

Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и управления»



Отчет по лабораторной работе № 8

«Javascript»

по курсу

“Разработка Интернет-приложений”

ИСПОЛНИТЕЛЬ:

Березин И.С.

Группа ИУ5-53

"__" _____ 2016 г.

ПРЕПОДАВАТЕЛЬ:

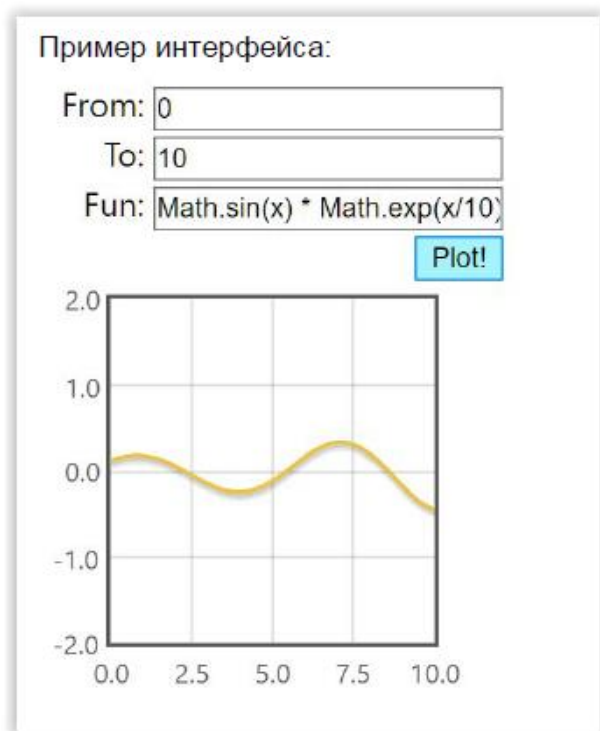
Гапанюк Ю.Е.

"__" _____ 2016 г.

Москва 2016

Задание и порядок выполнения:

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/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*

Файл 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>Title</title>

</head>

<body>

<label>From:</label>
```

```

<input class="from">
<br>
<label>To:</label>
<input class="to">
<br>
<label>Fun:</label>
<input class="fun">
<br>
<style>
.b1 {
    background: #03A9F4; /* Голубой цвет фона */
    color: #070000; /* Черные буквы */
    font-size: 10pt; /* Размер шрифта в пунктах */
}
</style>
<button action="/" class="button">Plot!</button>
<br>
<div class="graph"></div>
</body>
</html>

```

Файл index.css

```

label {
    width: 40px;
    display: inline-block;
}
.graph {
    margin: 20px;
    width: 400px;
    height: 400px;
    border: 2 npx solid black;
}

```

Файл index.js

```
$(function() {  
    var timerID = 0;  
    var $from=$("#.from");  
    var $to=$("#.to");  
    var $fun=$("#.fun");  
    var $graph=$("#.graph");  
    var $button=$("#.button");  
  
    $button.click(function(event) {  
        clearInterval(timerID);  
        event.preventDefault();  
        var from = parseInt($from.val());  
        var to = parseInt($to.val());  
        var fun = ($fun.val());  
        var points = [];  
        var dx = 0.05;  
        var interval = 10;  
        var start = from;  
        var end;  
  
        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;  
                x = start;  
                y = eval(fun);  
                points.push([x, y]);  
            }  
        }  
    });  
});
```

```
$.plot($graph, [{ label: fun, data: points }], [points], {});

var i = 0;

start -= 9.9;

while( points[i][0] < start ) {

    points.shift();

    i += 1;

}

}, 100);

});

});
```

From: 0
 To: 100
 Fun: $\text{Math.sin}(x) * \text{Math.exp}(x/10)$
 Plot!

