

## Отображение различных графиков с помощью charts.js на языке JavaScript

*Чингалаев Сергей Алексеевич*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### Аннотация

Целью данной статьи является разработка программы для построения и отображения графиков на языке Java Script. Для достижения этой цели была выбрана библиотека charts.js на языке JavaScript. В статье были рассмотрены способы отрисовки различных графиков

**Ключевые слова:** JavaScript, программа, charts js, разработка, графики.

### Displaying various graphs with charts.js in JavaScript

*Chingalaev Sergey Alekseevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

The purpose of this article is to develop a program for building and displaying graphs in the Java Script language. To achieve this goal, the charts.js library in JavaScript was chosen. The article discussed ways of drawing various charts.

**Keywords:** JavaScript, program, charts js, development, charts.

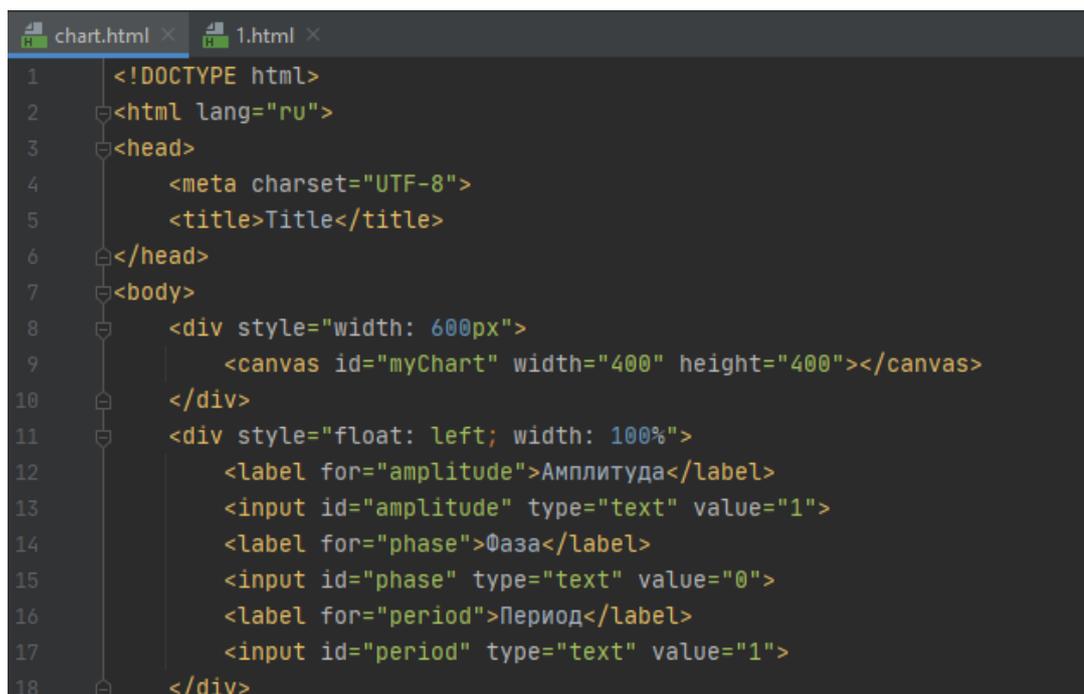
Множество людей используют графики синуса и косинуса для различных математических вычислений. Рассчитывать эти величины может быть очень сложно и довольно затратным по времени. Данная программа, созданная на языке программирования JavaScript, поможет автоматизировать этот процесс, что ускорит время вычисления синуса и косинуса.

Цель исследования: Разработка программы для построения и отображения графиков на языке программирования JavaScript с помощью charts js.

А.В. Кувайцев посвятил статью проблеме выбора среды разработки для веб-приложений на языке JavaScript. Проведенное исследование позволяет уменьшить затраты времени на выбор подходящей среды программирования, так как в статье рассмотрены современные решения от самых простых, вроде текстового редактора с подсветкой синтаксиса, до полноценных IDE [1]. Д.Д. Курмашев, И.К. Асмыкович в своей статье раскрыли и сформулировали сущности обработки цифровых данных в программном обеспечении различных категорий для динамической отрисовки графиков и диаграмм. В результате исследован основной подход и сформирован математический

алгоритм построения графиков и диаграмм на основании данных в независимом формате [2]. Ю.В. Строганов, Д.Ю. Пудов, А.Г. Сиденко в своей статье исследовали численные методы расчёта функций синуса и косинуса (степенные ряды и цепные дроби) для ЭВМ. Данные способы являются наиболее часто реализуемыми на компьютерах. Провели оценку вклада ошибки в вычисления и определена зависимость количества итераций от аргумента и от размера мантиссы поданного на вход числа. [3]. Также не мало важны иностранные источники [4].

Создаем файл html в котором будем делать верстку для отображения графика и изменяем параметр функции синуса и косинуса. Установим ширину графика 50% от ширины окна браузера для удобства. (Рис.1).

