.57	0.36



Benchmark. CANVAS.



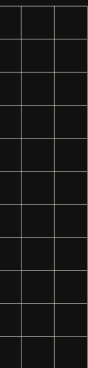
Library	Particles				AVG
	15000	30000	40000	50000	
gsap	1	0.96	1	1	0.99
animejs	0.84	0.85	0	0	0.42
mojs	0.21	0.15	0	0	0.09
popmotion	0.3	0	0	0	0.08



Benchmark. Результаты.



Library	DIVS	SVGS	CANVAS	AVG	AVG WITH CANVAS
gsap	0.89	0.74	0.99	0.82	0.87
velocity-js	0.91	0.59	-	0.75	-
popmotion	0.54	0.59	0.08	0.57	0.4
motion	0.43	0.69	-	0.56	-
animejs	0.48	0.61	0.42	0.55	0.5
react-spring	0.62	0.43	-	0.53	-
mojs	0.37	0.65	0.09	0.51	0.37
framer-motion	0.1	0.36	-	0.23	-

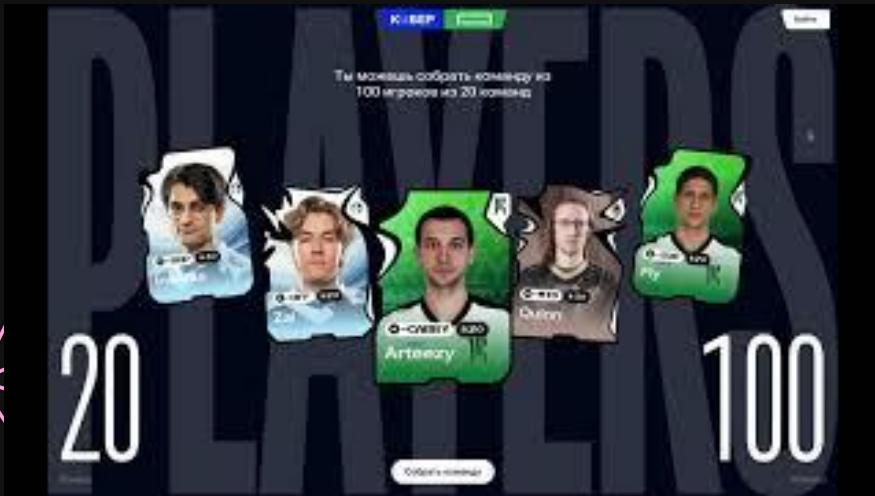


Как GSAP помог мне в работе?

