БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ

Кафедра «Автоматизированные системы управления»

ТЕХНОЛОГИЯ ИНТЕРНЕТ-ПРОГРАММИРОВАНИЯ

Методические указания к лабораторной работе № 11 для студентов специальности
09.03.01 Автоматизированные системы обработки информации и управления

Могилев 2020

Лабораторная работа №11
JQuery Основные методы

Цель работы: Изучить Основные методы JQuery

Порядок выполнения работы.
Изучить теоретические сведения.
Выполнить задание к лабораторной работе в соответствии с вариантом.
Оформить отчет.

Требования к отчету.
Цель работы.
Постановка задачи.
Текст программы.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

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 по работе с Ajax.

JavaScript - объектно-ориентированный скриптовый язык программирования. 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. На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами. Активно используется в Web разработке.

Основы выборки элементов с помощью JQuery

«Выборка элементов» — для тех кто хорошо знает CSS то это будет легко так как выборка в jQuery ничем не отличается от выборки в CSS.

С самого начала нужно подключить и открыть функцию jQuery смотрим:

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js"></script>

В CSS для выбора, например, всех ссылок <a> внутри абзаца <p> мы писали:

p a

В jQuery для этого используется следующее выражение:

$(selector)

или

jQuery(selector)

Здесь следует отметить, что $() является алиасом к функции jQuery(). Она возвращает специальный объект JavaScript, содержащий массив элементов DOM, соответствующий селектору.

$(“p a”)

Основные селекторы

Основные селекторы вам знакомы, если вы когда-нибудь создавали хотя бы один шаблон CSS. скорее всего, вы будете использовать данные селекторы на протяжении 90% всего времени работы с jQuery:

Селектор	Описание	Пример
Все	Выбирает все элементы на странице, включая head, body и так далее.	$("*")
Элемент	Выбирает все элементы с заданным тегом.	$("p") $("div")
Класс	Выбирает все элементы с заданным именем класса.	$(".myClass") $("p.myClass")
ID	Выбирает один элемент с заданным атрибутом id.	$("#myID") $("p.#myID")

Вы также можете комбинировать несколько селекторов в один с помощью запятой. jQuery выберет все элементы, которые соответствуют любому из селекторов. Например:

```
// Выбирает все div с классом .myClass, а также все параграфы
var selectedElements = $("div.myClass, p");
```

Выбор элементов по атрибуту

Основные селекторы отлично подходят в случаях, если нужно выбрать все параграфы на странице или элемент, который нужно выбрать имеет класс CSS или ID.

Однако, иногда нужно выбрать определенный элемент, у которого нет класса или ID, и нет возможности просто добавить класс или ID к данному элементу в разметке. Такая ситуация может сложиться при использовании CMS с фиксированным шаблоном HTML или при работе с контентом, созданным пользователем.

В данной ситуации возможно сузить область выбора с помощью атрибута HTML элемента, который надо выбрать. Например, можно выбрать:

- Изображение по атрибуту src
- Ссылку по атрибуту href

- Все поля input формы, которые имеют атрибут type="checkbox"

...и так далее.

jQuery имеет много селекторов, которые можно использовать для выбора элементов по атрибуту:

Селектор	Описание	Пример
Атрибут	Выбирает элемент(ы), которые содержат определенный атрибут вне зависимости от значения атрибута.	$("div[attributeName]")
Атрибут равен	Выбирает элемент(ы), которые содержат заданный атрибут с заданным значением.	$("div[attributeName='value']")
Атрибут не равен	Выбирает элемент(ы), которые не содержат заданного атрибута или сожержат заданный атрибут, но его значение не соответствует заданному.	$("div[attributeName!='value']")
Атрибут начинается с	Выбирает элемент(ы), которые содержат заданный атрибут, у которого значение начинается с заданной строки.	$("div[attributeName^='value']")
Атрибут заканчивается	Выбирает элемент(ы), которые содержат атрибут, у которого значение заканчивается заданной строкой.	$("div[attributeName$='value']")
Атрибут содержит	Выбирает элемент(ы), который содержит атрибут, у которого значение содержит заданную строку.	$("div[attributeName*='value']")
Атрибут содержит слово	Выбирает элемент(ы), который содержит атрибут, у которого значение содержит заданное слово. "Слово" - это последовательность символов без пробелов.	$("div[attributeName~='value']")
Атрибут содержит префикс	Выбирает элемент(ы), который содержит атрибут, у которого значение либо равно заданной строке, либо начинается с заданной строки с последующим	$("div[attributeName|='value']")

	дефисом.	