The image shows a code editor with two tabs: 'chart.html' and '1.html'. The code is HTML and is as follows:

```
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6 </head>
7 <body>
8   <div style="width: 600px">
9     <canvas id="myChart" width="400" height="400"></canvas>
10  </div>
11  <div style="float: left; width: 100%">
12    <label for="amplitude">Амплитуда</label>
13    <input id="amplitude" type="text" value="1">
14    <label for="phase">Фаза</label>
15    <input id="phase" type="text" value="0">
16    <label for="period">Период</label>
17    <input id="period" type="text" value="1">
18  </div>
```

Рис.1. Верстка

Для отображения графиков сначала необходимо найти на странице canvas. Далее необходимо инициализировать сам график, в который мы передаем canvas для его отображения и параметры отображения. Укажем, что будет две линии пока без значений. Зададим им цвета и обозначим названия каждой, а именно косинус и синус. (Рис.2).

```
19 <script src="https://cdn.jsdelivr.net/npm/chart.js@2.9.3/dist/Chart.min.js"></script>
20 <script type="text/javascript">
21   var ctx = document.getElementById('myChart').getContext('2d');
22   var myChart = new Chart(ctx, {
23     type: 'line',
24     data: {
25       labels: [],
26       datasets: [{
27         fill: false,
28         label: 'Синус',
29         data: [],
30         borderWidth: 1,
31         borderColor: '#333333'
32       },
33       {
34         fill: false,
35         label: 'Косинус',
36         data: [],
37         borderWidth: 1,
38         borderColor: '#aabbba'
39       }
40     ],
41     options: {
42       scales: {
43         yAxes: [{
44           ticks: {
45             beginAtZero: true
46           }
47         }]
48       }
49     }
50   });
```

Рис.2. Инициализация графика

Далее необходимо добавить данные для построения графика, а именно значения по координатам X и Y. Для этого будем использовать цикл, в котором X будем изменяться с шагом 1/10, а Y для каждой линии будет вычисляться математической функцией cos и sin. (Рис.3)

```
51   var phase = 0
52   var period = 1.5
53   var ampl = 2
54   for (var i=0; i < 10; i+=0.1){
55     var x = i/period + phase
56     myChart.chart.data.labels.push(x)
57     myChart.chart.data.datasets[0].data.push(Math.sin(x)*ampl)
58     myChart.chart.data.datasets[1].data.push(Math.cos(x)*ampl)
59   }
60   myChart.update()
61 </script>
62 </body>
63 </html>
```

Рис.3. Добавление данных графика

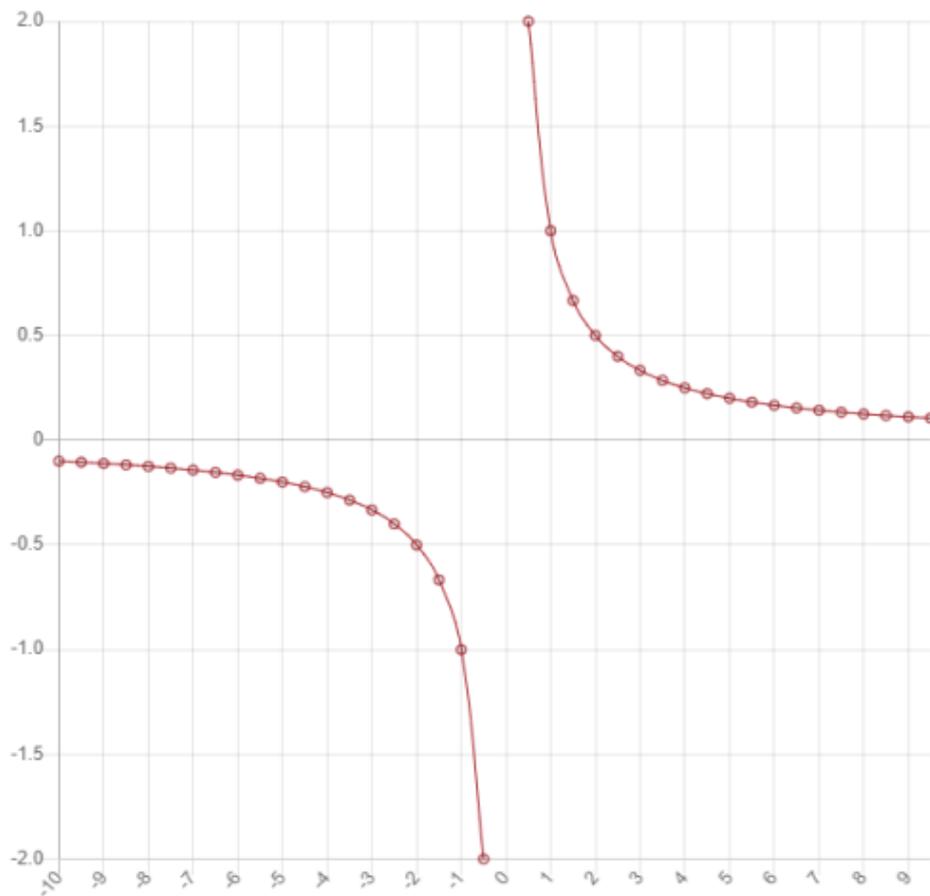
Открыв страницу, видим данный график. (Рис.4)



```
for (var i=-10; i < 10; i+=0.5){  
  myChart.chart.data.labels.push(i)  
  myChart.chart.data.datasets[0].data.push(1 / i)  
}
```