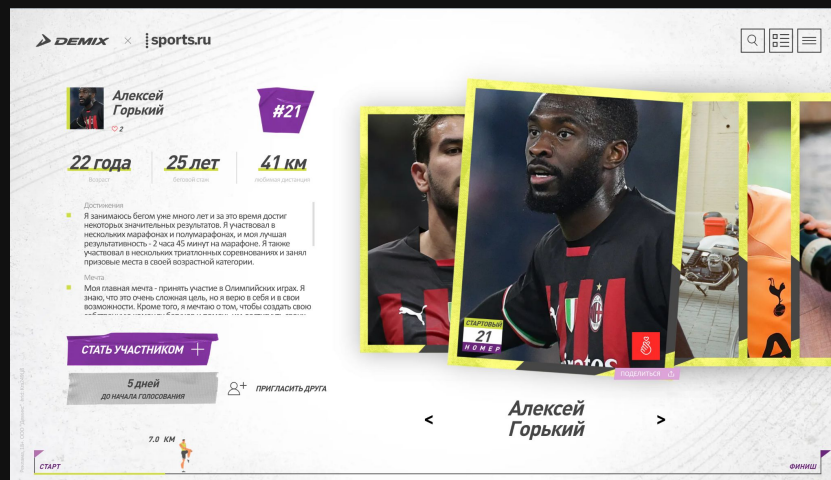


Было

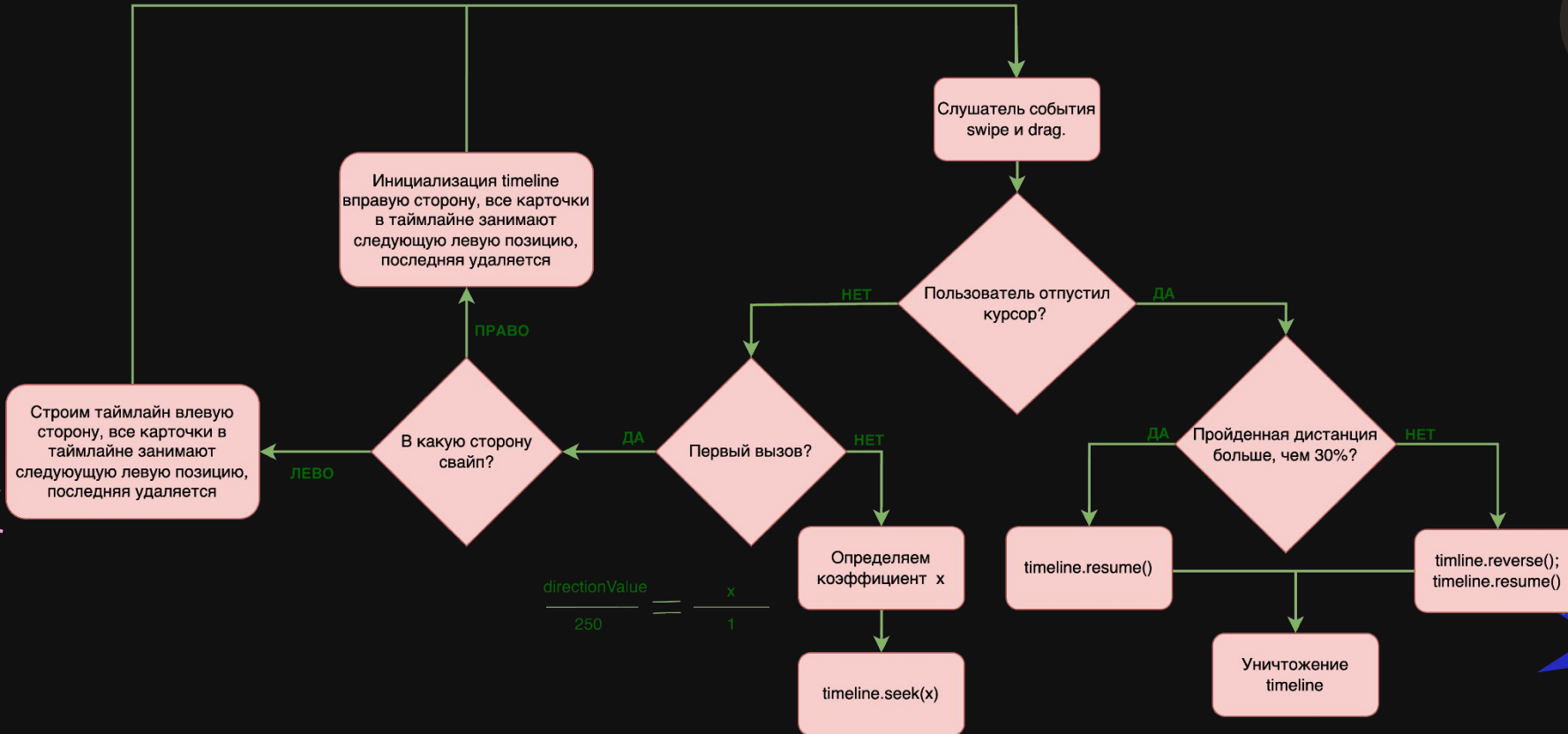
Стало



- Управления стрелками, WASD и жестами
- Ленивая загрузка
- Кнопка “В начало” с анимацией
- Ссылка на спортсмена
- Очень много кастомизации
- Миллион дополнительных желаний от дизайнера



Проекты. Demix x Sports



Проекты. Фоны.



[Redacted name]

10:40 AM



всем привет. я с проблемой. в этот раз плагин Лотти у меня просто отказывается работать и выгружать все должным образом. я за вчерашний вечер и за сегодня утро перепробовала уже раз 40, перебирала варики, как это можно экспортинуть, от чего отказаться, что заменить без потери сути и тд. выдает постоянно разный результат анимации (хреновый). причем проблема в основном в тексте, сделанный в самом ае (то есть вариант с 1 секундой тоже не работает). но если вдруг у меня текст выгрузился нормально, тут же летят фотки.

сейчас я могу еще посидеть и попробовать переформатировать так, чтобы выглядело ок и выгрузилось ок, возможно немного по-другому будет выглядеть. постараюсь минимум поменять.

либо возможно кто-то знает варианты, как это можно НЕ через Лотти достать?

Reply



1



4



December 15, 2023



Проекты. Фоны.

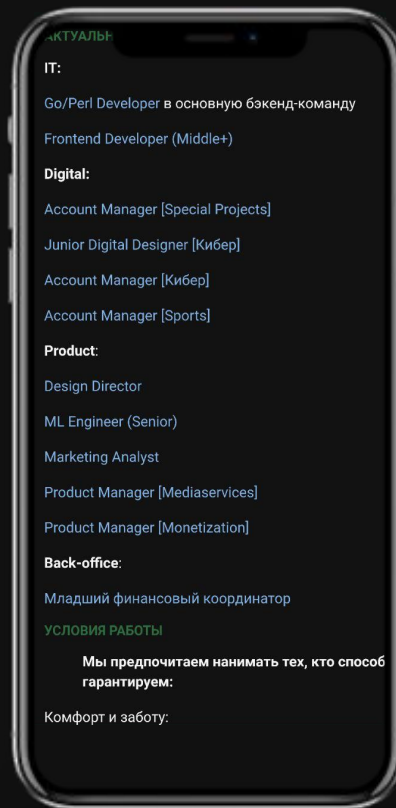
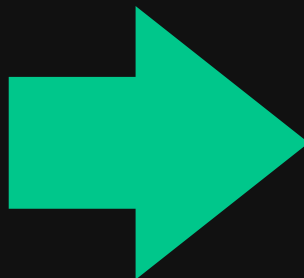


Выводы



sports^{ru}

sports.ru/tribuna/blogs/career/about/



Общие:

- React бенчмарк анимаций – <https://github.com/Sevochka/react-benchmark>
- Nodejs&puppeteer тесты – <https://github.com/Sevochka/animation-libraries-testing>
- Google Sheets расчеты – <https://docs.google.com/spreadsheets/d/1Aqt6sOVhgOFbpNpX6bBQAHQ3pkU2r0fZ3qXi6AniMNo/edit#gid=1806402544>

Спецпроекты с использованием GSAP

- Cyber Fantasy – <https://cyber.sports.ru/fantasy/csgo-2024--pgl-major-copenhagen-2024-03-17>
- Demix Run – <https://specials.sports.ru/demix-run/>
- Sport Quiz – <https://www.sports.ru/special/sport-quiz/>
- Samokat – <https://specials.sports.ru/samokat/>