

**РОСЖЕЛДОР**

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования**

**«Ростовский государственный университет путей сообщения»**

**(ФГБОУ ВО РГУПС)**

---

В.Л. Панасов

**ТЕХНИКА WEB-ПРОГРАММИРОВАНИЯ**

Учебно-методическое пособие

Ростов-на-Дону  
2016

УДК 681.3.06(07) + 06

Рецензент – кандидат технических наук, доцент Т.М. Линденбаум

**Панасов, В.Л.**

Техника web-программирования: учебно-методическое пособие / В.Л. Панасов; ФГБОУ ВО РГУПС. – Ростов н/Д, 2016. – 16 с.: ил. – Библиогр.: с. 16.

Рассматриваются следующие языки web-программирования: HTML 5, CSS 3 Javascript. Содержится краткое описание среды отладчика web-приложений, встроенного в Mozilla Firefox. Описаны приемы автоматизированной разработки интерфейсов web-приложений и отладки Javascript-кода. Описание выполнено на конкретном примере. Также содержатся задания к лабораторным работам по данному направлению.

Предназначено студентам магистратуры, бакалавриата и специалитета для углубленного изучения web-программирования на аудиторных занятиях и самостоятельного изучения материала по дисциплинам «Информатика», «Информационные технологии», «Компьютерные системы и сети» и спецкурсов.

*Учебное издание*

**Панасов** Виктор Леонидович

## **ТЕХНИКА WEB-ПРОГРАММИРОВАНИЯ**

Печатается в авторской редакции  
Технический редактор Т.И. Исаева

Подписано в печать 06.04.16. Формат 60×84/16.  
Бумага газетная. Ризография. Усл. печ. л. 0,93.  
Тираж        экз. Изд. № 5033. Заказ        .

Редакционно-издательский центр ФГБОУ ВО РГУПС.

---

Адрес университета: 344038, г. Ростов н/Д, пл. Ростовского Стрелкового Полка  
Народного Ополчения, 2.

© Панасов В.Л., 2016  
© ФГБОУ ВО РГУПС, 2016

## ПОДГОТОВКА СРЕДЫ WEB-ПРОГРАММИСТА

При разработке web-приложений минимальным набором инструментов является:

- 1 Блокнот.
- 2 Любой браузер.

Этих программ достаточно для разработки самых простых страниц. Создание более сложных страниц с таким набором инструментов превращается в непростую задачу, требующую много времени. Поэтому профессионалы пользуются дополнительным инструментарием:

- 1 Редактор кода с подсветкой синтаксиса и простым анализатором кода.
- 2 Браузер с отладчиком.

Одними из популярных программ среди профессиональных разработчиков являются Notepad++ в качестве редактора кода и встроенный отладчик Mozilla Firefox.

Рассмотрим работу с ними на примере создания web-страницы простого калькулятора факториала. На странице должна быть панель с несколькими элементами управления: поле для ввода аргумента, поле для вывода результата и кнопка, выполняющая расчет.

Чтобы максимально автоматизировать процесс разработки, будем создавать страницу поэтапно. Сначала сделаем заготовку HTML-кода без конкретных стилей расположения и форматирования элементов. Потом откроем ее в отладчике браузера Firefox и постепенно добавим все необходимые стили. После этого скопируем стили обратно в исходный код HTML-страницы.

Первоначальную заготовку наберем с помощью блокнота и сохраним под именем 1.htm:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<style>
[name=panel] {}
[name=parameter] {}
[name=resultat]{}
[name=fakt_knopka] {}
</style>
</head>
<body>
<div name="panel">
<input type="text" name="parameter" value=""/>
<div name="resultat"></div>
<a name="fakt_knopka">Посчитать факториал</a>
</div>
</body>
</html>
```

Здесь мы сделали все основные теги для элементов управления. Присвоили всем им имена через атрибут name, которые позволят нам подключить к ним стили, а позже позволят получать нам ссылки на их объекты в коде Javascript. Чтобы страница отвечала стандартам, мы вначале добавили директиву `<!DOCTYPE html>`. Чтобы русские буквы правильно отображались во всех браузерах, добавили мета-тег `<meta charset="utf-8">`, указывающий на использование кодировки UTF-8.

Чтобы можно было редактировать готовые классы стилей в отладчике Firefox, мы сделали их заготовки в теге `<style>`. Мы создали заготовки классов для каждого элемента управления.

Если этот код открыть в Notepad++, то можно увидеть подсветку синтаксиса разными цветами:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <style>
6 [name=panel] {}
7 [name=parameter] {}
8 [name=resultat] {}
9 [name=fakt_knopka] {}
10 </style>
11 </head>
12 <body>
13 <div name="panel">
14 <input type="text" name="parameter" value=""/>
15 <div name="resultat"></div>
16 <a name="fakt_knopka">Посчитать факториал</a>
17 </div>
18 </body>
19 </html>
```

Рис. 1. Подсветка синтаксиса в Notepad++

Если такую страницу открыть в браузере, то мы увидим следующее (рис. 2):

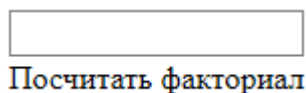


Рис. 2. Заготовка страницы в браузере

На данном этапе не следует беспокоиться о том, что страница выглядит совсем не так, как нужно. Например, мы не видим поля вывода результата, а вместо кнопки виден просто текст. Чтобы увидеть все элементы, откроем окно отладчика. Для этого нажмем клавишу F12 (рис. 3):

