

Лабораторная работа №5

Оглавление

Теория. Конструкции jQuery2

 Основные функции jQuery2

 Селекторы2

 jQuery: Атрибуты элементов5

 Манипуляции с элементами DOM6

 Таблицы стилей CSS7

 События jQuery8

 Эффекты jQuery10

 jQuery AJAX11

 Полезные ссылки11

Задание. Добавление скриптов на страницу12

Практика. Пример использования jQuery13

Теория. Конструкции jQuery

Основные функции jQuery

`$(sel)` – Функция jQuery, где «sel» - селектор.

`$(sel, context)` – Функция jQuery, где «sel» - селектор, а context – корневой (контекстный) элемент поиска. Так, `$("div", "body")` возвращает элементы div тела страницы.

Доступ к объектам jQuery

Выполняет функцию для каждого элемента набора

```
$(sel).each(function () { /* операторы */ });
```

Возвращает количество элементов набора

```
$(sel).size();
```

Обращается к i-му элементу набора

```
$(sel).get(i);
```

Селекторы

Селектор – конструкция, передаваемая в функцию jQuery для выборки элементов объектной модели документа (Document Object Model, DOM).

Виды селекторов

`#id` - Соответствует единственному элементу с заданным атрибутом id.

`<div id="test"></div> => $("#test")`

`Element` - Соответствует всем элементам с заданным тегом.

`$("div")`

`.class` - Соответствует всем элементам данного класса.

`<p class="test"></p><div class="test"></div><h1 class="test2"></h1> => $(".test")`

`.class.class` - Соответствует всем элементам данных классов. `=> $(".test.test2")`

`[attribute]` – Соответствует всем элементам с указанным атрибутом.

`<div width="200"></div> => $("[width] ")`

Иерархия селекторов

`"ancestor descendant"` - Соответствует всем элементам-потомкам, указанным в аргументе «descendant», элементов, указанных в аргументе «ancestor».

`"parent > child"` Соответствует всем дочерним элементам, указанным в аргументе «child» элемента, указанного в «parent».

`"prev + next"` Соответствует всем элементам, указанным в аргументе «next» которые находятся рядом с элементом, указанным в аргументе «prev».

`"prev ~ siblings"` Поиск элементов после элемента, указанного в аргументе «prev», которые соответствуют фильтрующему селектору «siblings».

Базовые фильтры:

- :first Соответствуют первому выбранному элементу.
- :last Соответствует последнему выбранному элементу.
- :not(селектор) Отфильтровывает все элементы, соответствующие данному селектору.
- :even Поиск четных элементов, начиная с нуля.
- :odd Поиск нечетных элементов, начиная с нуля.
- :eq(индекс) Соответствует единственному элементу по его индексу.
- :gt(индекс) Соответствует всем элементам с индексом выше заданного.
- :lt(индекс) Соответствует всем элементам с индексом ниже заданного.
- :header Соответствует всем заголовкам, например h1, h2, h3 и так далее.
- :animated Поиск анимированных на данный момент элементов.

Фильтры контента:

- :contains(текст) Соответствует элементам, которые содержат заданный текст.
- :empty Соответствует всем элементам, которые не имеют дочерних объектов (включая текстовые узлы).
- :has(селектор) Соответствует элементам, содержащим в себе хотя бы один элемент, который совпадает с указанным селектором.
- :parent Соответствует всем родительским элементам (имеют дочерние элементы, включая текст).

Фильтры видимости на странице:

- :hidden Соответствует всем скрытым элементам.
- :visible Соответствует всем видимым элементам.

Селекторы атрибутов

- [атрибут] Соответствует элементам, которые имеют указанный атрибут.
- [атрибут=значение] Соответствует элементам, которые имеют указанный атрибут с конкретным значением.
- [атрибут!=значение] Соответствует всем элементам, которые либо не имеют указанного атрибута, либо все же имеют, но значение его отличается от заданного.
- [атрибут^=значение] Соответствует элементам, обладающим указанным атрибутом, который начинается с конкретного значения.
- [атрибут\$=значение] Соответствует элементам, обладающим указанным атрибутом, который заканчивается конкретным значением.
- [атрибут*=значение] Соответствует элементам, обладающим указанным атрибутом, который содержит конкретное значение.
- [attributeFilter1][attributeFilter2][attributeFilterN] Поиск элементов, которые соответствуют всем указанным фильтрам атрибутов.

Фильтры дочерних объектов:

:nth-child(индекс/even/odd/формула) Соответствует всем элементам, которые являются “N-ными потомками” либо четными/нечетными “потомками”.

:first-child Соответствует всем элементам, которые являются первыми дочерними объектами у “родителей”.

:last-child Соответствует всем элементам, которые являются последними дочерними объектами у “родителей”.

:only-child Соответствует элементам, которые являются единственными дочерними объектами у “родителей”.

Формы:

:input Соответствует всем элементам input, текстовым полям, элементам выбора и кнопкам.

:text Соответствует всем элементам ввода типа text.

:password Соответствует всем элементам ввода типа password.

:radio Соответствует всем элементам ввода типа radio.

:checkbox Соответствует всем элементам ввода типа checkbox.

:submit Соответствует всем элементам ввода типа submit.

:image Соответствует всем элементам ввода типа image.

:reset Соответствует всем элементам ввода типа reset.

:button Соответствует всем элементам ввода типа button.

:file Соответствует всем элементам ввода типа file.

:hidden Соответствует всем скрытым элементам.

Фильтры форм:

:enabled Соответствует всем активным элементам.

:disabled Соответствует всем неактивным элементам

:checked Соответствует всем отмеченным элементам.

:selected Соответствует всем выбранным элементам.

jQuery: Атрибуты элементов

Атрибут элемента – свойство HTML-элемента, значение которого можно получить или назначить

Атрибут класса

```
<div class="someClass"></div>
```

`$(sel).addClass(класс)` Добавляет указанный(е) класс(ы) к каждому совпавшему элементу.

`$(sel).hasClass(класс)` Возвращает true, если хотя бы один из набора совпавших элементов обладает указанным классом.

`$(sel).removeClass(класс)` Удаляет все или указанный(е) класс(ы) из набора совпавших элементов.

`$(sel).toggleClass(класс)` Добавляет указанный класс к элементу, если его нет, и удаляет указанный класс, если элемент уже обладает таковым.

`$(sel).toggleClass(класс, переключатель)` Добавляет указанный класс, если переключатель установлен в true, и удаляет указанный класс, если переключатель установлен в false.

Прочие атрибуты

`attr(имя)` Получение доступа к свойству первого совпавшего элемента.

Используя этот метод можно легко получить значение свойства первого совпавшего элемента. Если элемент не имеет указанного атрибута, то возвращается undefined (не определено). Атрибутами могут быть: title, alt, src, href, width, style и т.д.

`attr(свойства)` Устанавливает атрибуты всех элементов набора, используя при этом объект, который содержит пары ключ/значение.

`attr(ключ, значение)` Изменяет значение единственного свойства для каждого совпавшего элемента.

`attr(ключ, функция)` Изменяет значение единственного свойства для каждого совпавшего элемента.

Но вместо указания непосредственно значения указывается функция, которая возвращает значение как свой результат.

`removeAttr(имя)` Удаляет указанный атрибут из каждого совпавшего элемента.

Содержимое элемента:

`html()` Получает содержимое HTML (innerHTML) каждого совпавшего элемента. С документами XML использовать данный метод нельзя, но можно использовать с документами XHTML.

`html(val)` Добавляет код HTML для каждого совпавшего элемента. С документами XML использовать данный метод нельзя, но можно использовать с документами XHTML.

`text()` Получает содержимое всех совпавших элементов..

`text(val)` Вставка текста во все совпавшие элементы.

`val()` Получает содержимое атрибута value для первого элемента ввода в наборе.

`val(val)` Устанавливает значение атрибута value для каждого совпавшего элемента в наборе.

`val(val)` Может отмечать значение или делать выбор в таких элементах, как radio, checkbox и select.

Манипуляции с элементами DOM

Вставка внутрь:

`append(контент)` Добавляет контент внутрь каждого элемента набора. Добавляемый контент следует за уже существующим.

`appendTo(селектор)` Добавляет все элементы набора к другому указанному набору элементов. Добавляемые элементы следуют после уже существующих. В jQuery версии 1.3.2 данная функция возвращает все вставленные элементы.

`prepend(контент)` Добавляет контент внутрь каждого элемента набора. Добавляемый контент следует перед уже существующим.

`prependTo(селектор)` Добавляет все элементы набора к другому указанному набору элементов. Добавляемые элементы следуют перед уже существующими. В jQuery версии 1.3.2 данная функция возвращает все вставленные элементы.

Вставка снаружи:

`after(контент)` Вставляет контент после каждого элемента набора.

`before(контент)` Вставляет контент перед каждым элементом набора.

`insertAfter(селектор)` Вставляет все элементы набора в другой указанный набор элементов. Добавляемые элементы следуют после уже существующих. В jQuery версии 1.3.2 данная функция возвращает все вставленные элементы.

`insertBefore(селектор)` Вставляет все элементы набора в другой указанный набор элементов. Добавляемые элементы следуют перед уже существующими. В jQuery версии 1.3.2 данная функция возвращает все вставленные элементы.

«Обертывание»:

`wrap(html)` Вставляет каждый совпавший элемент набора в указанную конструкцию HTML кода.

`wrap(элемент)` Вставляет каждый совпавший элемент набора в указанный элемент.

`wrapAll(html)` Заключает все элементы в наборе в единственную конструкцию HTML.

`wrapAll(элемент)` Заключает все элементы в наборе в единственный элемент.

`wrapInner(html)` Заключает содержимое дочерних объектов каждого элемента из набора совпавших элементов (включая текстовые блоки) в конструкцию HTML.

`wrapInner(элемент)` Заключает содержимое дочерних объектов каждого элемента из набора совпавших элементов (включая текстовые блоки) в элемент DOM.

Замена:

`replaceWith(контент)` Производит замену всех элементов набора элементами HTML или DOM. Возвращает замененный элемент JQuery, который был удален из DOM.

`replaceAll(селектор)` Замещает все элементы, которые соответствуют заданному селектору, указанным(и) элементами.

Удаление:

`empty()` Удаляет все дочерние блоки из каждого элемента в наборе совпавших элементов.

`remove(выражение)` Удаляет все совпавшие элементы из DOM. Эта функция НЕ удаляет элементы из объекта jQuery, позволяя Вам воспользоваться ими позже. Обратите внимание, начиная с версии 1.2.2 эта функция также удаляет все обработчики событий и внутренние закешированные данные.

Копирование:

`clone()` Клонировывает соответствующие элементы DOM и выбирает клонов.

`clone(bool)` Клонировывает соответствующие элементы DOM, а также и все их обработчики событий, и выбирает клонов.

Таблицы стилей CSS

CSS:

`css(имя)` Возвращает свойство стиля для первого совпавшего элемента.

`css(свойства)` Устанавливает свойства стиля CSS всех элементов набора, используя при этом объект, который содержит пары ключ/значение.

`css(имя, значение)` Устанавливает значение одного свойства стиля CSS для всех элементов набора.

Позиционирование:

`offset()` Возвращает текущие значения отступов относительно документа для первого элемента в наборе.

`position()` Возвращает значение позиции элемента сверху и слева относительно отступов его "родителя".

`scrollTop()` Возвращает значение отступа прокрутки сверху для первого элемента в наборе.

`scrollTop(значение)` Устанавливает значение отступа прокрутки сверху для всех элементов набора.

`scrollLeft()` Возвращает значение отступа прокрутки слева для первого элемента в наборе.

`scrollLeft(значение)` Устанавливает значение отступа прокрутки слева для всех элементов набора.

Высота и ширина:

`height()` Возвращает текущее значение высоты в пикселях для первого совпавшего элемента в наборе.

`height(значение)` Устанавливает свойство высоты стиля CSS для каждого элемента в наборе.

`width()` Возвращает текущее значение ширины в пикселях для первого совпавшего элемента в наборе.

`width(значение)` Устанавливает свойство ширины стиля CSS для каждого элемента в наборе.

`innerHeight()` Возвращает значение внутренней высоты (исключая border, но включая padding) для первого совпавшего элемента в наборе.

`innerWidth()` Возвращает значение внутренней ширины (исключая border, но включая padding) для первого совпавшего элемента в наборе.

`outerHeight(margin)` Возвращает значение внешней высоты (включая по умолчанию border и padding) для первого совпавшего элемента в наборе.

`outerWidth(margin)` Возвращает значение внешней ширины (включая по умолчанию border и padding) для первого совпавшего элемента в наборе.

События jQuery

Загрузка страницы:

`$(document).ready(функция)` Добавляет функцию, выполняющуюся всякий раз, когда объектная модель документа (DOM) готова к использованию.

Управление событиями:

`$(sel).bind(тип, данные, функция)` Назначает обработчик к одному или более событиям (например `click`) для каждого совпавшего элемента. Можно связывать также и пользовательские события.

`$(sel).one(тип, данные, данные)` Назначает обработчик одному или более событиям для каждого совпавшего элемента, который выполняется единожды.

`$(sel).trigger(событие, данные)` Вызывает событие для каждого элемента набора.

`$(sel).triggerHandler(событие, данные)` Вызывает все связанные с элементом обработчики (для указанного типа события) БЕЗ выполнения действия браузера по умолчанию, передачи события или “живых” событий.

`$(sel).unbind(тип, функция)` Действие этой функции противоположно функции `bind`: удаляет привязанные события из каждого элемента набора.

«Живые» события:

`$(sel).live(тип, функция)` Привязывает обработчик к событию (например, `click`) для всех текущих и будущих элементов набора. Может также связывать пользовательские события.

`$(sel).die(тип, функция)` Действие данной функции противоположно методу `live` — удаляет привязанные “живые” события.

Взаимодействие:

`$(sel).hover(over, out)` Имитирует эффект `hovering` (отслеживание попадания указателя мыши в пределы объекта).

`$(sel).toggle(fn, fn2, fn3, fn4, ...)` Переключается между двумя или более функциями, вызываемыми поочередно каждым следующим нажатием ЛКМ.

События:

`$(sel).blur()` Вызывает событие `blur` для каждого элемента набора.

`$(sel).blur(функция)` Назначает функцию к событию `blur` для каждого элемента набора.

`$(sel).change()` Вызывает событие `change` для каждого элемента набора.

`$(sel).change(функция)` Назначает функцию к событию `change` для каждого элемента набора.

`$(sel).click()` Вызывает событие `click` для каждого элемента набора.

`$(sel).click(функция)` Назначает функцию к событию `click` для каждого элемента набора.

`$(sel).dblclick()` Вызывает событие `dblclick` для каждого элемента набора.

`$(sel).dblclick(функция)` Назначает функцию к событию `dblclick` для каждого элемента набора.

`$(sel).error()` Вызывает событие `error` для каждого элемента набора.

`$(sel).error(функция)` Назначает функцию к событию `error` для каждого элемента набора.

`$(sel).focus()` Вызывает событие `focus` для каждого элемента набора.

`$(sel).focus(функция)` Назначает функцию к событию `focus` для каждого элемента набора.

`$(sel).keydown()` Вызывает событие `keydown` для каждого элемента набора.

`$(sel).keydown(функция)` Назначает функцию к событию `keydown` для каждого элемента набора.

`$(sel).keypress()` Вызывает событие `keypress` для каждого элемента набора.

`$(sel).keypress(функция)` Назначает функцию к событию `keypress` для каждого элемента набора.

`$(sel).keyup()` Вызывает событие `keyup` для каждого элемента набора.

`$(sel).keyup(функция)` Назначает функцию к событию `keyup` для каждого элемента набора.

`$(sel).load(функция)` Назначает функцию к событию `load` для каждого элемента набора.

`$(sel).mousedown(функция)` Назначает функцию к событию `mousedown` для каждого элемента набора.

`$(sel).mouseenter(функция)` Назначает функцию к событию `mouseenter` для каждого элемента набора.

`$(sel).mouseleave(функция)` Назначает функцию к событию `mouseleave` для каждого элемента набора.

`$(sel).mousemove(функция)` Назначает функцию к событию `mousemove` для каждого элемента набора.

`$(sel).mouseout(функция)` Назначает функцию к событию `mouseout` для каждого элемента набора.

`$(sel).mouseover(функция)` Назначает функцию к событию `mouseover` для каждого элемента набора.

