

Московский государственный технический университет имени Н.Э. Баумана

**Кафедра ИУ5
«Системы обработки информации и управления»**

Отчет по лабораторной работе №8

«Javascript»

по дисциплине «Разработка Интернет-приложений»

**Выполнил:
студент группы ИУ5-53
Слимов Никита**

Москва, 2016

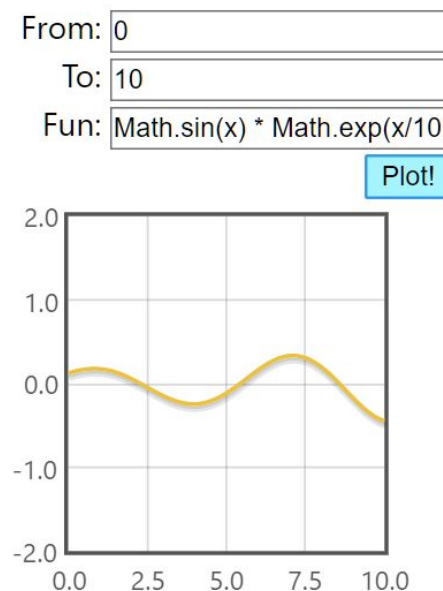
Задание

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:
5. два поля ввода для области определения аргумента (<input>)
6. поле для ввода функции (<input>)
7. кнопка "Построить график" (<button>)
8. поле вывода графика (<div>)
9. При помощи css укажите размеры блока графика, отличные от нуля
10. Присвойте каждому полю уникальный class (например, from, to, fun, output и т.д.)



11. Убедитесь, что ваша страница отображается в браузере нормально
12. 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>`
13. Переходим к разработке скрипта
14. Дождитесь загрузки страницы `$(function() { // ... })`
15. Найдите все элементы управления на вашей странице `var $from = $('from');`
16. Подпишитесь на событие нажатия кнопки `$button.click(onClick);`
17. Отмените действие по-умолчанию (отправку формы) `e.preventDefault()`
18. Получите значения из полей ввода `$from.val()`
19. Не забудьте преобразовать числовые значения из строк в числа `parseFloat`, `parseInt`
20. Создайте массив пар значений `const points = [[x1, y1], ..., [xn, yn]];`

21. 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);`
22. 18. Постройте график по точкам `$. plot ($ output, [points] , {})`;
23. Проверьте правильность работы приложения, в случае проблем, воспользуйтесь отладчиком Chrome DevTools
24. Проверьте построение графиков функций: а. Math.sin(x) б. Math.random() с. Math.exp(x)
25. Выведите название построенной функции в легенду:
<http://www.flotcharts.org/plot/examples/basic-options/index.html>
26. Дополнительное задание: сделайте анимацию графика функции как на осциллографе для этого по таймеру setInterval() / clearInterval() перестраивайте график функции, прибавляя к x изменяющийся коэффициент dx

Исходный код

Файл index.css

```
html {
  font-family: monospace;
  -webkit-font-smoothing: antialiased;
  color: #333;
  font-size: 137.5%;
}

h2 {
  -webkit-margin-before: 0;
  margin: .5em 0;
  font-size: 2.828em;
  line-height: 1.5em;
}

.body-container {
  max-width: 600px;
  margin: 0 auto;
  text-align: center;
}

.header {
  margin-bottom: 1em;
}

.range-header {
  margin-bottom: 1.4em;
}

.function-container {
  margin-top: 1.4em;
}

.function {
  background-color: rgba(0, 0, 0, 0);
  border: none;
  border-bottom: 1px solid #ccc;
  outline: none;
  height: 2em;
  width: 50%;
  font-size: 1.7em;
}
```

```

    font-family: monospace;
    margin: .5em 0 .2em 0;
    padding: 0;
    box-shadow: none;
    box-sizing: content-box;
    transition: all 0.3s;
    text-align: center;
}

.function.error {
    color: red;
}

a, a:visited {
    color: #4068B2;
}

a:hover, a:focus {
    color: #7BAAFF;
}

.buttons {
    margin: .8em 0;
}

.button {
    font-size: 1.5em;
    margin: 0 1em;
    padding: .5em 1em;
}

.graph {
    height: 250px;
}