Вы можете комбинировать селекторы атрибутов для сужения области выбора. Например:

```
// Выбираем только те изображения, которые имеют ширину 200 px и высоту 300 px
var selectedImages = $("img[width=300][height=200]");
```

Селектор "Атрибут содержит префикс" выглядит неэффективно, но он сделан для обработки атрибутов языка, таких как lang="en" и lang="en-US".

Выбор элементов по содержанию

Если нет возможности сузить область выбора по основным селекторам и по атрибутам, то можно "покопаться" в содержании элемента для определения, подходит он для выбора или нет. jQuery имеет 4 селектора для данного назначения:

Селектор	Описание	Пример
:contains()	Выбирает элемент(ы), которые содержат заданный текст. Текст может быть в самом элементе, так и в любом элементе внутри в выбранного элемента. Примечание: :contains() чувствительно к регистру -"Hello" не соответствует "hello".	// Выбираем все div, которые содержат "myText": $("div:contains('myText')")
:has()	Выбирает элемент(ы), которые содержат 1 или более элементов, соответствующих заданной строке. Селектор просматривает на соответствие все элементы внутри заданного элемента.	// Выбираем все div, которые содержат параграфы: $("div:has(p)")
:parent	Выбирает элемент(ы), которые содержат другие элементы или текстовые узлы.	// Выбираем все div, которые содержат что-нибудь: $("div:parent")
:empty	Выбирает элемент(ы), которые не содержат других элементов или текстовых узлов.	// Выбираем все div, которые не содержат ничего: $("div:empty")

Следующий пример показывает, как использовать :contains(), наряду с селектором класса для выбора параграфов, которые имеют определенный класс, дополнительным условием служит условие содержания определенной строки (или строк):

```
// Выбираем все параграфы,которые имеют класс
"intro"
// и также содержат текст "MegaWidget"
var selectedElements =
$("p.intro:contains('MegaWidget')");

// Выбираем все параграфы, которые имеют класс
"intro"
// и содержат и "MegaWidget" и "WonderWidget"
var selectedElements =
$("p.intro:contains('MegaWidget'):contains('WonderWidget')");
```

Выбор элементов по иерархии

Другой способ выбрать элементы в jQuery - это рассмотреть, как они соотносятся друг к другу на странице. Вероятно, вы знаете много таких селекторов по работе с CSS:

Селектор	Описание	Пример
Ребенок	Выбирает элемент(ы), которые являются ребенком (прямым потомком) заданного предка(ов).	// Выбираем все элементы списка с классом "highlight", которые являются ребёнком для списка с ID "nav": $("ul#nav > li.highlight")
Потомок	Выбирает элемент(ы), которые являются потомками (детьми, внуками и так далее) заданного предка(ов). Это более общая форма селектора Ребенок.	// Выбираем все ссылки внутри списка с ID "nav": $("ul#nav a")
Следующий соседний элемент	Выбирает элемент(ы), который следует непосредственно за заданным элементом, где оба элемента имеют одного и того же родителя.	// Выбираем все параграфы, которые следуют непосредственно за заголовком H1: $("h1 + p")
Следующий сестринский элемент	Выбирает элемент(ы), который следует за другим элементом, где оба элемента являются потомками одного родителя. Это более обобщенная версия селектора следующий соседний элемент.	// Выбираем все ячейки таблицы после ячейки, которая содержит слово "Hello": $("td:contains('hello') ~ td")
Первый ребенок	Выбирает элемент(ы), который является первым ребенком его	// Выбираем первые пункты во всех списках на странице:

	родителя.	$("li:first-child")
Последний ребенок	Выбирает элемент(ы), который является последним ребенком его родителя.	// Выбираем последние пункты во всех списках на странице: $("li:last-child")
N-й ребенок	Выбирает элемент(ы), который является n-м ребенком его родителя (смотри примечание ниже).	// Выбираем третьи пункты во всех списках на странице: $("li:nth-child(3)")
Только ребенок	Выбирает элемент(ы), которые являются ребенком родителей, у которых есть только дети.	// Выбираем только элементы в списках с одним пунктом: $("li:only-child")

Кроме задания определенного номера ребенка с помощью :nth-child(), можно указывать even (для выбора всех детей с четными номерами), odd (для выбора всех детей с нечетными номерами), или выражение (например,"li:nth-child(3n+2)" выбирает каждый третий элемент в списке, а отсчет начинается со второго элемента).