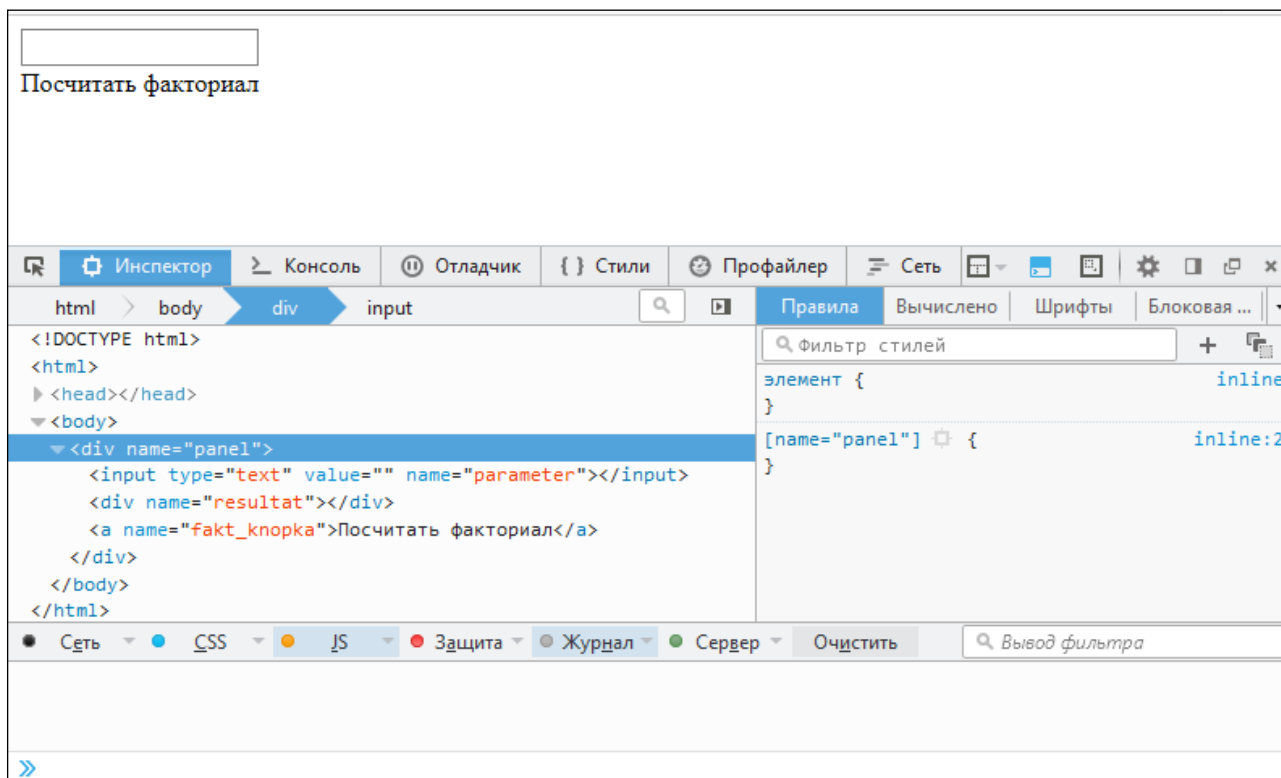


Рис. 3. Отладчик в режиме прикрепления к низу окна

Окно отладчика открывается в одном из трех режимов: прикрепленное к низу окна браузера, прикрепленное к правому краю окна браузера или в отдельном окне.

Первый и последний режим имеет примерно одинаковое расположение элементов (см. рис. 3). В режиме прикрепления к правому краю внутренние элементы отладчика располагаются вертикально (рис. 4). Этот режим удобен в тех случаях, когда Web-приложение вытянуто по вертикали. Для переключения режимов используется кнопка «Прикрепить», расположенная на узкой панели инструментов сверху отладчика. На рис. 4 эта кнопка выделена красным кружком.

В отладчике имеются три основные области:

- 1) область структуры страницы расположена слева;
- 2) область свойств выбранного объекта расположена справа;
- 3) область консоли расположена снизу. Здесь отображаются сообщения об ошибках, и доступна прямая работа с объектами DOM через написание простых скриптов.