Рис.6. Параметры разрывной функции

Далее необходимо добавить данные для построения графика, а именно значения по координатам X и Y. После ввода этих данных мы получаем разрывную функцию (Рис.7).

Рис. 7. Пример разрывной функции  $1/x$ 

Для пропорциональности оси x необходимо установить следующие настройки: (Рис.8)

```
options: {
  scales: {
    xAxes: [{
      type: 'linear', /* <--- this */
      scaleLabel: {
        display: true,
      },
      ticks: {
        autoSkip: true,
        maxTicksLimit: 10
      }
    }],
    yAxes: [{
      scaleLabel: {
        display: true
      },
      ticks: {
        suggestedMin: 0,
        suggestedMax: 3700
      }
    }],
  },
}
```

Рис. 8. Пропорциональность оси X

9). Далее напишем функцию движения тела, брошенного под углом (Рис.

```
var v0 = 550
var a_degrees = 35
var a = a_degrees * (Math.PI / 180)
for (var t = 0; t < 100; t += 1) {
  var x = v0 * t * Math.cos(a)
  var y = v0 * t * Math.sin(a) - 9.8 * (t * t) / 2
  myChart.chart.data.labels.push(x)
  myChart.chart.data.datasets[0].data.push({x: x, y: y})
}
```

Рис. 9. Функция движения под углом

После введения данных значений мы получаем пример графика функции движения, брошенного под углом (Рис. 10).

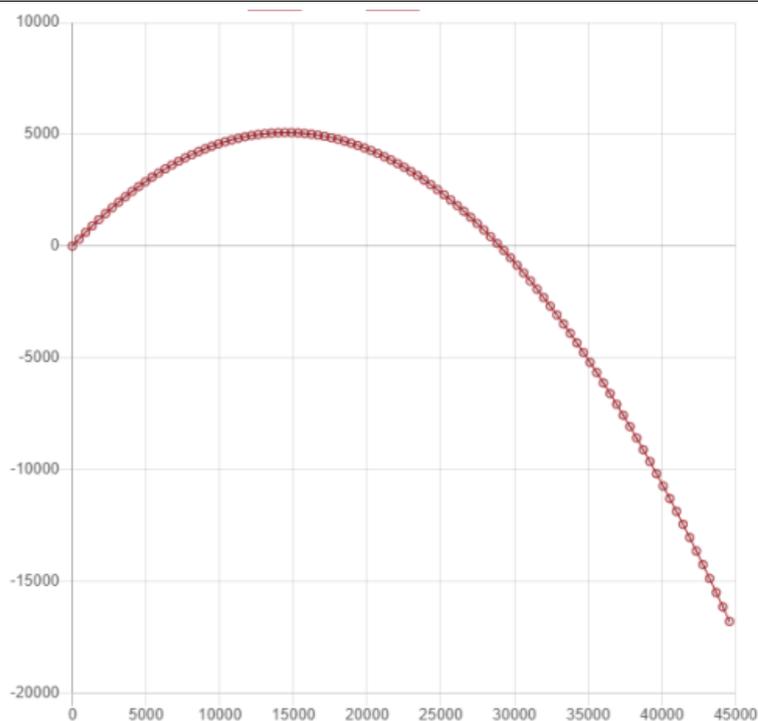


Рис. 10. График функции движения тела под углом

Таким образом, была написана программа, для построения и отображения графиков на языке программирования JavaScript с помощью charts.js.

### Библиографический список

1. Кувайцев А.В. Анализ сред разработки веб-приложений на языке javascript // Современные научные исследования и инновации. 2016. № 9 (65). С. 121-124.
2. Курмашев Д.Д., Асмыкович И.К. Анализ и обработка данных для построения графиков // Роль математики в становлении специалиста-2020. Материалы Международной научно-практической конференции. Уфимский государственный нефтяной технический университет. 2020. С. 49-53.
3. Строганов Ю.В., Пудов Д.Ю., Сиденко А.Г. Исследование численных методов для синуса и косинуса // Новые информационные технологии в автоматизированных системах. 2018. № 21. С. 363-366.
4. Duval B.K., Main L. Javascript // Library Software Review. 1997. Т. 16. № 3. С. 164-173.