```

Файл index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Лабораторная работа 8</title>
  <link rel="stylesheet" type="text/css" href="/index.css"/>
  <link rel="stylesheet" type="text/css" href="/assets/jquery.range.css"/>
  <script type="text/javascript" src="/assets/jquery-
3.1.1.min.js"></script>
  <script type="text/javascript" src="/assets/jquery.range-
min.js"></script>
  <script type="text/javascript" src="/assets/jquery.flot.min.js"></script>
  <script type="text/javascript" src="/index.js"></script>
</head>
<body>
<div class="body-container">
  <div class="header">
    <h2>Build a graph</h2>
  </div>

  <div class="range-container">
    <div class="range-header">Range</div>
    <div><input class="range" type="hidden" value=""/></div>
  </div>

```

```

<div class="function-container">
  Function: <input class="function" type="text" value=""
placeholder="Function"/>
</div>

<div class="buttons">
  <a href="#" class="button button-go">Go</a>
  <a href="#" class="button button-start">Start</a>
  <a href="#" class="button button-stop">Stop</a>
</div>

<div class="graph">
</div>
</div>
</body>
</html>

```

Файл index.js

```

var cache = (function () {
  var isSupported = function (storageName) {
    try {
      return (storageName in window && window[storageName]);
    }
    catch (e) {
      return false;
    }
  };

  var ls = 'localStorage';
  if (isSupported(ls)) {
    var storage = window[ls];
    return {
      get: function (key) {
        var res = storage.getItem(key);

        try {
          return JSON.parse(res);
        }
        catch (e) {
          return res || undefined;
        }
      },
      set: function (key, val) {
        if (val === undefined) {
          storage.remove(key);
          return;
        }

        if (typeof val !== 'string') {
          val = JSON.stringify(val);
        }

        return storage.setItem(key, val);
      }
    };
  }

  var local = {};
  return {
    get: function (key) {
      return local[key];
    }
  };
}

```

```

    },
    set: function (key, val) {
        local[key] = val;
    }
}
})();

$(function () {
    var stepsPerGraph = 1000;
    var stepsOscilloscope = 100;
    var timeOscilloscope = 300;

    var $function = $('.function');
    var cacheFunctionKey = 'function';

    var $range = $('.range');
    var cacheRangeKey = 'range';

    var currentRange = cache.get(cacheRangeKey);

    if (!currentRange) {
        currentRange = '-100,100';
    }

    $range.jRange({
        from: -100,
        to: 100,
        step: 1,
        scale: [-100, -75, -50, -25, 0, 25, 50, 75, 100],
        format: '%s',
        width: 'auto',
        showLabels: true,
        isRange: true,
        onstatechange: function (val) {
            cache.set(cacheRangeKey, val);
        }
    }).jRange('setValue', currentRange);

    var getCurrentRange = function () {
        var res = $range.val().split(',');
        return {from: parseFloat(res[0]), to: parseFloat(res[1])};
    };

    var currentFunction = cache.get(cacheFunctionKey);

    if (!currentFunction) {
        currentFunction = 'Math.sin(x)';
    }

    $function.val(currentFunction)
        .keyup(function () {
            cache.set(cacheFunctionKey, $(this).val());
            $(this).trigger('check');
        }).on('check', function () {
            var isError = false;
            var func = $(this).val();

            try {
                var range = getCurrentRange();
                var x = range.from;

                isError = isNaN(eval(func));
            }
        });

```

```

    }
    catch (e) {
        isError = true;
    }

    $(this).toggleClass('error', isError);
}).trigger('check');

var $buttonGo = $('.button-go');
var $buttonStart = $('.button-start');
var $buttonStop = $('.button-stop');

var $graph = $('.graph');

var countGraphData = function (func, from, to) {
    var data = [];
    var step = Math.abs(to - from) / stepsPerGraph;

    try {
        for (var x = from; x < to; x += step) {
            data.push([x, eval(func)]);
        }
    }
    catch (e) {
        return [];
    }
    return data;
};

var renderGraph = function (fun, from, to) {
    $.plot($graph, [
        {
            data: countGraphData(fun, from, to),
            label: fun,
            color: '#4068B2'
        }
    ]);
};