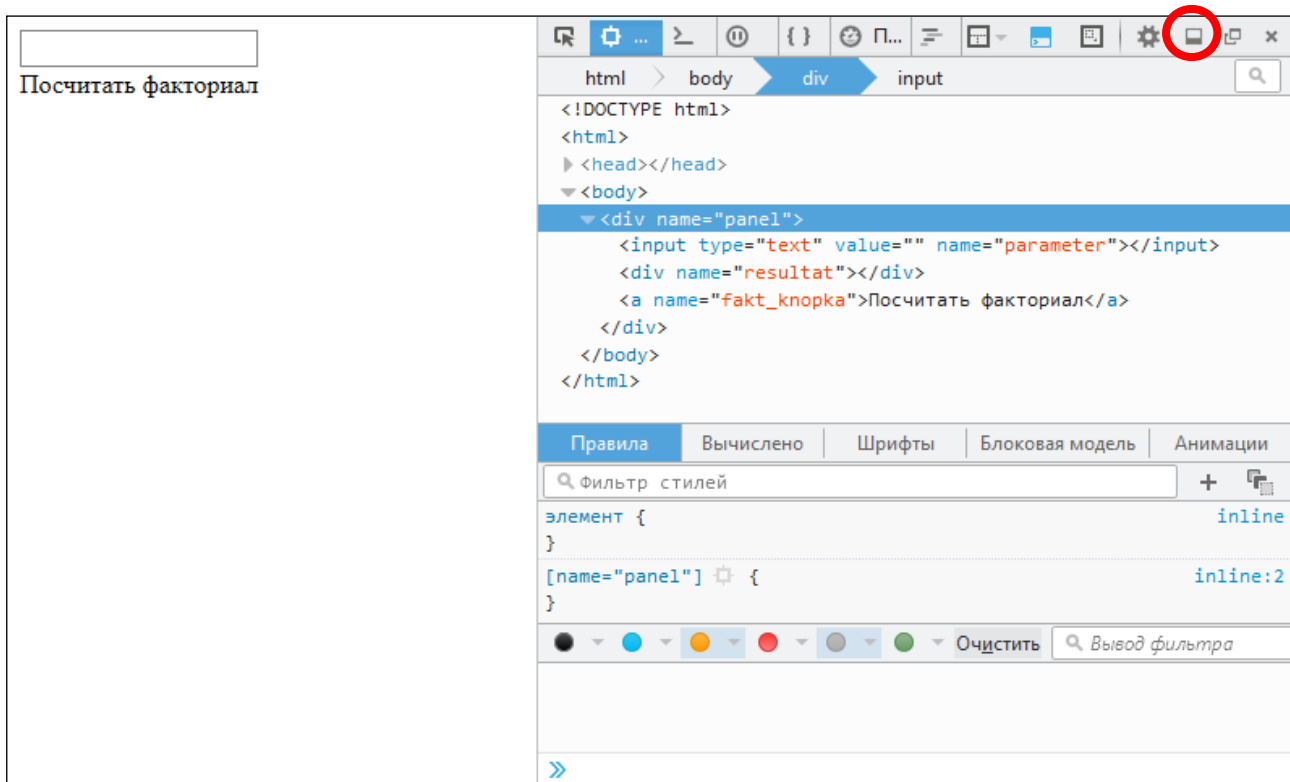


Рис. 4. Отладчик в режиме прикрепления к правому краю окна

Если область консоли скрыта, то ее можно открыть, нажав кнопку «Показать/скрыть консоль» на панели инструментов отладчика.

Работа с отладчиком возможна в нескольких режимах, которые переключаются кнопками на панели инструментов. Основные режимы:

- режим «Инспектор» позволяет видеть структуру DOM документа;
- режим «Отладчик» позволяет отлаживать Javascript-код;
- режим «Консоль» позволяет раскрыть консоль шире;
- режим «Стили»;
- режим «Сеть» позволяет анализировать взаимодействие браузера с сервером.

Чтобы добавить стили расположения элементов в нашей заготовке страницы, потребуется *Инспектор*. В левой части структуры DOM можно увидеть иерархическую структуру тегов нашей страницы. Если, например, выбрать тег панели `<div name="panel">`, то можно справа увидеть его стили. Если еще навести на него мышку, то на самой странице можно увидеть голубой прямоугольник, обозначающий область, занимаемую этим тегом, и всплывающую подсказку с размерами этой области на странице (рис. 5):

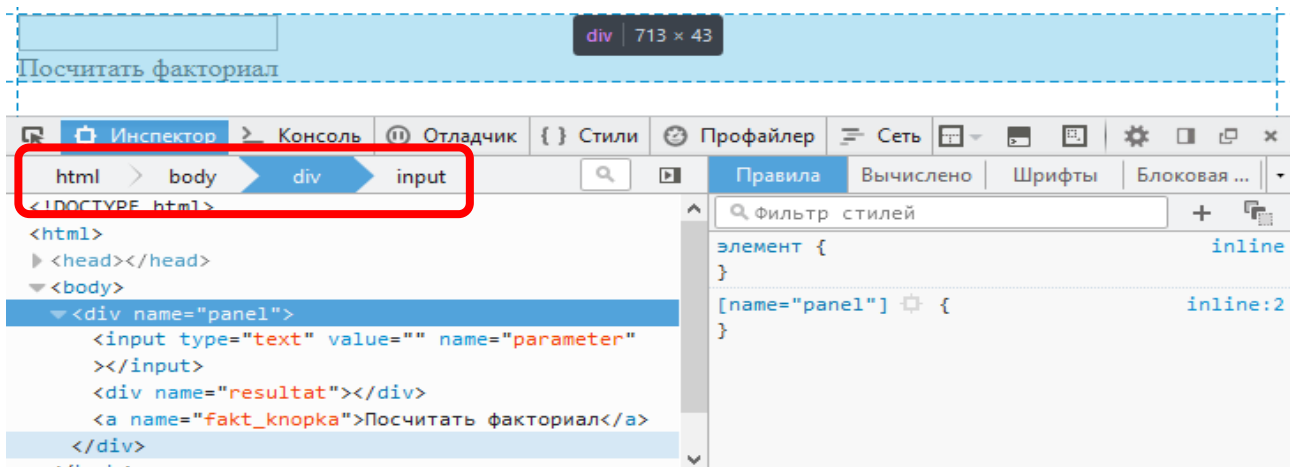


Рис. 5. Навигация по объектам DOM в *Инспекторе*

Выбирать теги можно и непосредственно на странице. Для этого нужно нажать кнопку «Выбрать элемент со страницы», отмеченную на рис. 5 красным кружком. После этого можно мышкой выделить любой видимый объект. При щелчке на нем он будет выбран и в *Инспекторе*. Также в выделенной красным овалом на рис. 5 области можно видеть для выбранного тега все родительские теги. Щелчком мышки можно легко переходить к ним и возвращаться назад.