`$(sel).mouseup(функция)` Назначает функцию к событию `mouseup` для каждого элемента набора.

`$(sel).resize(функция)` Назначает функцию к событию `resize` для каждого элемента набора.

`$(sel).scroll(функция)` Назначает функцию к событию `scroll` для каждого элемента набора.

`$(sel).select()` Вызывает событие `select` для каждого элемента набора.

`$(sel).select(функция)` Назначает функцию к событию `select` для каждого элемента набора.

`$(sel).submit()` Вызывает событие `submit` для каждого элемента набора.

`$(sel).submit(функция)` Назначает функцию к событию `submit` для каждого элемента набора.

`$(sel).unload(функция)` Назначает функцию к событию `unload` для каждого элемента набора.

Эффекты jQuery

Базовые:

`$(sel).show()` Отображает каждый из совпавших элементов набора, если они были скрыты (не имеет значения как: либо через функцию `hide()`, либо посредством `display:none` в таблице стилей).

`$(sel).show(скорость, вызов)` Отображает все совпавшие элементы набора с использованием анимационных эффектов, после чего опционально запускает указанную в аргументе вызов функцию.

`$(sel).hide()` Скрывает каждый из совпавших элементов набора, если они были видимыми.

`$(sel).hide(скорость, вызов)` Скрывает все совпавшие элементы набора с использованием анимационных эффектов, после чего опционально запускает указанную в аргументе вызов функцию.

`$(sel).toggle()` Переключает режим отображения каждого из элементов набора.

`$(sel).toggle(переключатель)` Переключает режим отображения каждого из элементов набора в зависимости от значение переключателя (`true` — отображает элементы, `false` — скрывает элементы).

`$(sel).toggle(скорость, вызов)` Переключает режим отображения всех совпавших элементов набора с использованием анимационных эффектов, после чего опционально запускает указанную в аргументе вызов функцию.

Эффект «скольжения»:

`$(sel).slideDown(скорость, вызов)` Раскрывает все элементы набора, используя эффект изменения высоты элементов. Также, по завершению операции возможен запуск функции, переданной в аргумент “вызов”.

`$(sel).slideUp(скорость, вызов)` Скрывает все элементы набора, используя эффект изменения высоты элементов. Также, по завершению операции возможен запуск функции, переданной в аргумент “вызов”.

`$(sel).slideToggle(скорость, вызов)` Переключает видимость всех совпавших элементов набора, используя эффект изменения высоты элементов. Также, по завершению операции возможен запуск функции, переданной в аргумент “вызов”.

Эффекты появления/исчезания:

`$(sel).fadeIn(скорость, вызов)` Делает видимыми все элементы набора, используя изменение прозрачности элементов. Также, по завершению операции возможен запуск функции, переданной в аргумент “вызов”.

`$(sel).fadeOut(скорость, вызов)` Делает невидимыми все элементы набора, затем устанавливает CSS свойство `display` в “none”. Также, по завершению операции возможен запуск функции, переданной в аргумент “вызов”.

`$(sel).fadeTo(скорость, видимость, вызов)` Делает менее видимыми все элементы набора, изменяя прозрачность элементов до величины, указанной в аргументе “видимость”.

Пользовательская анимация:

`$(sel).animate(параметры, duration, easing, вызов)` Данная функция предназначена для создания пользовательской анимации.

`$(sel).animate(параметры, опции)` Данная функция предназначена для создания пользовательской анимации.

`$(sel).stop(clearQueue, gotoEnd)` Останавливает все запущенные анимационные эффекты для всех указанных элементов.

jQuery AJAX

`$.ajax(...)`

Это самый основной метод, а все последующие методы лишь обертки для метода `jQuery.ajax`. У данного метода лишь один входной параметр – объект включающий в себя все настройки (выделены параметры которые стоит запомнить):

