

**Московский государственный технический университет
им. Н.Э. Баумана**

**Разработка интернет-приложений
Лабораторная работа № 8
“Javascript”
(с доп. заданием)**

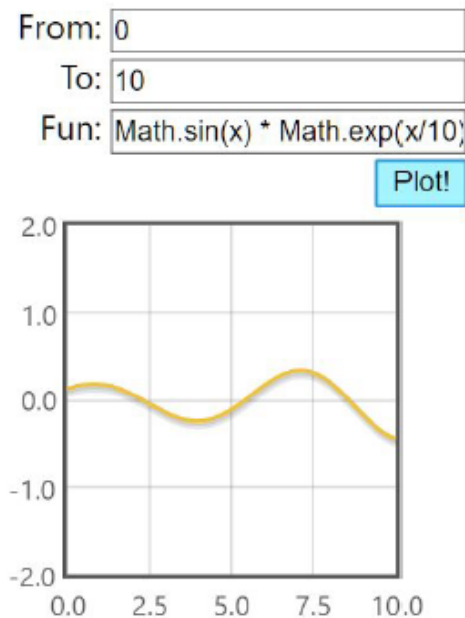
Выполнил:
студент группы ИУ5-53
Пирмамедов М. Э.
Подпись:
Дата:

Москва 2017г.

Задание и порядок выполнения

Разработать приложение для построения графиков тригонометрических функций на языке Javascript с HTML интерфейсом.

Пример интерфейса:



Ход работы:

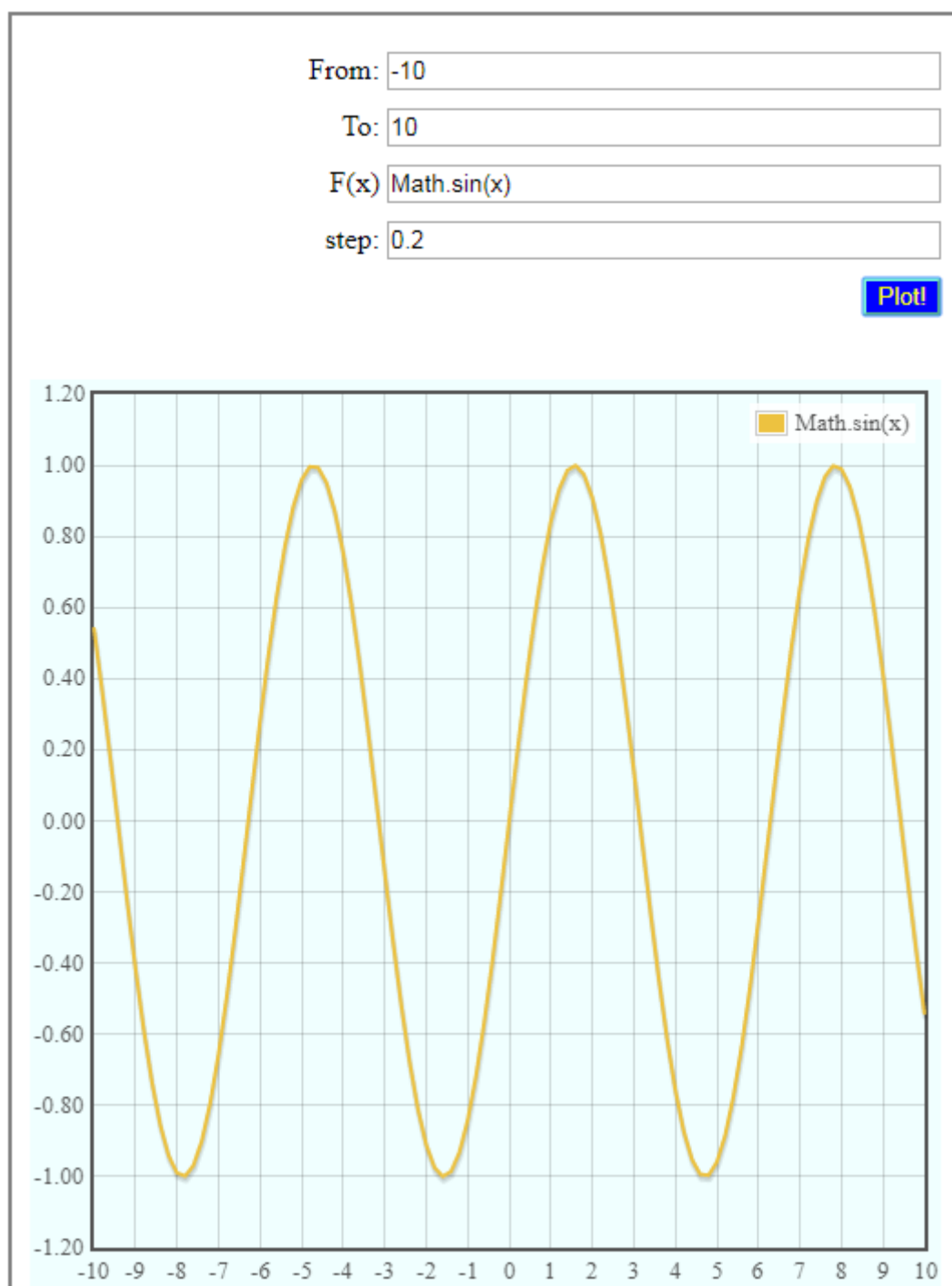
1. Ознакомиться с теоретической частью
2. Создайте новый проект PyCharm
тип проекта: Pure Python
(мы не будем использовать Python в этой работе, просто это позволяет создать абсолютно пустой проект без зависимостей)
3. Добавьте в проект 2 файла:
 - a. index.html
 - b. index.js
4. Сверстайте страницу со следующими элементами:
 - a. два поля ввода для области определения аргумента (`<input>`)
 - b. поле для ввода функции (`<input>`)
 - c. кнопка "Построить график" (`<button>`)
 - d. поле вывода графика (`<div>`)
5. При помощи css укажите размеры блока графика, отличные от нуля
6. Присвойте каждому полю уникальный class (например, from, to, fun, output и т.д.)