$buttonGo.click(function (e) {
    e.preventDefault();

    var range = getCurrentRange();
    var functionVal = $function.val();

    renderGraph(functionVal, range.from, range.to);
});

var latestInterval = null;

$buttonStart.click(function (e) {
    e.preventDefault();

    if (latestInterval) {
        clearInterval(latestInterval);
    }

    var range = getCurrentRange();
    var from = range.from;
    var to = range.to;

```

```

var functionVal = $function.val();
var dx = Math.max(0.1, Math.abs(to - from) / stepsOscilloscope);

var next = function () {
    renderGraph(functionVal, from, to);
    from += dx;
    to += dx;
};

latestInterval = setInterval(next, timeOscilloscope);
});

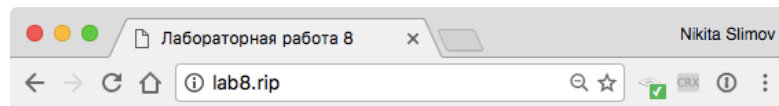
$buttonStop.click(function (e) {
    e.preventDefault();

    if (!latestInterval) {
        return;
    }

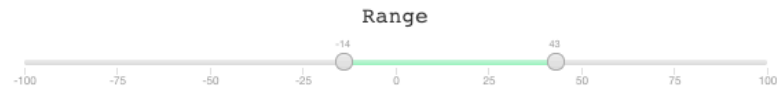
    clearInterval(latestInterval);
    latestInterval = null;
});
});

```

Результат



Build a graph

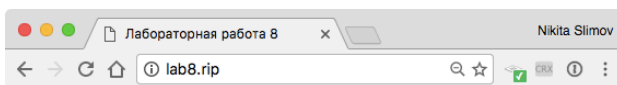
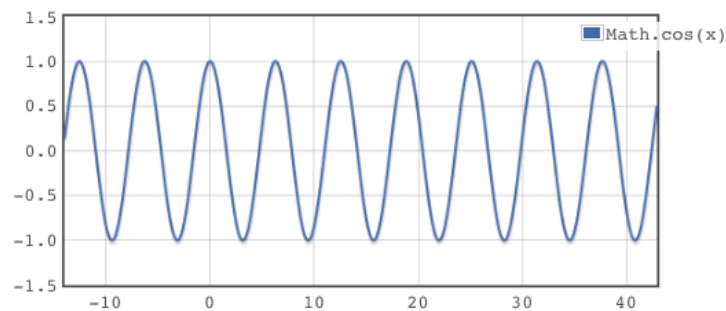


Function: `Math.cos(x)`

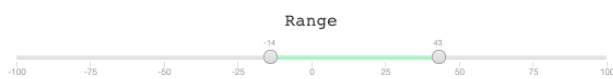
[Go](#)

[Start](#)

[Stop](#)



Build a graph

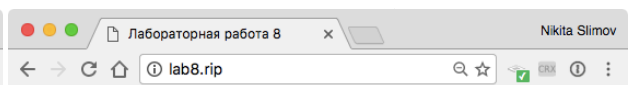
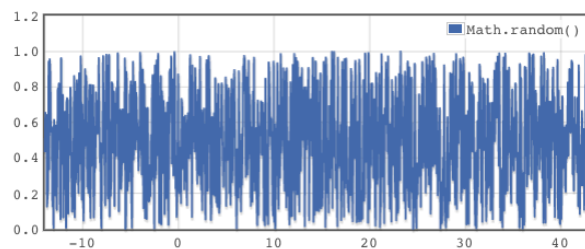


Function: `Math.random()`

[Go](#)

[Start](#)

[Stop](#)



Build a graph



Function: `Math.random()`

[Go](#)

[Start](#)

[Stop](#)