Теперь можно справа увидеть набор стилей, применимых к этому тегу панели. Здесь мы видим два набора стилей: «элемент» отвечает за inline-стили тега, и [name="panel"] отвечает за класс, который мы предварительно описали в разделе <style>. Добавим несколько стилей, чтобы увидеть нашу панель и придать ей более привлекательный вид (рис. 6):

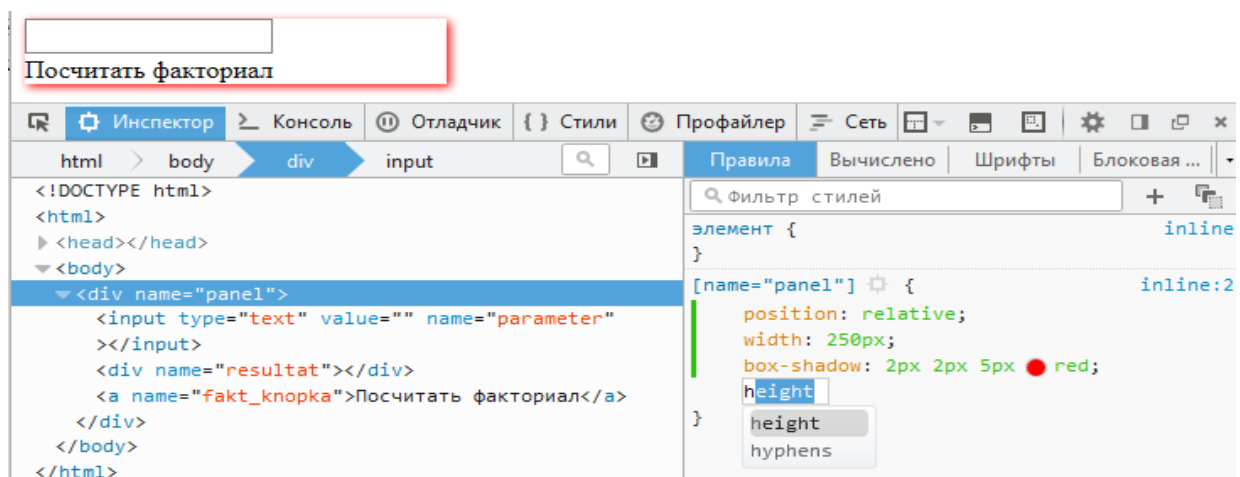


Рис. 6. Добавление стилей

Чтобы добавить стили, для начала нужно щелкнуть внутри фигурных скобок «{ }». Потом набрать имя стиля, нажать клавишу Tab и набрать его значение. Чтобы продолжить остальные стили, нужно нажимать клавишу Tab. При

наборе имен и значений стилей появляются всплывающие подсказки. Можно просто выбрать подходящий вариант, не набирая всего текста.

После первоначального ввода значений стилей появляется возможность более точной настройки. Например, чтобы точнее подстроить размер панели, можно щелкнуть на значении ширины `width`, и далее использовать клавиши стрелок *Вверх/Вниз* для изменения его по одному пикселю. При этом можно сразу видеть результат на странице. Если при нажатии клавиш стрелок удерживать `Shift`, то шаг подстройки будет равен 10 пикселям. Также можно точнее подстроить цвет, щелкнув на красном кружке у стиля `box-shadow` (рис. 7):

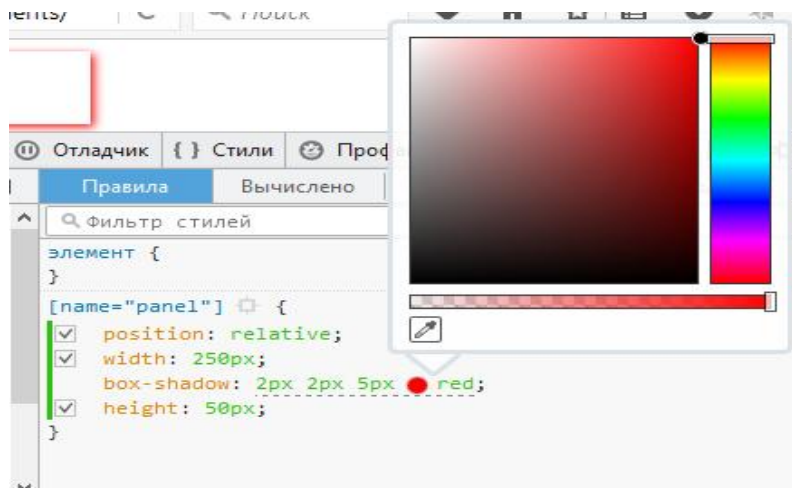


Рис. 7. Редактор цветов

Выбирая поэтапно каждый тег и настраивая его стили в правом окне, можно добиться следующего внешнего вида страницы (рис. 8):

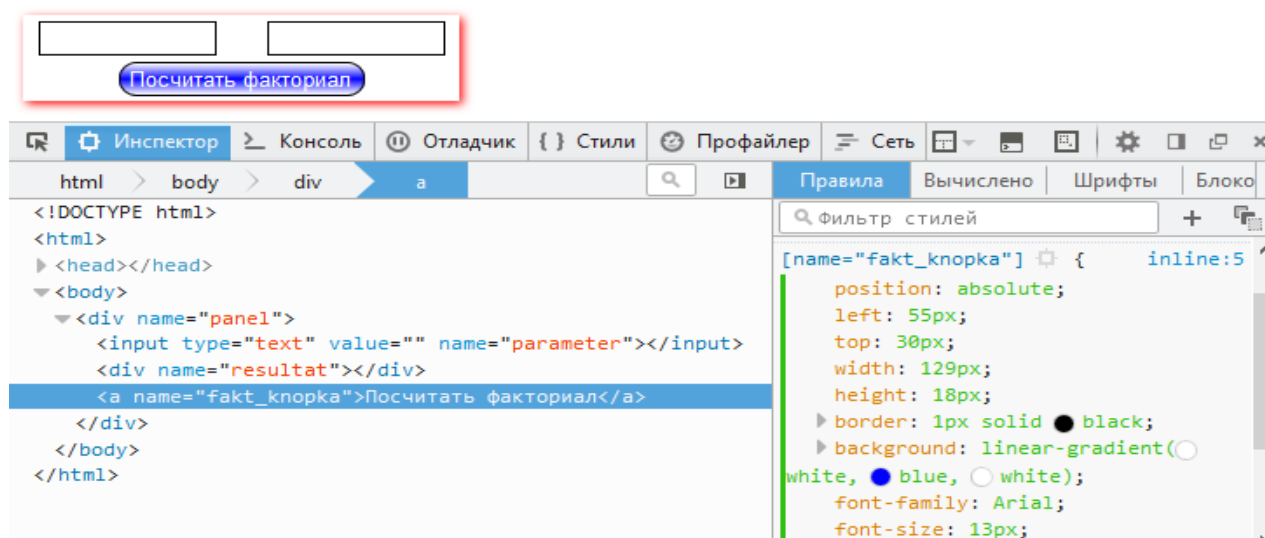


Рис. 8. Страница после добавления стилей элементов

Теперь необходимы перенести созданные стили в исходный файл. Для этого нужно выбрать режим отладчика «Стили» (рис. 9):