7. Убедитесь, что ваша страница отображается в браузере нормально
8. Подключите jQuery, flot и ваш скрипт в index.html, используя теги `<script>`
`<script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>`
`<script src="https://cdnjs.cloudflare.com/ajax/libs/flot/0.8.3/jquery.flot.js"></script>`
9. Переходим к разработке скрипта
10. Дождитесь загрузки страницы

```
$(function() {  
    // ...  
})
```
11. Найдите все элементы управления на вашей странице
`var $from = $('from');`
12. Подпишитесь на событие нажатия кнопки
`$button.click(onClick);`
13. Отмените действие по-умолчанию (отправку формы)
`e.preventDefault();`
14. Получите значения из полей ввода
`$from.val();`
15. Не забудьте преобразовать числовые значения из строк в числа
`parseFloat, parseInt`
16. Создайте массив пар значений
`const points = [[x1, y1], ..., [xn, yn]];`
17. Для того, чтобы получить значение функции, заданной в виде строки, используйте функцию `eval()`

```
const x = 0.1;  
const fun = 'Math.sin(x)';  
const y = eval(fun);  
// ...
```
18. Постройте график по точкам
`$.plot($output, [points], {});`
19. Проверьте правильность работы приложения, в случае проблем, воспользуйтесь отладчиком Chrome DevTools
20. Проверьте построение графиков функций:
 - a. `Math.sin(x)`
 - b. `Math.random()`
 - c. `Math.exp(x)`
21. Выведите название построенной функции в легенду:
<http://www.flotcharts.org/flot/examples/basic-options/index.html>
22. Дополнительное задание:
сделайте анимацию графика функции как на осциллографе
для этого по таймеру `setInterval()` / `clearInterval()` перестраивайте график функции, прибавляя к **x** изменяющийся коэффициент **dx**

Результат выполнения:



Исходники

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <div class="plot" style="height: 680px ; width: 500px; border: solid gray 2px; padding: 10px">
    <div class="input_group" style="text-align: right; display: inline-block; width: 500px; height: 180px;">
      <label style="display: inline-block; margin-top: 10px;">
        From: <input id="from" title="From" style="width: 300px;">
      </label>
      <label style="display: inline-block; margin-top: 10px;">
        To: <input id="to" title="To" style="width: 300px;">
      </label>
      <label style="display: inline-block; margin-top: 10px;">
        F(x) <input id="fun" title="Fun" style="width: 300px;">
      </label>
      <label style="display: inline-block; margin-top: 10px;">
        step: <input id="step" title="step" style="width: 300px;">
      </label>
      <div class="but" style="display: block; margin-top: 10px;">
        <button id="btn" style="border-color: aquamarine; background-color: blue; color: yellow">Plot!</button>
      </div>
    </div>
    <div id="plot_canvas"
      style="display: block; height: 500px; width: 500px; background-color: azure; margin-top: 10px">
    </div>
  </div>
  <script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/flot/0.8.3/jquery.flot.js"></script>
  <script src="index.js"></script>
</body>
</html>
```

Index.js

```
$(document).ready(onload);
function onload() {
    var $button = $('#btn');
    function draw_graph(from, to, step, fun_str, canvas) {
        var array_of_dots = [];
        var x = from;
        var refreshInterval = setInterval(function () {
            var dot = [x, eval(fun_str)];
            array_of_dots.push(dot);
            $.plot(canvas, [
                {
                    label: fun_str,
                    data: array_of_dots
                }
            ], {
                series: {
                    lines: {show: true},
                    points: {show: false}
                },
                xaxis: {
                    min: from,
                    max: to,
                    ticks: to - from
                },
                yaxis: {
                    ticks: 10,
                    tickDecimals: 2
                }
            }
        );
        x = x + step;
        if (x > to) clearInterval(refreshInterval);
    }, 100);
    console.log("end of drawing");
}

$button.click(function (e) {
    e.preventDefault();
    var from_val = parseFloat($('#from').val());
    var to_val = parseFloat($('#to').val());
    var fun_str = $('#fun').val();
    var $canvas = $("#plot_canvas");
    var step_val = parseFloat($('#step').val());
    draw_graph(from_val, to_val, step_val, fun_str, $canvas);
});
}
```