

Защищено:

Гапанюк Ю.Е.

Демонстрация ЛР:

Гапанюк Ю.Е.

"__" _____ 2016 г.

"__" _____ 2016 г.

Отчет по лабораторной работе №8
по курсу РИП

Вариант № <26>

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-52

(подпись)

Чекулина М.Ю.

"__" _____ 2016 г.

1. Задание

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

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/plot/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 .

2. Код

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="index.css">
  <script type="text/javascript" src="jquery-3.1.1.min.js"></script>
  <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/flot/0.8.3/jquery.flot.js" ></script>
  <script src="index.js"></script>
  <title>Построение графика функции</title>
</head>
<body>
<form>
  <label>From:</label>
  <input class="from">
  <br>
  <p> </p>
  <label>To:</label>
  <input class="to">
  <br>
  <p> </p>
  <label>Fun:</label>
  <input class="fun">
  <br>
  <p> </p>
  <button action="/" class="button">Plot!</button>
  <br>
  <div class="graph"
    style="margin: 15px;
    width: 600px;
    height: 600px;
    border: solid black;">
  </div>
</form>
</body>
</html>
```

index.js

```
/**
 * Created by Марина on 12.12.2016.
 */

$(function()
{
  // 11. Найдите все элементы управления на вашей странице var $from = $('from');
  var $from=$(".from");
  var $to=$(".to");
  var $fun=$(".fun");
  var $graph=$(".graph");
  var $button=$(".button");
  var timerId = 0;

  // 12. Подпишитесь на событие нажатия кнопки $button.click(onClick);
  $button.click(function(e)
  {
    // 13. Отмените действие по-умолчанию (отправку формы) e.preventDefault()
```

```

        e.preventDefault();
// 14. Получите значения из полей ввода $from.val()
// 15. Не забудьте преобразовать числовые значения из строк в числа parseFloat,
parseFloat
        var from = parseFloat($from.val());
        var to = parseFloat($to.val());
        var fun = ($fun.val());
// 16. Создайте массив пар значений const points = [[x1, y1], ..., [xn, yn]];
        const points = [];
        var dx = 0.01;
        var interval = 10;
        var start = from;
        var end;
// сделайте анимацию графика функции как на осциллографе для
// этого по таймеру setInterval() / clearInterval() перестраивайте
// график функции, прибавляя к x изменяющийся коэффициент dx
        clearInterval( timerId );
        timerId = setInterval( function()
        {
            end = start + interval;
            if( end > to )
                clearInterval( timerId );
            if( start != from )
                start += 9.9;
            for( ; start < end; start += dx ) {
                if( start > to )
                    break;
// 17. Для получения значение функции, заданной в виде строки,
// используйте функцию eval() const x = 0.1
// ;const fun = 'Math.sin(x)' ;const y = eval (fun);
                const x = start;
                const y = eval(fun);
                points.push([x, y]);
            }
// 18. Постройте график по точкам $. plot ( $ output, [ points ], {});
            $.plot($graph, [{ label: fun, data: points }], [points], {});

            var i = 0;
            start -= 9.9;
            while( points[i][0] < start ) {
// arr.shift()-удаляет элемент с индексом 0 и сдвигает остальные элементы на один вниз.
// Возвращает удаленный элемент
                points.shift();
                i += 1;
            }
        }, 10);
    });
});

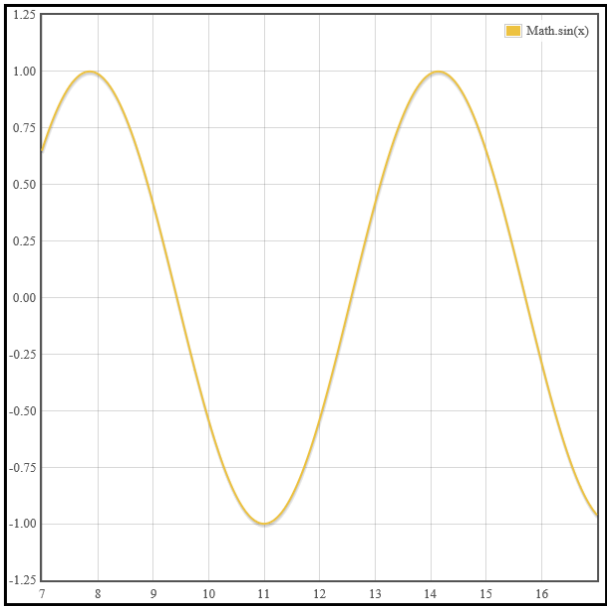
```

3. Результат

From:

To:

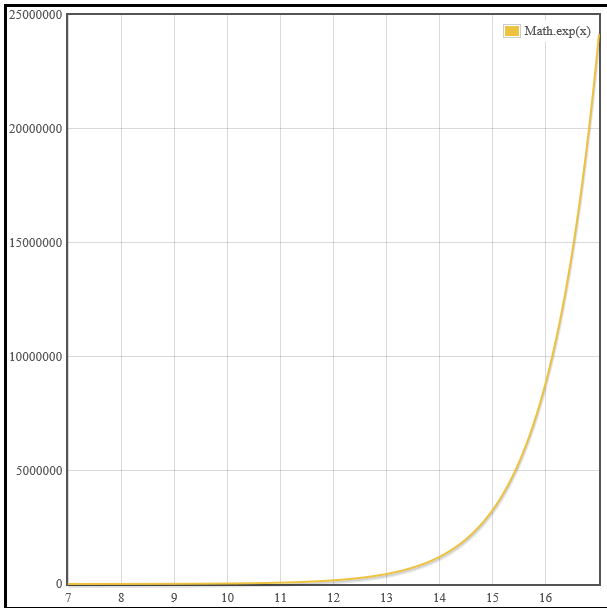
Fun:



From:

To:

Fun:



From:

To:

Fun:

