

Университет ИТМО
Мегафакультет компьютерных технологий и управления
Факультет программной инженерии и компьютерной техники

Лабораторная работа №1 по веб-программированию

Вариант: 211006

Выполнил: Горшков Артем Владимирович

Преподаватель: Киреев Валерий Юрьевич

Санкт-Петербург

2019 год

Вариант 211006

Разработать PHP-скрипт, определяющий попадание точки на координатной плоскости в заданную область, и создать HTML-страницу, которая формирует данные для отправки их на обработку этому скрипту.

Параметр R и координаты точки должны передаваться скрипту посредством HTTP-запроса. Скрипт должен выполнять валидацию данных и возвращать HTML-страницу с таблицей, содержащей полученные параметры и результат вычислений - факт попадания или непадения точки в область.

Кроме того, ответ должен содержать данные о текущем времени и времени работы скрипта.

Разработанная HTML-страница должна удовлетворять следующим требованиям:

Для расположения текстовых и графических элементов необходимо использовать табличную верстку.

Данные формы должны передаваться на обработку посредством GET-запроса.

Таблицы стилей должны располагаться в самом веб-документе.

При работе с CSS должно быть продемонстрировано использование селекторов псевдоэлементов, селекторов идентификаторов, селекторов атрибутов, селекторов псевдоклассов а также такие свойства стилей CSS, как наследование и каскадирование.

HTML-страница должна иметь "шапку", содержащую ФИО студента, номер группы и номер варианта. При оформлении шапки необходимо явным образом задать шрифт (monospace), его цвет и размер в каскадной таблице стилей.

Отступы элементов ввода должны задаваться в пикселях.

Страница должна содержать сценарий на языке JavaScript, осуществляющий валидацию значений, вводимых пользователем в поля формы. Любые некорректные значения (например, буквы в координатах точки или отрицательный радиус) должны блокироваться.

Вопросы к защите лабораторной работы:

1. Протокол HTTP. Структура запросов и ответов, методы запросов, коды ответов сервера, заголовки запросов и ответов.
2. Язык разметки HTML. Особенности, основные теги и атрибуты тегов.
3. Структура HTML-страницы. Объектная модель документа (DOM).
4. HTML-формы. Задание метода HTTP-запроса. Правила размещения форм на страницах, виды полей ввода.
5. Каскадные таблицы стилей (CSS). Структура - правила, селекторы. Виды селекторов, особенности их применения. Приоритеты правил. Преимущества CSS перед непосредственным заданием стилей через атрибуты тегов.
6. LESS, Sass, SCSS. Ключевые особенности, сравнительные характеристики. Совместимость с браузерами, трансляция в "обычный" CSS.
7. Клиентские сценарии. Особенности, сферы применения. Язык JavaScript.
8. Версии ECMAScript, новые возможности ES6 и ES7.

9. Синхронная и асинхронная обработка HTTP-запросов. AJAX.
10. Библиотека jQuery. Назначение, основные API. Использование для реализации AJAX и работы с DOM.
11. Реализация AJAX с помощью SuperAgent.
12. Серверные сценарии. CGI - определение, назначение, ключевые особенности.
13. FastCGI - особенности технологии, преимущества и недостатки относительно CGI.
14. Язык PHP - синтаксис, типы данных, встраивание в веб-страницы, правила обработки HTTP-запросов. Особенности реализации принципов ООП в PHP.

Исходный код программы:

Расположен на: https://github.com/gorshik/WEB_Lab

Разработанная страничка:

https://se.ifmo.ru/~s265087/lab1_web/WEB_Lab/input.php

Php, javascript:

```
<?php
    $start = microtime(true);
    if (isset($_GET['X']) && isset($_GET['Y']) && isset($_GET['R'])) {
        $Y = $_GET['Y'];
        $Y = str_ireplace(",", ".", $Y);
        $Y_Num = substr($_GET['Y'], 0, 10);
        $X = $_GET['X'];
        $R = $_GET['R'];
        $validX = false;
        $validR = false;
        for ($i = -2; $i <= 2; $i += 0.5) {
            if (strcasecmp(strval($i), $X) == 0) $validX = true;
        }
        for ($i = 1; $i <= 5; $i += 1) {
            if (strcasecmp(strval($i), $R) == 0) $validR = true;
        }
        if (!is_numeric($Y)) {
            echo '<tr id="servErr"><td>Y не число!</td></tr>';
        } elseif (!is_numeric($R)) {
            echo '<tr id="servErr"><td>R не число!</td></tr>';
        } elseif (!is_numeric($X)) {
            echo '<tr id="servErr"><td>X не число!</td></tr>';
        } elseif ($Y_Num <= -5 || $Y_Num >= 3) {
            echo '<tr id="servErr"><td>Y не в диапазоне</td></tr>';
        } elseif ($validX == false) {
            echo '<tr id="servErr"><td>Недопустимое значение X</td></tr>';
        } elseif ($validR == false) {
            echo '<tr id="servErr"><td>Недопустимое значение R</td></tr>';
        } elseif (!(($Y_Num <= -5 || $Y_Num >= 3))){
            echo "<tr><td><table id='answer'><tr
class='bold'><td>X</td><td>Y</td><td>R</td><td>Ответ</td></tr>";
```

```

$Ans = "<tr><td>" . $X . "</td><td class='word-break'>" . $Y . "</td><td>" . $R .
"</td><td>";
if ($X >= 0) {
    if ($Y_Num >= 0) {
        if ($X <= $R / 2 && $Y_Num <= $R) {
            $Ans = $Ans . "Точка в зоне";
        } else {
            $Ans = $Ans . "Точка не в зоне";
        }
    } else {
        if ($X ^ 2 + $Y_Num ^ 2 <= $R ^ 2) {
            $Ans = $Ans . "Точка в зоне";
        } else {
            $Ans = $Ans . "Точка не в зоне";
        }
    }
} else {
    if ($Y >= 0) {
        $Ans = $Ans . "Точка не в зоне";
    } else {
        if ($Y_Num + $X > -$R / 2) {
            $Ans = $Ans . "Точка в зоне";
        } else {
            $Ans = $Ans . "Точка не в зоне";
        }
    }
}
$Ans = $Ans . "</tr></table>";
echo $Ans;
echo 'Время выполнения скрипта: ' . round((microtime(true) - $start) * pow(10, 6),
3) . ' микросек.';
echo '</td></tr><tr><td><div id="time">';
echo date(DATE_RFC850) . '</div></td></tr>';
}
}
?>
<script type="text/javascript">
    let er = document.getElementById("error");
    let textField = document.getElementById("textfieldY");
    let Xfield = document.getElementsByName("X");
    let Rfield = document.getElementsByName("R");

    function valid() {
        let validX = false;
        let validR = false;
        Xfield.forEach(function (button) {
            if (button.checked) validX = true;
        });
        Rfield.forEach(function (button) {
            if (button.checked) validR = true;
        });
    }

```

```

    let value = textField.value.substring(0, 10);
    value = value.replace(",", ".");
    if (value === "" || isNaN(value) || value <= -5 || value >= 3) {
    if (document.getElementById("servErr") != null)
document.getElementById("servErr").innerHTML = "";
    er.innerHTML = "Значение Y должно быть в диапазоне (-5;3)";
    textField.style.borderColor = "red";
    return false;
    }
    if (!validX) {
    if (document.getElementById("servErr") != null)
document.getElementById("servErr").innerHTML = "";
    er.innerHTML = "Укажите значение X";
    return false;
    }
    if (!validR) {
    if (document.getElementById("servErr") != null)
document.getElementById("servErr").innerHTML = "";
    er.innerHTML = "Укажите значение R";
    return false;
    }
    return true;
}
</script>

```

Вывод:

В ходе выполнения данной лабораторной работы я узнал как работает протокол http, освоил правила задания селекторов в css, написал скрипт на php и javascript валидатор для своей html страницы.