

# **Лабораторная работа №8.**

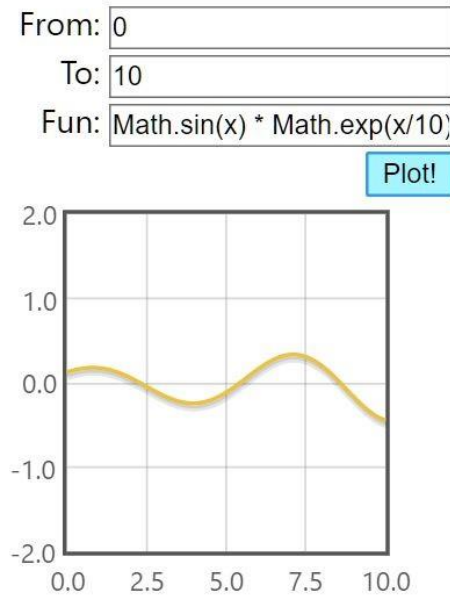
**Тема: "Javascript"**

**Студент:** Щипицин Р.А.,  
группа ИУ5-51.

## Описание задания лабораторной работы

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>

## Реализация.

### Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
  <meta charset="UTF-8">
  <title>График</title>
</head>
<body>
  <form>
    <p> Введите ограничения для оси координат X </p>
    <p> От: <input name="from" class="from"> </p>
    <p> До: <input name="to" class="to"> </p>
    <p> Введите данную вам функцию </p>
    <p> Функция: <input name="fun" class="fun"> </p>
    <p><button class="plot" type="button">Построить
```

```

график</button></p>
</form>
<div class="graph"></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"></scr
ipt>
<script src='index.js'></script>
</body>
</html>

```

## Index.js

```

var graf;
$('.plot').click(function (e) {
    clearInterval(graf);
    console.log(graf);
    var x = parseFloat($('.from').val());
    const x1 = x;
    var i = x;
    var step = 1;
    const x2 = parseFloat($('.to').val());
    const fun = ($.fun').val();
    var pointst = [x, eval(fun)];

    console.log(pointst);
    console.log(fun);

    graf = setInterval(function () {
        if (x < x2) {
            $.plot($('.graph'), [{label: fun, data: pointst}], {});
            x = x + (x2 - x1) / 100;
            console.log(pointst);
            if (pointst.length > 100) {
                pointst= pointst.splice(1)
            }
            pointst.push([x, eval(fun)])
            i += parseFloat(step)
        }
        else {
            clearInterval(graf);
        }
    }, 100);
});

```

## style.css

```

.graph{
    border:1px solid black;
    width: 500px;
    height: 500px;
}

```

## Результат работы.

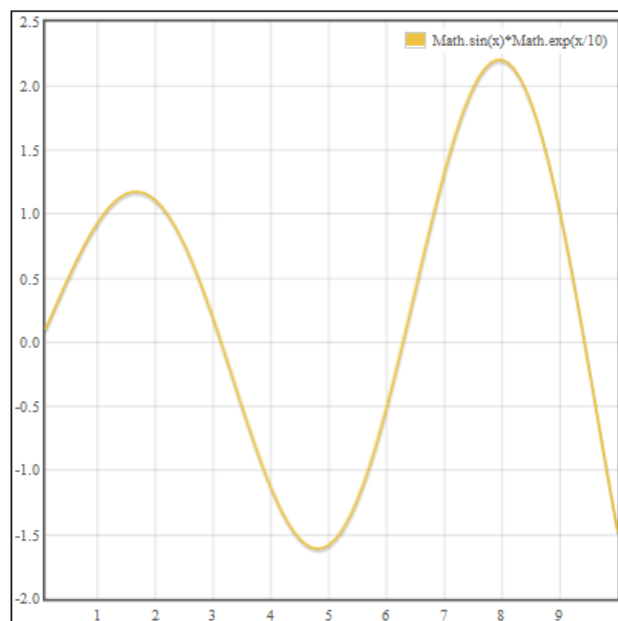
Введите ограничения для оси координат X

От:

До:

Введите данную вам функцию

Функция:



## Проверка построения графиков функций:

а.  $\text{Math.sin}(x)$

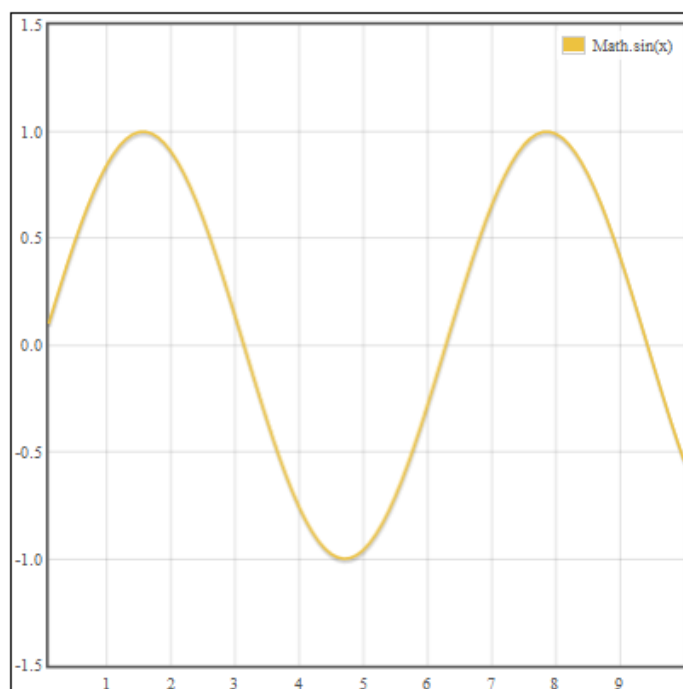
Введите ограничения для оси координат X

От:

До:

Введите данную вам функцию

Функция:



b.  $\text{Math.random}()$

Введите ограничения для оси координат X

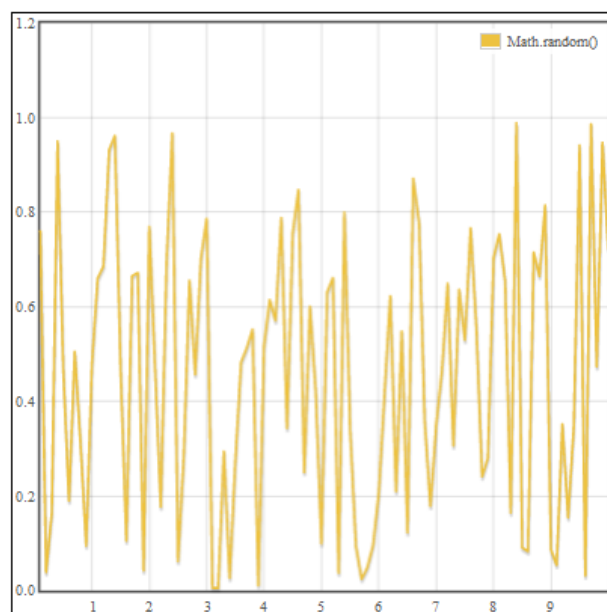
От:

До:

Введите данную вам функцию

Функция:

[Построить график](#)



c.  $\text{Math.exp}(x)$

Введите ограничения для оси координат X

От:

До:

Введите данную вам функцию

Функция:

[Построить график](#)

