

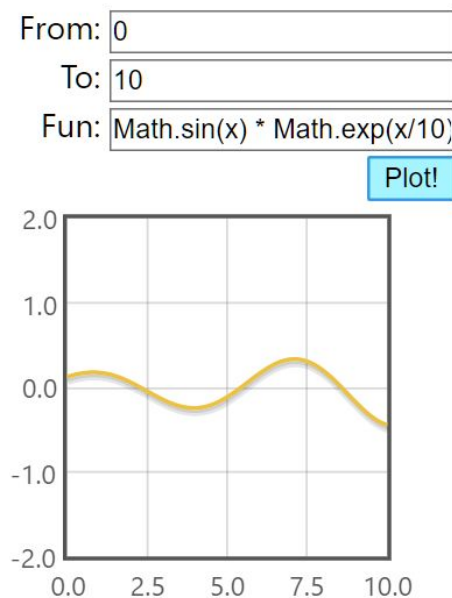
Лабораторная работа №8

Javascript

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**

Теория и примеры

Javascript

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

Javascript обычно выполняется на стороне клиента, в браузере. С помощью JS можно взаимодействовать с DOM (дерево HTML элементов) текущей открытой страницы, управлять навигацией, отправлять данные на сервер (AJAX) и получать обратно.

Однако браузер является лишь окружением для Javascript, предоставляя ему своё API. Javascript также может использоваться и как серверный язык, аналогично, например, Python (Django, Tornado) или C# (ASP .Net). Также возможно собрать standalone десктопное приложение аналогично C++ (Qt) или C# (WPF). Однако это уже скорее экзотика.

Отличное руководство по JS можно найти здесь: <https://learn.javascript.ru>

Примеры синтаксиса

Подсчёт среднего

```
function avg(arr) {  
    var result = 0;  
  
    if (arr.length == 0)  
        return 0;  
  
    for (var i = 0; i < arr.length; ++i) {  
        result += arr[i];  
    }  
  
    return result / arr.length;  
}  
  
console.log(avg([10, 20]));
```

jQuery

jQuery — библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX.

Возможности

- Движок кросс-браузерных CSS-селекторов Sizzle, выделившийся в отдельный проект;
- Переход по дереву DOM;
- События;
- Визуальные эффекты;
- AJAX-дополнения;
- JavaScript-плагины.

jQuery поставляется в виде одного огромного js-файла, который можно скачать здесь: <https://jquery.com/download/>

Compressed версия функционально не отличается от обычной. Она сжата для уменьшения объема файла.

Примеры использования

Подключение jQuery

Вариант со ссылкой:

```
<script src="https://code.jquery.com/jquery-3.1.0.min.js"></script>
```

Вариант с локальным файлом:

```
<script src="jquery-3.1.0.min.js"></script>
```

Дожидаемся загрузки страницы, перед тем как начать взаимодействовать с DOM

```
$(onLoad);
```

```
function onLoad() {  
    var $button = $('button');  
}
```

Подписываемся на нажатие кнопки

```
$button.click(function(e) {  
    e.preventDefault();  
})
```

Здесь в функцию `click()` передаётся другая функция (анонимная), она будет вызвана тогда, когда пользователь нажмёт на кнопку `$button`. Переменная `e` (`event`) - содержит дополнительную информацию о произошедшем событии (нажатии на кнопку), такую как: координаты, целевой объект.

Отправка AJAX-запроса

```
$.post('/api/message', data)
  .done(function(response) {
    console.log(response);
  })
  .fail(function(error) {
    alert('Failed to create message');
    console.log(error);
  });
```

Этот код отправит на сервер, с которого получена текущая страница запрос на ресурс `/api/message`. В этом запросе на сервер будет передан объект `data`. Можно передать произвольный JSON-документ. В случае успешного запроса (если сервер ответить кодом 200), выполнится callback, переданный в функцию `done`. В противном случае выполнится callback, переданный в `fail`.

Про Ajax вообще можно почитать здесь: <https://ru.wikipedia.org/wiki/AJAX>

Про Ajax применительно к jQuery: <http://api.jquery.com/category/ajax/shorthand-methods/>

Вспомогательные материалы

1. Неплохой online-учебник по Javascript на русском языке: <https://learn.javascript.ru>
2. Большой сборник приложений, демонстрирующих возможности JS (в основном WebGL): <https://www.chromeexperiments.com>
3. Подробный справочник по JS:
https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects
4. Документация по jQuery:
<http://api.jquery.com>
5. Документация по flotjs:
<https://github.com/flot/flot/blob/master/API.md>

Контрольные вопросы

1. Отличия JS от C++ и C#
2. Типы данных в JS
3. Функциональное программирование в JS
4. Особенности асинхронного программирования