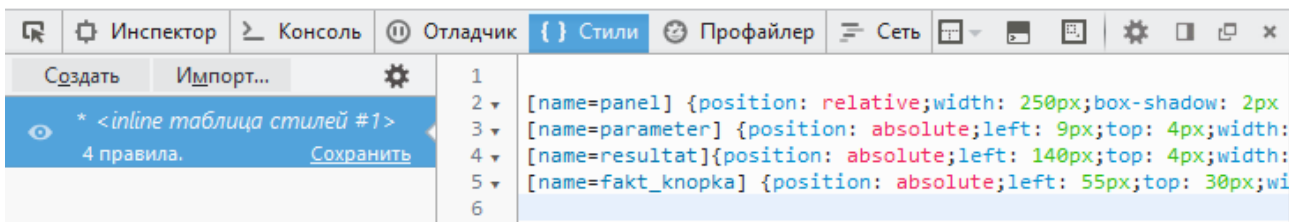


Рис. 9. Режим отображения стилей

Здесь можно выделить нужные стили данной страницы и скопировать их в исходный файл через Notepad++. Чтобы убедиться, что все работает правильно, следует обновить страницу в Firefox. Если внешний вид не изменился, значит, все сохранилось верно.

Чтобы повысить степень интерактивности, добавим несколько визуальных эффектов. Сделаем так, чтобы при наведении мышки на поле ввода аргумента его фон становился желтым. Это послужит подсказкой пользователям, что сюда можно что-либо ввести. Еще сделаем эффект «надавливания» для кнопки «Посчитать факториал». Для этого сначала добавим заготовки классов для псевдо-стилей `:hover` и `:active`:

```

[name=parameter]:hover {}
[name=resultat]:active {}

```

Далее наполним содержимым в отладчике. Чтобы увидеть псевдо-стили в отладчике, нужно выбрать нужный тег, и сначала нажать в правой панели на кнопку «Показать/скрыть панель псевдо-классов», отмеченную красным кружком на рис. 10:

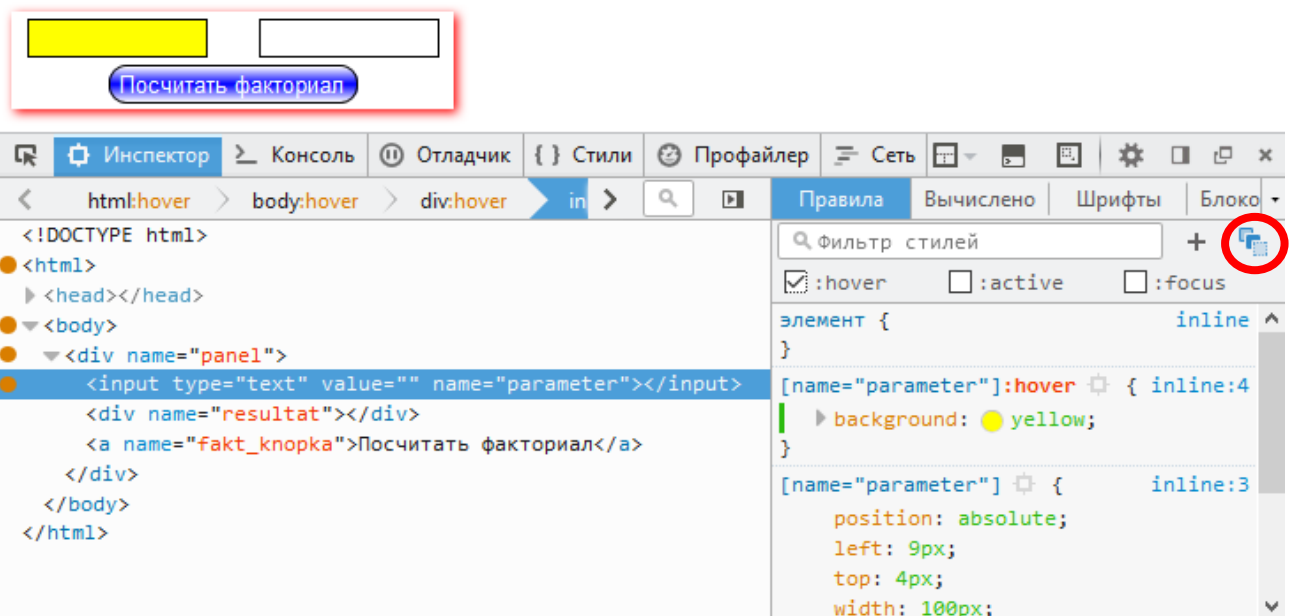


Рис. 10. Настройка псевдо-стилей

Далее в панели нужно выбрать нужные псевдо-стили. При этом псевдо-классы отображаются в правой панели, и становятся активными на самой странице. Поэтому можно прежним способом наполнить их стилями, и сразу увидеть результат на странице. Например, можно добавить стиль `background:yellow` и сразу увидеть желтый фон у поля аргумента, как будто на него наведена мышка. Таким же способом можно добавить и стиль `background:linear-gradient(blue, white, blue)` для кнопки, чтобы получить эффект «надавливания». В конце важно не забыть скопировать стили в исходный файл.

