

Школа:		
Дата:	ФИО учителя:	
Класс:	Участвовали:	Не участвовали:
Тема урока: Библиотеки jQuery, jQuery UI, CSS		
Цели обучения, которые достигаются на данном уроке	Научить обучающихся как создавать анимации в среде веб программирование с использованием Java script	
Цели урока	<p>Все учащиеся:</p> <ul style="list-style-type: none"> Знать библиотеку jQuery, jQuery UI, CSS <p>Большинство учащихся:</p> <ul style="list-style-type: none"> Распознать назначения библиотеки jQuery, jQuery UI, CSS <p>Некоторые учащиеся:</p> <ul style="list-style-type: none"> создавать веб страницу с использованием библиотеки jQuery, jQuery UI, CSS 	
Воспитание ценностей	Воспитание уважения к мнению друг друга, ответственность, коммуникативные способности, критическое мышление	
Межпредметные связи	Информатика, математика	
Запланированные этапы урока	Запланированная деятельность на уроке	Ресурсы
Начало урока _5_ мин	<p>Приветствую.</p> <p>Деление на подгруппы.</p> <p>Стратегия «Пазлы»</p> <p>Дескрипторы</p> <p>Последовательно разложите картинки.</p> <p>Подпишите картинки.</p> <p>В ходе обсуждения выполненного задания приходим к постановки цели.</p> <p>Повторение предыдущей темы:</p> <p>Метод «Горячий микрофон» Один ученик задает вопрос и передает микрофон и следующий ученик отвечает на вопрос и задает свой вопрос следующему ученику и т.д.</p>	<p>Карточки с картинками</p> <p>Микрофон</p>
Середина урока _20_ мин	<p>Для начала вспомним, для чего нам нужны плагины? Мой ответ — создание повторно используемого кода, и да — с удобным интерфейсом. Давайте напишем такой код, вот простая задача: "По клику на параграф, текст должен измениться на красный".</p> <p>Дабы не забывать истоков — начнем с реализации на нативном JavaScript'e:</p> <pre>var loader = function () { // находим все параграфы var para = document.getElementsByTagName('P'); // перебираем все, и вешаем обработчик for (var i=0,size=para.length;i<size;i++) { // обработчик para[i].onclick = function() { this.style.color = "#FF0000"; } }</pre>	<p>Карточки с новой темой</p> <p>Приложение № 1</p>

```
}  
}  
// естественно, весь код должен работать после  
загрузки всей страницы  
document.addEventListener("DOMContentLoaded",  
loader, false);
```

Данный код не является кроссбраузерным, и написан с целью лишний раз подчеркнуть удобство использования библиотеки ;)

Теперь можно этот код упростить, подключаем jQuery и получаем следующий вариант:

```
$(function(){  
$('p').click(function(){  
$(this).css('color', '#ff0000');  
})  
});
```

С поставленной задачей мы справились, но где тут повторное использование кода? Или если нам надо не в красный, а в зеленый перекрасить? Вот тут начинается самое интересное, чтобы написать простой плагин достаточно расширить объект \$.fn:

```
$.fn.mySimplePlugin = function() {  
$(this).click(function(){  
$(this).css('color', '#ff0000');  
})  
}
```

Если же писать более грамотно, то нам необходимо ограничить переменную \$ только нашим плагином, а так же возвращать "this", чтобы можно было использовать цепочки вызовов (т.н. "chaining"), делается это следующим образом:

```
(function($) {  
$.fn.mySimplePlugin = function(){  
// код плагина  
return this;  
};  
})(jQuery);
```

Внесу небольшое пояснение о происходящей тут "магии", код "(function(\$){...})(jQuery)" создает анонимную функцию, и тут же вызывает ее, передавая в качестве параметра объект jQuery, таким образом внутри анонимной функции мы можем использовать алиас \$ не боясь за конфликты с другими библиотеками — так как теперь \$ находится



лишь в области видимости нашей функции, и мы имеем полный контроль над ней. Если у вас возникло ощущение дежавю – то всё верно, я об этом уже рассказывал