- `async` – асинхронность запроса, по умолчанию `true`
- `cache` – вкл/выкл кэширование данных браузером, по умолчанию `true`
- `contentType` – по умолчанию `"application/x-www-form-urlencoded"`
- **`data`** – передаваемые данные – строка или объект
- `dataFilter` – фильтр для входных данных
- **`dataType`** – тип данных возвращаемых в callback функцию (`xml`, `html`, `script`, `json`, `text`, `_default`)
- `global` – триггер – отвечает за использование глобальных AJAX Event'ов, по умолчанию `true`
- `ifModified` – триггер – проверяет были ли изменения в ответе сервера, дабы не слать еще запрос, по умолчанию `false`
- `jsonp` – переустановить имя callback функции для работы с JSONP (по умолчанию генерируется на лету)
- `processData` – по умолчанию отправляемые данные заворачиваются в объект, и отправляются как `"application/x-www-form-urlencoded"`, если надо иначе – отключаем
- `scriptCharset` – кодировка – актуально для JSONP и подгрузки JavaScript'ов
- `timeout` – время таймаут в миллисекундах
- **`type`** – GET либо POST
- **`url`** – url запрашиваемой страницы

Полезные ссылки

Задание. Добавление скриптов на страницу

- Изучить структуру class_site/index.html;
- Подключить jQuery;
- Не изменяя структуры файлов HTML и CSS исходного документа, используя jQuery, выполнить:
 - При нажатии на Homepage:
 - Изменить цвет ссылок блока Categories на зеленый;
 - Изменить цвет ссылок блока Blogroll на желтый;
 - Уменьшить на два пункта размер надписей Categories, Blogroll, Archives, Aliquam tempus;
 - Добавить черные границы, синий фон и желтый цвет текста элементу input рядом с кнопкой GO;
 - При наведении мыши на ссылку COMMENTS блок с постом должен менять цвет фона на светло-зеленый;
 - Дата поста исчезает при нажатии на нее. Вместо даты появляется слово “Скрыто”;
 - Удалить все ссылки в каждом втором посте при нажатии на кнопку Blog;
 - При нажатии пункта меню ABOUT скрытые даты постов появляются, а не скрытые – меняют цвет текста на фиолетовый;
 - В посте с идентификатором «special» сделать первую букву абзаца (обрамлена в) в два раза больше, полужирной и окрасить красным, важную информацию (в тексте обрамлена тегом span) окрасить красны;
 - Для таблицы в самом нижнем посте при нажатии ссылки «Do more» выполнить:
 - Залить четные строки светло-желтым цветом;
 - Добавить внутренние отступы ячеек;
 - Уменьшить размер шрифта ячеек до 10px;
 - Первую строку таблицы залить желтым цветом, увеличить размер шрифта и сделать текст полужирным;
 - Окрасить внутренние границы каждой строки серым цветом, при этом внутренние границы столбцов должны быть проведены пунктирной линией;
 - Заменить текст последней ячейки каждой строки на ее порядковый номер (сверху вниз);
 - Текст первой ячейки каждой строки сделать зеленым, исключая первую строку;
 - При нажатии на кнопку GO в текстовом поле должен быть отображен определитель матрицы, хранимой в таблице Small table;
 - При нажатии на кнопку Contact изменить цвет копирайта внизу на ярко-желтый и увеличить размер шрифта в два раза;
 - При нажатии на кнопку Photos содержимое (entry) нечетных постов нужно заменить какой-либо картинкой. При нажатии на картинку – скрыть её.