Теперь, когда внешний вид страницы закончен, можно перейти к разработке и отладке кода. Для начала добавим событие для кнопки:

```
<a name="fakt_kнопка" onclick="fakt()">Посчитать факториал</a>
```

Потом добавим саму функцию расчета факториала в раздел `<script>`:

```
function fakt() {
    var n = parseInt(document.querySelector('[name=parameter]').value);
    var p = 1;
    for(var i = 1; i <= n; i++) {
        p = p * i;
    }
    document.querySelector('[name=resultat]').innerHTML = p;
}
```

Обновим браузер и проверим работу кнопки. Чтобы посмотреть код в браузере, и получить возможность его отладки, можно сразу открыть режим «Отладчик»:

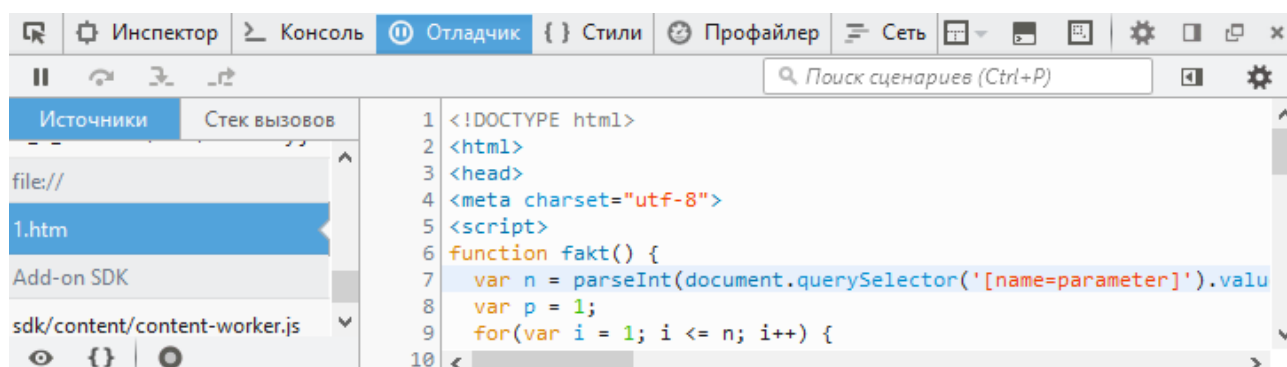


Рис. 11. Режим отладки кода Javascript

В этом режиме слева показывается список файлов, содержащих Javascript код. В нем нужно выбрать файл 1.htm. Слева мы увидим код нашего файла.

Чтобы увидеть, как выполняется код нашей функции, нужно поставить точку останова. Для этого нужно щелкнуть мышкой на номере строки, где должна быть точка останова. Появится синий кружок. Далее нужно просто щелкнуть на кнопке «Посчитать факториал», чтобы попасть в состояние отладки (рис. 12):

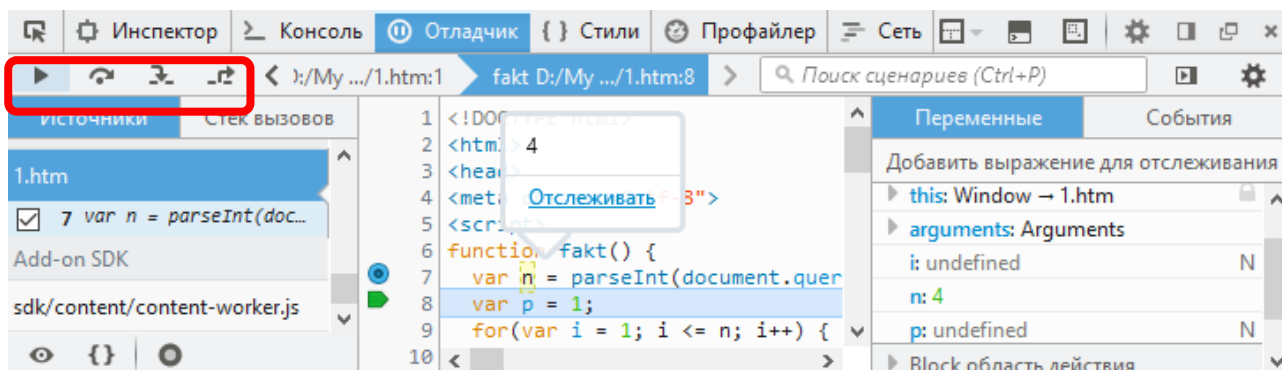


Рис. 12. Срабатывание точки останова в отладчике кода

Для отслеживания хода выполнения скрипта используются кнопки слева под панелью инструментов, выделенные на рис. 12 красным овалом, или клавиши: F10 – перешагнуть на одну строчку; F11 – зайти внутрь выделенной функции; F8 – продолжить скрипт в обычном режиме без пошагового выполнения.

На рис. 12 показано состояние после одного шага. Здесь зеленой стрелкой рядом с номером строки обозначена текущая строка, которая будет выполняться при следующем продвижении отладчика. Справа открывается панель всех переменных, доступных в данный момент. Здесь их можно не только наблюдать, но и изменять. Также можно навести курсор мышки на любую переменную прямо в коде в центральном окне, и увидеть всплывающую подсказку с ее текущим значением.

Обратим внимание, что в левом окне «Исходники» под именем файла 1.htm появилась пометка с текстом строки, на которой стоит точка останова. Если снять пометку, то точка останова временно «выключится». Это можно использовать во время отладки для временного пропуска некоторых точек останова без их полного удаления.