Добавим опцию по выбору цвета и получим рабочий плагин:

```
<!DOCTYPE html>
<html dir="ltr" lang="en-US">
<head>
  <meta charset="UTF-8"/>
  <meta name="viewport" content="width=device-
width, initial-scale=1.0" />
  <title>Пример плагина с лобальными
настройками</title>
  <link rel="profile" href="http://gmpg.org/xfn/11"/>
  <link rel="shortcut icon"
href="http://anton.shevchuk.name/favicon.ico"/>
  <link rel="stylesheet" href="css/styles.css"/>
  <script type="text/javascript"
src="js/jquery.js"></script>
  <script type="text/javascript"
src="js/code.js"></script>
  <script type="text/javascript">
    (function($) {
      // значени по умолчанию
      var defaults = {
        color:'green'
      };
      // актуальные настройки, глобальные
      var options;
      $.fn.mySimplePlugin = function(params){
        // при многократном вызове функции
настройки будут сохраняться, и замещаться при
необходимости
        options = $.extend({}, defaults, options,
params);
        $(this).click(function(){
          $(this).css('color', options.color);
        });
        return this;
      };
    })(jQuery);
  </script>
</head>
<body>
  <div id="content" class="wrapper box">
    <menu>
      <a href="index.html" title="go prev"
class="button alignleft" rel="prev"><← Back </a>
```

```
<a href="#" title="reload" class="button alignleft"
onclick="window.location.reload();return false">Reload
⌘</a>
```

```
<hr/>
```

```
<pre><code>$(<span>p:eq(0)</span>).mySimplePlugin
()</code></pre>
```

```
<button type="button" class="code">Run
Code</button>
```

```
<pre><code>$(<span>p:eq(1)</span>).mySimplePlugin
({'color':'red'})</code></pre>
```

```
<button type="button" class="code">Run
Code</button>
```

```
<pre><code>$(<span>p:eq(2)</span>).mySimplePlugin
({'color':'blue'})</code></pre>
```

```
<button type="button" class="code">Run
Code</button>
```

```
</menu>
```

```
<header>
```

```
<h1>Пример плагина с глобальными
настройками</h1>
```

```
<h2>Запустите плагин с новыми настройками
и прокляцайте все параграфы</h2>
```

```
</header>
```

```
<article>
```

```
<h2>Article Title</h2>
```

```
<p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Phasellus rutrum,
```

```
lectus eu varius consectetur, libero velit hendrerit
augue, ut posuere enim neque
```

```
in libero. Donec eget sagittis nibh. Suspendisse
sed tincidunt urna. Cras quis
```

```
euismod neque. Maecenas auctor ultricies
posuere. Pellentesque luctus pulvinar dui
```

```
eget semper. Donec sodales odio eu sapien varius
luctus. Donec dictum feugiat diam
```

```
at malesuada. Sed nec massa in augue
condimentum faucibus quis ut diam. Quisque
```

```
nisl sem, semper nec vulputate vel, mattis sit amet
justo. Aliquam purus felis,
```

```
tempor at scelerisque quis, tincidunt in neque.
Etiam ut risus diam. Pellentesque
```

```
fermentum risus id elit feugiat cursus. Ut fringilla
dictum diam, sed iaculis
```

```
lorem pulvinar ut. Cras vel elit id velit commodo
viverra sit amet vel orci.</p>
```

```
</article>
```

```
<article>
```

```
<h2>Article Title</h2>
```

```
<p>Duis in vestibulum sem. Cras euismod
```

	<p>tincidunt dui, et scelerisque tellus condimentum vel. Maecenas et urna sit amet risus fermentum rhoncus nec porttitor ligula. Maecenas sit amet turpis enim, ut iaculis est. Duis feugiat, lacus id placerat porttitor, lorem augue gravida nisi, eu porta eros risus et lectus. Maecenas vestibulum nunc vel ipsum tincidunt sit amet blandit sapien bibendum. Proin vel vulputate nisl. Duis tempor imperdiet placerat. Pellentesque faucibus consequat magna, et bibendum nisl egestas non. Pellentesque sit amet mattis augue. Aenean at diam tincidunt purus sollicitudin gravida non in nisi. Fusce bibendum, magna in adipiscing mattis, sem risus fringilla mi, nec gravida lectus lectus at nibh. Suspendisse adipiscing elementum laoreet. Suspendisse sem erat, varius quis aliquet vitae, dapibus sed nibh. Nullam iaculis sem at mauris faucibus in vestibulum libero pretium. Aliquam eu turpis libero. Fusce et ultrices lectus.</p> </article> <article> </div> </body> </html> (function(\$) { // значение по умолчанию - ЗЕЛЁНЫЙ var defaults = { color:'green' }; // актуальные настройки, глобальные var options; \$.fn.mySimplePlugin = function(params){ // при многократном вызове настройки будут сохраняться // и замещаться при необходимости options = \$.extend({ }, defaults, options, params); \$(this).click(function(){ \$(this).css('color', options.color); }); return this; }; })(jQuery);</p>	
<p>Конец урока _15_ мин</p>	<p>Рефлексия: Метод «Рефлексивные карточки» Было интересно... Я понял, что... У меня получилось... Расскажу дома, что...</p>	<p>«Рефлексивные карточки»</p>

Дифференциация – каким образом Вы планируете оказать больше поддержки? Какие задачи Вы планируете поставить перед более способными учащимися?	Оценивание – как Вы планируете проверить уровень усвоения материала учащимися?	Охрана здоровья и соблюдение техники безопасности