**Лабораторная работа 2\_3 Библиотека jQuery**

Целью JQuery является - упростить использование JavaScript на веб-сайте. JQuery является библиотекой JavaScript и значительно упрощает программирование JavaScript.

JQuery имеет плагины для любых задач.

JQuery работает одинаково во всех основных браузерах,

**Добавление библиотеки JQuery**

Существует несколько способов подключения JQuery на веб-страницу.

- Скачать библиотеку JQuery, источник - jQuery.com

- Включить JQuery из CDN, например, Google

Есть две версии JQuery доступны для скачивания:

- Производственная версия (Productionversion) - сжатая версия

- Разрабатываемая версия (Developmentversion) - это для тестирования и разработки (несжатый и читаемый код)

Обе версии могут быть загружены с jQuery.com.

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js"></script>

Библиотека JQuery представляет собой один JavaScript файл, ссылка указывается на него с помощью HTML тега <script> (тег <script> должен находиться в секции <HEAD>):

<head><script src="jquery-1.11.3.min.js"></script></head>

Поместить скачанный файл в тот же каталог, где находятся веб-страницы. Или создать отдельную папку и указать к ней путь.

Код jQuery состоит из последовательно идущих команд. Стандартный синтаксис jQuery команд:

**$(селектор).метод();**

Знак $ сообщает, что символы, идущие после него, являются jQuery кодом; селектор позволяет выбрать элемент на странице; метод задает действие, которое необходимо совершить над выбранным эле-ментом. Методы в jQuery разделяются на следующие группы:

 методы для манипулирования DOM;

 методы для оформления элементов;

 методы для создания AJAX запросов;

 методы для создания эффектов;

 методы для привязки обработчиков событий.

jQuery можно комбинировать с обычным JavaScript. Если строка начинается с $ – это jQuery, если $ в начале строки отсутствует – это строка JavaScript кода.

**Селекторы jQuery**

С помощью селекторов можно выбирать элементы на странице для применения к ним определенных действий.

Ниже располагается таблица с различными примерами использования селекторов для выбора желаемых элементов:

Таблица 10.

|  |  |
| --- | --- |
| Пример | Результат |
| $("p") | Будут выбраны все элементы p, которые находятся на странице. |
| $(".par") | Будут выбраны все элементы на странице с class="par". |
| $("#par") | Будет выбран первый элемент на странице с id="par". |
| $(this) | Позволяет выбрать текущий HTML элемент. |
| $("p.par") | Будут выбраны все элементы p на странице с class="par". |
| $("p#par") | Будут выбраны все элементы p на странице с id="par". |
| $(".par,.header,#heat") | Будут выбраны все элементы на странице со значениями атрибутов class="par", class="header" и id='heat'. |
| $("[src]") | Будут выбраны все элементы на странице, имеющие атрибут src. |
| $("[src='значение']") | Будут выбраны все элементы со значениям атрибута src="значение". |
| $("[src$='.gif']") | Будут выбраны все элементы со значениями атрибута src заканчивающимися на .gif. |
| $("div#wrap .par1") | Будут выбраны все элементы с class=par1, которые находятся внутри элементов div с id=wrap. |
| $(":input") | Будут выбраны все input элементы на странице. |
| $(":тип") | Будут выбраны все input элементы с <input type='тип' />. Например :button выберет все <input type='button' /> элементы, :text выберет все <input type='text' /> элементы и т.д. |

**2 Методы обработчиков событий jQuery**

**2.1 События (Events)**

Все действия различных пользователей на которые веб-страница может реагировать называются событиями. Примеры:

- перемещение мыши над элементами

- выбор радиокнопок (переключателей)

- нажатие мыши

Распространенные события DOM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MouseEvents** | | **KeyboardEvents** | **FormEvents** | **Document/WindowEvents** |
| click | | keypress | submit | load |
| dblclick | | keydown | change | resize |
| mouseenter | | keyup | focus | scroll |
| mouseleave | blur | | | unload |

**2.2 Синтаксис методов обработчиков событий JQuery.**

Большинство событий DOM имеют эквивалентный метод JQuery.

*2.2 Синтаксис методов обработчиков событий JQuery.*

Назначить событие по клику для всех элементов <p> на странице возможно следующим образом: **$("p").click();**

Следующим шагом является определение того, что должно произойти, после запуска события. Необходимо передать функцию событию:

**$("p").click(function(){ // что-то делаем});**

2.3Часто используемые методы обработчиков событий JQuery.

|  |  |  |
| --- | --- | --- |
| Метод | Пример | Описание |
| $(document).ready() | $("p").click(function(){ $(this).hide(); }) | обеспечивает выполнение функции, после полной загрузки документа. |
| dblclick() | $("p").dblclick(function(){ $(this).hide(); }); |  |
| mouseenter() | $("#p1").mouseenter(function(){ alert("You entered p1!"); }); | Функция выполняется, когда указатель мыши наведен на HTML объект |
| mouseleave() | $("#p1").mouseleave(function(){ alert("Bye! You now leave p1!"); }); | Функция выполняется, когда указатель мыши покидает HTML объект |
| mousedown() | $("#p1").mousedown(function(){ alert("Mouse down over p1!"); }); | присоединяет функцию обработчика события к элементу HTML. Функция выполняется, когда нажата левая, средняя или правая кнопка мыши, а мышь над элементом HTML |
| mouseup() | $("#p1").mouseup(function(){ alert("Mouse up over p1!"); }); | Когда мышь освободилась |
| hover() | $("#p1").hover(function(){ alert("You entered p1!"); }, function(){ alert("Bye! You now leave p1!"); }) | Имеет две функции, которые выполняются как комбинации mouseenter() andmouseleave() |
| focus() | $("input").focus(function(){ $(this).css("background-color", "#cccccc"); }); | Добавляет обработчик события к полям формы, выполняется, когда поле формы получает фокус |
| blur() | $("input").blur(function(){ $(this).css("background-color", "#ffffff"); }); | выполняется, когда поле формы теряет фокус |

**3 ЭФФЕКТЫ JQUERY**

**3.1. Показ и скрытие**

Синтаксис:

$(selector).hide(speed,callback);

$(selector).show(speed,callback);

Параметры (необязательные)

speed – определяет скорость скрытия/ показа, и может принимать следующие значения: "slow", "fast", или milliseconds.

Необязательный параметр callback - функция, которая будет выполнена после завершения методов hide() или show().

Пример 1.

|  |
| --- |
| <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("#hide").click(function(){  $("p").hide();  });  $("#show").click(function(){  $("p").show();  });  });  </script>  </head>  <body>  <p>If you click on the "Hide" button, I will disappear.</p>  <button id="hide">Hide</button>  <button id="show">Show</button>  </body>  </html>  Пример 2. **Используем параметр скорости**  <script>  $(document).ready(function(){  $("button").click(function(){  $("p").hide(1000);  });  });  </script>  </head>  <body>  <button>Hide</button>  <p>This is a paragraph with little content.</p>  <p>This is another small paragraph.</p>  </body>  </html>  3**.2. Переключение**  $(selector).toggle(speed,callback);  Переключение между показом и скрытием.  Пример:  $("button").click(function(){ $("p").toggle(); });  **3.3. Видимость элементов**  Методы:  -fadeIn()  - fadeOut()  - fadeToggle()  - fadeTo() |

|  |  |  |
| --- | --- | --- |
| fadeIn() | $(*selector*).fadeIn(*speed,callback*); | используется для плавного отображения скрытого элемента |
| **fadeOut().** | $(*selector*).fadeOut(*speed,callback*); | используется для удаления из видимости элемента |
| **fadeToggle().** | $(*selector*).fadeToggle(*speed,callback*); | Переключение между вид/невид |
| fadeTo() | $(selector).fadeTo(speed,opacity,callback); | позволяет изменить цвет задавая значение непрозрачности (valuebetween 0 and 1). |

|  |  |  |
| --- | --- | --- |
| Пример:  $("button").click(function(){ $("#div1").**fadeIn();**  $("#div2").fadeIn("slow"); $("#div3").fadeIn(3000); }); | $("button").click(function(){ $("#div1").fadeToggle(); $("#div2").fadeToggle("slow"); $("#div3").fadeToggle(3000); }); | $("button").click(function(){ $("#div1").fadeTo("slow", 0.15); $("#div2").fadeTo("slow", 0.4); $("#div3").fadeTo("slow", 0.7); }); |

**Пример: Скрытие абзаца.**

$(document).ready(function(){

$("#but1").click(function(){$("#par1").fadeOut(3000)});

$("#but2").click(function(){$("#par1").fadeIn(3000)});

$("#but3").click(function(){$("#par1").fadeTo(3000,0.3)});

$("#but4").click(function(){$("#par1").fadeTo(3000,0.8)});

$("#but5").click(function(){$("#par1").fadeOut(3000,function(){ alert("Абзац был полностью скрыт.");});

});

});

**3.4. Скольжение**

slideDown() - Метод выполняет плавный разворот элемента до исходного размера вниз.

slideUp() - Метод выполняет плавное изменяет высоту элемента до 0.

slideToggle() - переключает методы slideDown() и slideUp().

**Синтаксис**

$(selector).slideUp(speed,callback);

$(selector).slideDown(speed,callback);

$(selector).slideToggle(speed,callback);

Пример 3 (толькоскрипт)

|  |
| --- |
| <script>  $(document).ready(function(){  $("#flip").click(function(){  $("#panel").slideUp("slow");  });  });  </script>  <style>  #panel, #flip {  padding: 5px;  text-align: center;  background-color: #e5eecc;  border: solid 1px #c3c3c3;  }  #panel {  padding: 50px;  }  </style>  </head>  <body>  <div id="flip">Click to slide up panel</div>  <div id="panel">Hello world!</div>  </body>  </html> |

пример 4 меню

С помощью метода **slideToggle** можно создавать на страницах удобные выпадающие меню.

$(document).ready(function(){

$("#menu").click(function(){$("#list").slideToggle(2000)});

$("#menu").toggle(function(){

$("#img").attr("src","menudown.gif")},function(){

$("#img").attr("src","menuup.gif")

});

$("#menu").mouseover(function(){$("#menu").css("background-color","#01939a")});

$("#menu").mouseout(function(){$("#menu").css("background-color","#006064")});

});

**Задание для лабораторной работы**

1. **Найди кота.**
   1. Создать страницу (фон, текстовые блоки- 2 абзаца, графический блок). Тема – что-то про котиков. Обязательно должно быть :слово КОТ (несколько раз повторяется в разных абзацах), картинка с изображением котов.
   2. Используя методы JQuery реализовать следующее: после открытия страницы появляется текст. При наведении мышью на слово КОТ, слово исчезает и появляется изображение. При нажатии на картинку, картинка «уплывает», а слова в тексте появляются.

2. Используя *пример 3* реализовать скрытие/отображение панелей ( событие - клик мышью на панели)

вариант а) сначала отображается панель 1, при клике на панель 1 - появляется панель 2, итак далее. Все панели должны иметь различные цвета.

1

2

3

4

вариант б)

Горизонтальное меню на странице

История Химия Математика

Алгебра

Арифметика

Органическая

История РБ

Новейшая

Древний мир

Неорганическая

Тригонометрия

С помощью метода **slideToggle** можно создавать на страницах удобные выпадающие меню. Например,

$(document).ready(function(){

$("#menu").click(function(){$("#list").slideToggle(2000)});

$("#menu").toggle(function(){

$("#img").attr("src","menudown.gif")},function(){

$("#img").attr("src","menuup.gif")

});

$("#menu").mouseover(function(){$("#menu").css("background-color","#01939a")});

$("#menu").mouseout(function(){$("#menu").css("background-color","#006064")});

});