В следующем примере выбирается первая ячейка всех нечетных строк в таблице, которая имеет id "myTable":

```
var selectedElements = $("table#myTable tr:nth-child(odd) > td:first-child");
```

Выбор полей формы

Элементы формы имеют некоторые специфические свойства, которые вынуждают использовать некоторые трюки для их выбора. Например, чекбоксы, наряду со многими другими типами полей, являются элементами input. Таким образом,, чтобы выбрать чекбоксы в форме нужно использовать $("input[type='checkbox']").

Аналогично, идентификация выбранных элементов в списке select или отмеченных чекбоксов в форме может быть утомительным занятием./

К счастью, jQuery обеспечивает несколько специфических для форм селекторов, которые облегчают жизнь:

Селектор	Описание	Пример
:button	Выбирает все кнопки формы.	$("input:button")
:checkbox	Выбирает все чекбоксы.	$("input:checkbox")
:file	Выбирает все поля загрузки фала.	$("input:file")
:image	Выбирает все поля ввода изображения.	$("input:image")
:password	Выбирает все поля пароля.	$("input:password")

:radio	Выбирает все радио кнопки.	$("input:radio")
:reset	Выбирает все кнопки перезагрузки формы.	$("input:reset")
:submit	Выбирает все кнопки отправки формы.	$("input:submit")
:text	Выбирает все поля ввода текста.	$("input:text")
:input	Выбирает все поля формы, включая элементы input, textarea, и select.	$(":input")
:checked	Выбирает все отмеченные чекбоксы радио кнопки.	$("input:checked")
:selected	Выбирает все элементы option.	$("option:selected")
:disabled	Выбирает все недоступные поля формы.	$("input:disabled")
:enabled	Выбирает все Доступные поля формы.	$("input:enabled")

Кроме того, что написано, можно использовать, например, $("input:checkbox") для выбора всех чекбоксов на странице. Однако, $("input:checkbox") действует быстрее, так как jQuery может использовать функцию JavaScriptgetElementsByTagName() для быстрого ограничения объемов поиска только элементами input. Это действительно для jQuery вообще - для ускорения кода всегда ограничивайте объем для работы селекторов там, где возможно. Например, $("*") действует очень медленно, так как jQuery должен найти каждый отдельный элемент на странице.

Следующий пример ищет форму, которая имеет атрибут action со значением "mailform.php", затем выбирает все отмеченные радио кнопки в форме:

```
var selectedElements = $("form[action='mailform.php'] input:radio:checked");
```

Выбор элементов по их положению

Иногда нужно выбрать элемент, про который известно, что он расположен в наборе ранее выбранных элементов. Например, нужно выделить первый параграф, который имеет класс "myClass". Для усложнения примера допустим, что нужно выбрать все пункты в списке, который имеет класс "myClass", а затем выбрать 5-й пункт из получившегося набора.

jQuery имеет семь селекторов, которые можно использовать для ограничения области поиска на основе позиции элемента:

Селектор	Описание	Примеры
:first	Выбирает первый элемент в наборе отобранных элементов.	// Выбираем первый параграф, который имеет класс "myClass": $("p.myClass:first")
:last	Выбирает последний элемент в наборе отобранных элементов.	// Выбираем последний параграф, который имеет

		класс "myClass": $("p.myClass:last")
:eq()	Выбирает единственный элемент в наборе отобранных элементов. Выбор элемента осуществляется по последовательному номеру индекса (0 = первый элемент, 1 = второй и так далее).	// Выбираем з-й параграф, который имеет класс "myClass": $("p.myClass:eq(2)")
:lt()	Выбирает элементы в наборе отобранных элементов, которые расположены перед элементом с заданным индексом. Например, если задан индекс 2 (то есть 3-й элемент), то будут выбраны первые 2 элемента (с индексами 0 и 1).	// /Выбираем первые 2 параграфа, которые имеют класс "myClass": $("p.myClass:lt(2)")
:gt()	Выбираем элементы в наборе отобранных элементов, которые расположены после элемента с заданным индексом. Например, если задан индекс 2 (то есть 3-й элемент), то будут выбраны все элементы после третьего.	// Выбираем все параграфы, которые имеют класс "myClass", за исключением первых трех: $("p.myClass:gt(2)")
:even	Выбирает все элементы с четными индексами в наборе отобранных элементов. Заметьте, что индексы начинаются с 0, таким образом в действительности отбираются 1-й, 3-й и так далее элементы.	// Выбираем 1й, 3й, 5й, и так далее параграфы, которые имеют класс "myClass": $("p.myClass:even")
:odd	Выбирает все элементы с нечетными индексами в наборе уже отобранных элементов. Заметьте, что индексы начинаются с 0, таким образом в действительности отбираются 2-й, 4-й и так далее элементы	// Выбираем 2й, 4й, 6й, и так далее параграфы, которые имеют класс"myClass": $("p.myClass:odd")