Левое и правое окно также имеют по дополнительной панели (рис. 13):

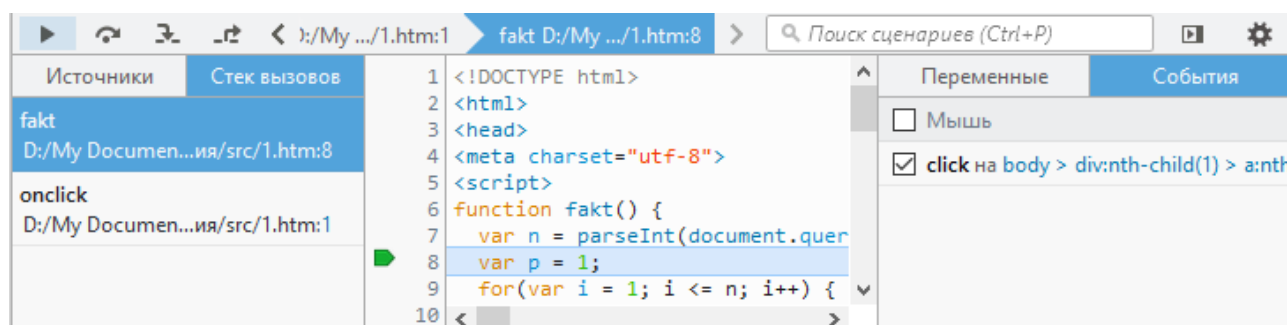


Рис. 13. Стек вызовов и список событий отладчика кода

В левом окне можно открыть панель «Стек вызовов», которая отображает «предысторию» вызова данной функции, т. е. каким путем программа пришла в данную функцию, начиная от исходного события. В данном случае – от события `onclick` на кнопке «Посчитать факториал». Можно выбрать любую функцию в этом списке, чтобы увидеть, из какой именно ее строки был сделан этот вызов.

В правом окне можно открыть панель «События». Здесь показан список всех событий, используемых на странице. Если поставить пометку напротив события, то браузер остановится при его срабатывании. Таким образом, можно не искать вручную обработчик события, а заставить браузер показать его нам.

Другим удобным способом нахождения нужных функций и переменных в исходном коде является строка «Поиск сценариев», расположенная под панелью инструментов (рис. 14):

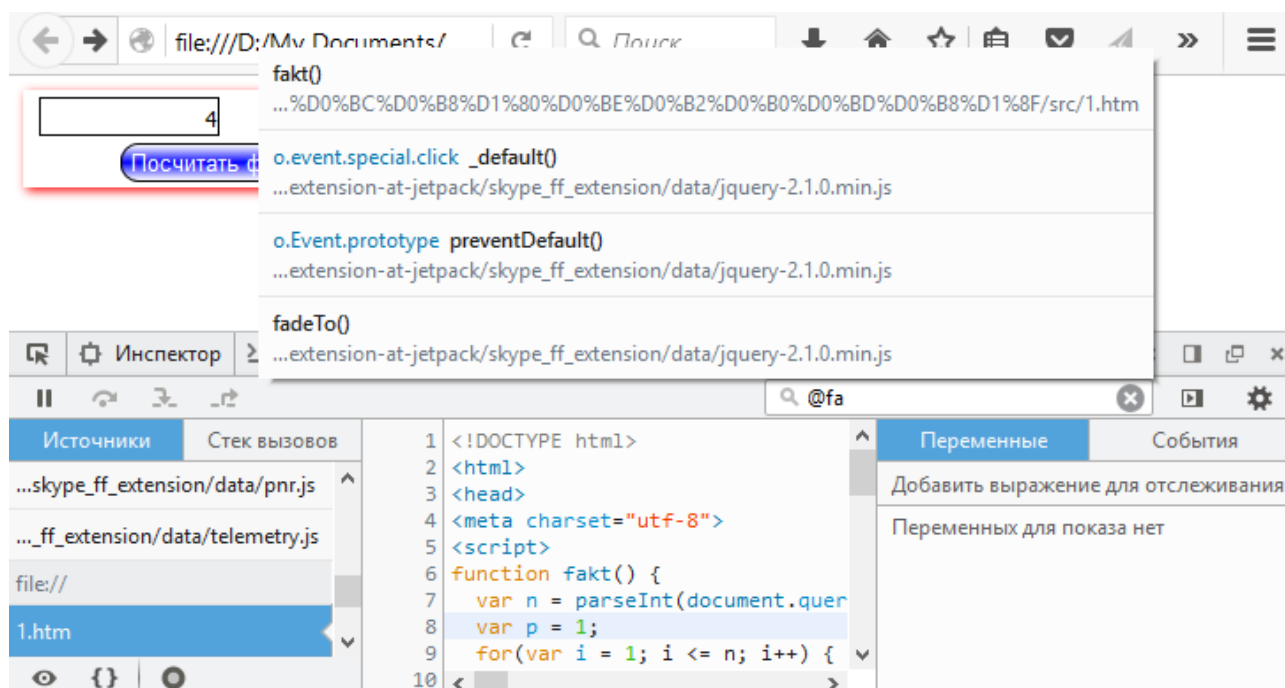


Рис. 14. Поиск функций в отладчике кода

Чтобы найти функцию, достаточно ввести символ «@» и первые несколько букв имени функции. Появится список имен функций, начинающихся с этих букв, найденных во всех загруженных файлах. Можно выбрать любую функцию в этом списке, чтобы перейти к ее исходному коду. Если нужно найти какие-либо переменные в правой панели «Переменные», то нужно начать с символа «\*» и набрать первые несколько букв искомой переменной.

## **ЗАДАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ**

### **Лабораторная работа № 1**

Создать пример, описанный в первой главе. Настроить среду разработки. Прodelать все операции с отладчиком, описанные в первой главе.

### **Лабораторная работа № 2**

1 Создать XML-файл с анкетами 5 человек. В состав анкеты входят следующие данные: фамилия, имя, отчество, пол, дата рождения, должность, ученая степень (кандидат наук, доктор наук, профессор).