Практика. Пример использования jQuery

```
<html>
<head>
  <style type="text/css">
    * {margin:0;padding:0;color:black;font-family:Arial;}
    #layout {
      width: 1000px;
      margin: 0 auto;
    }
    .bordered {border: 1px solid #000000;}
    .redbordered {border-color: #FF0000;}
    .blue {color: darkblue;}
    .rdotted {border-style:solid; border-right-style: dotted;}
    .ldotted {border-style:solid; border-left-style: dotted;}
    .lrdotted {border-style:solid; border-left-style: dotted; border-right-style: dotted;}
    li {color: #ff0000;background: #CCC;border-bottom:1px solid red;}
  </style>
  <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
  <script type="text/javascript">
    //внешняя функция
    function onBoxMove(element) {
      $(element).toggleClass("redbordered");
    }
    //код, исполняемый при полной загрузке документа
    $(document).ready(function () {
      //все элементы с классом button должны выглядеть как кнопки
      $(".button").addClass("dotted").css("cursor","pointer").css("background","#DDDDDD").width(100);

      //все блоки с классом box должны иметь внешние отступы, а при наведении мыши еще и красные
      границы
      $("div.box")
        .addClass("bordered")
        .css("margin",20)
        .mouseover(function () {onBoxMove(this);})
        .mouseout(function () {onBoxMove(this);});

      //все элементы с классом button при щелчке будут выполнять:
      $(".button").click(function () {
        //добавлять границы к шапке
        $("#header").addClass("bordered");
        //Удлинять гиперссылки с атрибутом width="100"
        $("li[width=100]").width(200);
        //Все таблицы примут ширину 300 и толстые внешние границы
        $(".table").width(300).addClass("bordered").css("border-width","3px");
        //Внутренняя граница между ячейками должна быть пунктирной, а остальные - сплошными
        $(".table")
          .find("td") //находим все ячейки таблицы
          .addClass("bordered lrdotted") // применяем два класса
          .end() //возвращаемся к таблице
          .find("td:first-child") //находим первую ячейку каждой строки
          .addClass("rdotted") //применяем класс
          .end() //возвращаемся к таблице
          .find("td:last-child") //находим последний элемент внутри каждого дочернего
          .addClass("ldotted"); //применяем класс
        //Все ячейки всех таблиц будут отцентрированы
        $(".table > td").css("text-align","center");
        //Первая ячейка каждой четной строки получит желтый фон
        $(".tr:odd > td:first-child").css("background","yellow");
        //Нечетные строки станут серыми
        $(".tr:nth-child(even)").css("background","#CCCCCC");
        //Текст абзацев в блоках должен быть синим, а важный в них - красным
        $(".div.box > p").addClass("blue").find("span").css("color","red");
      });

      //все картинки при клике на них будут исчезать
      $(".img").click(function() {
        $(this).hide();
      });
      //при нажатии на любую ссылку с цифрой 5, все прочие ссылки сокращаются до 100px
      $(".a:contains(5)").click(function () {
        $("li > a:not(this)").width(100);
      });

      //при нажатии на ссылку с цифрой 4, все скрытые картинки появляются назад
      $(".a:contains(4)").click(function () {
        $(".img:hidden").show();
      });
    });
  </script>
</head>
<body>
```

```

//при нажатии на ссылку с цифрой 2 элементы li становятся строчными, при повторном - блочными
$("a:contains(2)").click(function () {
  if ($("li").css("display")=="block") {
    $("li").css("display","inline");
  } else {
    $("li").css("display","block");
  }
});
//при нажатии на любую ссылку у нее меняется цвет на синий и, при повторном, обратно на черный
$("a").click(function () {
  $(this).toggleClass("blue");
});
});
</script>
</head>
<body>
  <div id="layout">
    <div id="header">
      <ul>
        <li width="100"><a href="#">Link 1</a></li>
        <li width="100"><a href="#">Link 2</a></li>
        <li width="100"><a href="#">Link 3</a></li>
        <li width="100"><a href="#">Link 4</a></li>
        <li width="150"><a href="#">Link 5</a></li>
      </ul>
    </div>
    <div class="box">
      <p>Some <span>important</span> text.</p>
      <p>Some <span>important</span> text.</p>
      <p>Some <span>important</span> text.</p>
    </div>
    <div class="box">
      <p align="left">Some <span>important</span> text.</p>
      <p>Some images .</p>
      <p>Some <span>important</span> text.</p>
    </div>
    <div class="box">
      <p>Some <span>most important</span> text.</p>
      <p align="left">Some button</p>
      <div class="button">Do the job!</div>
    </div>
    <div class="box">
      <p>Some <span>most important</span> text.</p>
      <p>And tables!</p>
      <table>
        <tr><td>1</td><td>John</td><td>Peter</td><td>Jane</td></tr>
        <tr><td>2</td><td>Jane</td><td>John</td><td>Peter</td></tr>
        <tr><td>3</td><td>Peter</td><td>Jane</td><td>John</td></tr>
        <tr><td>4</td><td>Philip</td><td>Peter</td><td>John</td></tr>
        <tr><td>5</td><td>Oliver</td><td>John</td><td>Philip</td></tr>
      </table>
    </div>
  </div>
</body>
</html>

```

Исходный код в файле test.html