Отметим, что данные селекторы не работают также как :first-child, :last-child, и так далее. Например,li.myClass:first-child выбирает только пункт списка с классом "myClass" , который является первым пунктом в соответствующем списке. А li.myClass:first находит все пункты списка на странице, которые имеют класс "myClass", а затем выбирает первый пункт списка в наборе результата поиска.

В следующем примере выбираются все ячейки в первых двух строках таблицы, которая имеет ID "myTable":

```
var selectedElements = $("table#myTable tr:lt(2)
> td");
```

Другие полезные селекторы jQuery

Если ни один из перечисленных селекторов не помог вам выделить нужные элементы, попробуйте использовать следующие селкторы:

Селектор	Описание	Пример
:not()	Выбирает все элементы, которые не соответствуют заданному слектору.	// Выбираем все параграфы, которые не имеют класса "myClass": $("p:not(.myClass)")
:animated	Выбирает все элементы, которые в текущий момент анимируются jQuery (например, затухают).	// Выбираем все div, которые анимируются в текущий момент: $("div:animated")
:hidden	Выбирает все скрытые элементы. Элемент полагается "скрытым" если: его свойство display установлено в значение "none"; поля формы - type"hidden"; если ширина и высота установлены в 0; если один из элементов, которые содержат заданный элемент является скрытым. Однако, элемент несчитается "скрытым" если только его свойство visibility установлено в значение "hidden".	// Выбираем все скрытые праграфы, которые имеют класс "myClass": $("p.myClass:hidden")
:visible	Выбирает все видимые элементы. Это противоположный селектор для :hidden.	// Выбираем все видимые параграфы, которые имеют класс "myClass": $("p.myClass:visible")
:header	выбирает все элементы заголовков (h1, h2 и так далее).	// Выбираем все заголовки, которые имеют класс "myClass": $(":header.myClass")

В следующем примере выбираются все элементы на странице, кроме заголовков h1:

view source
print?

```
var selectedElements =
$(":header:not(h1)");

$(document).ready()
```

Это первая вещь, которую вы должны знать перед тем как начать изучать jQuery: если вы хотите, чтобы события выполнялись на ваших страницах, вы должны вызывать их из функции $(document).ready(). Каждая операция внутри него выполняется сразу после того, как страница закончит загружаться и перед тем как содержимое страницы будет показано.

```
$(document).ready(function() {
  //Здесь будет ваш код
 });
```

С помощью $(document).ready() , вы можете загрузить событие или вызвать событие или любую другую операцию которая выполнится перед загрузкой страницы. Все, что вы ставите внутрь скобок будет готово к выполнению в самый ранний момент - в момент как DOM будет записан браузером, который позволяет нескольким красивым эффектам и другим вещам выполниться сразу как пользователь увидит элементы страницы первый раз.

Пример 1

```
$(document).ready(function () {
   $("p").text("DOM загружена! ");
});
```

Результат выполнения кода выше:

DOM загружена!

Задания к лабораторной работе

Вариант 1

Отобрать в набор элементы “li”, которые находятся внутри блока с id=" my_links" , в которых содержатся элементы “a”.

Отобрать в набор все ссылки, которые находятся внутри блока с id="my links" и при этом значение атрибута href у них начинается со слова "documents".

Отобрать в набор только последние ячейки, которые находятся внутри таблицы с id=" moto_table " в четных строках

Вариант 2

Отобрать в набор элементы li, которые находятся внутри блока с id=" moto_models" , в которых содержится текст “ мотоцикл”.

4. Отобрать в набор только те картинки, которые находятся внутри блока с id="forfooter" и в качестве значения атрибута title имеют значение

"Производители".

Отобрать в набор только последние ячейки, которые находятся внутри таблицы с id=" moto_table " в четных строках