2 Разработать страницу, на которой будет список всех людей из XML-файла по Ф.И.О. Список должен отображаться сразу после открытия страницы.

3 При выборе любого человека в анкете должна появляться «всплывающая панель» «Анкета» с подробными данными. Панель должна быть модальной, а элементы страницы под ней должны затемняться и быть неактивными.

4 Для отображений пола следует применить нестандартные (рисованные) радиокнопки. Для отображения ученой степени следует создать нестандартный элемент управления типа «выпадающий список».

5 Все панели должны иметь бордюры с тенями. Поля ввода и кнопки должны меняться при наведении мышки. Кнопку закрытия панели «Анкета» следует сделать с градиентной заливкой и закругленным бордюром.

### **Лабораторная работа № 3**

1 Создать 5 XML-файлов. В каждом файле должен быть набор из 20 произвольных вещественных чисел. Создать еще один XML-файл со списком имен первых пяти файлов и назвать его list.xml.

2 Сделать панель со следующими элементами управления: нестандартный выпадающий список с именами файлов, загружаемый из list.xml; нестандартный выпадающий список со списком статистических операций: «среднее арифметическое», «минимальное значение», «максимальное значение»; кнопка «Рассчитать».

3 По нажатию кнопки «Рассчитать» должна появляться новая модальная панель с одним полем с результатом расчетов. Под панелью все должно затемняться и становиться неактивным.

4 Панели должны иметь бордюры с тенями. Кнопки и поля должны меняться при наведении мышки.

### Лабораторная работа № 4

1 В рабочую папку сложить 9 произвольных картинок. Создать XML-файл fotos.xml со списком файлов картинок, в котором они будут условно разбиты на 3 группы: «Семья», «Работа», «Друзья».

2 Создать панели для трех закладок «Семья», «Работа», «Друзья».

3 На каждой панели должен быть маленькие иконки картинок из соответствующей категории. Список по категориям должен загружаться из файла fotos.xml.

4 При щелчке на любую картинку должна открываться панель с крупным изображением картинки. У панели должны быть стрелки влево и вправо, позволяющие переключать картинки в пределах данной группы, и кнопка закрывания панели.

5 Панели должны иметь бордюры с тенями. Закладки и кнопки должны меняться при наведении мышки. Кнопки должны иметь градиентную заливку и закругленные бордюры.

### Лабораторная работа № 5

1 Создать XML-файл с набором из 30 произвольных точек, представленных координатами  $(x, y)$ .

2 На странице сделать тег `<canvas>`, в котором будет отображаться график функции из файла.

3 Пределы графика должны определяться автоматически так, чтобы график занимал всю область.

4 Слева и справа должны быть поля ввода, в которые изначально выставляются рассчитанные пределы по оси  $X$ .

5 Ввод новых значений в поля должен изменять пределы и перерисовывать для них график.

6 Возле полей должны быть кнопки «+» и «-». При их нажатии значение в поле должно увеличиваться или уменьшаться на 1.

7 Кнопки должны меняться при наведении мышки. Кнопки должны иметь градиентную заливку и закругленные бордюры.

### Лабораторная работа № 6

1 Сделать на странице тег `<canvas>` и изобразить в нем с помощью методов рисования поезд, состоящий из локомотива и трех вагонов. Изображение должно быть схематичным и содержать окна, колеса, пантограф, двери и сцепку вагонов.

2 Под рисунком должна быть кнопка «Старт». При ее нажатии поезд начинает двигаться слева направо. После выхода за пределы рисунка он должен снова появляться из левой части.

3 При движении поезда кнопка «Старт» должна меняться на «Стоп». Ее нажатие должно приостанавливать поезд.

4 Сделать бегунок,двигающийся мышкой влево и вправо. Смещение бегунка должно изменять скорость движения поезда. Рядом с ним должны быть подписи: «Медленно» и «Быстро».

5 Кнопки и бегунок должны меняться при наведении мышки. Они также должны иметь градиентную заливку и закругленные бордюры.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 **Мэтьюз, М.** Динамическое веб-программирование (Dynamic Web Programming: A Beginner's Guide) : пер. с англ. / М. Мэтьюз, Д. Кронан. – М. : Эксмо, 2010. – 384 с. : ил., табл.

2 **Основы WEB-технологий** : курс лекций / П.Б. Храмцов, С.А. Брик, А.М. Русак, А.И. Сурин ; ред. П.Б. Храмцов. – М. : Интернет-университет информ. технологий, 2003. – 510 с. : ил., табл.

3 **Панасов, В.Л.** Разработка web-страниц : учебно-метод. пособие / В.Л. Панасов ; РГУПС. – Ростов н/Д, 2008. – 44 с. : ил. – Библиогр.: 6 назв.

4 **Панасов, В.Л.** Разработка веб-страниц : учебно-метод. пособие. В 2 ч. Ч. 2 / В.Л. Панасов ; ФГБОУ ВПО РГУПС. – Ростов н/Д, 2013. – 24 с. : ил. – Библиогр.: с. 22.

5 **Панасов, В.Л.** Технология разработки web-страниц : учеб. пособие / В.Л. Панасов ; РГУПС. – Ростов н/Д, 2007. – 89 с. : ил. – Библиогр. : 8 назв.

6 **Никсон, Р.** Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Робин Никсон. – СПб. : Питер, 2016. – 688 с.

## ОГЛАВЛЕНИЕ

Подготовка среды web-программиста .....	3
Задания к лабораторным работам .....	13
Лабораторная работа № 1 .....	13
Лабораторная работа № 2 .....	13
Лабораторная работа № 3 .....	13
Лабораторная работа № 4 .....	14
Лабораторная работа № 5 .....	14
Лабораторная работа № 6 .....	14
Библиографический список .....	16