

<b>School:</b>		
<b>Date:</b>	<b>Teacher's name:</b>	
<b>Grade:</b>	<b>Number present:</b>	<b>absent:</b>
<b>Topic of the lesson: JQuery, jQuery UI, CSS Libraries</b>		
<b>Learning objective(s) that this lesson is contributing to</b>	To teach students how to create animations in a web programming environment using a Java script	
<b>Lesson objectives</b>	<b>All learners will be able to:</b> <ul style="list-style-type: none"> <li>Know jQuery library, jQuery UI, CSS</li> </ul> <b>Most learners will be able to:</b> <ul style="list-style-type: none"> <li>Recognize jQuery, jQuery UI, CSS library assignments</li> </ul> <b>Some learners will be able to:</b> <ul style="list-style-type: none"> <li>create a web page using jQuery, jQuery UI, CSS library</li> </ul>	
<b>Value links</b>	Building respect for each other's opinions, responsibility, communication skills, critical thinking	
<b>Cross curricular links</b>	Computer science, mathematics	
<b>Time</b>	<b>Planned activities</b>	<b>Resources</b>
<b>Beginning 5 min</b>	<p>Greetings.</p> <p>Division into subgroups.</p> <p>Puzzles Strategy</p> <p>Descriptors</p> <p>Lay out the pictures in sequence.</p> <p>Sign the pictures.</p> <p>During the discussion of the completed assignment, we arrive at the goal setting.</p> <p>Repeating the previous topic:</p> <p>“Hot Microphone” Method: A student asks a question and passes the microphone, and the next student answers the question and asks his question to the next student, etc.</p>	<p>Cards with pictures</p> <p>Microphone</p>
<b>Middle 20 min</b>	<p>To begin with, remember what we need plugins for? My answer is to create reusable code, and yes, with a user-friendly interface. Let's write such a code, here is a simple task: "By clicking on a paragraph, the text should change to red."</p> <p>In order not to forget the origins, let's start with the implementation in native JavaScript:</p> <pre>var loader = function () { // find all paragraphs var para = document.getElementsByTagName ('P'); // iterate over everything and hang up the handler for (var i = 0, size = para.length; i &lt;size; i ++ ) { // handler para [i] .onclick = function () {</pre>	<p>Cards with the new lesson</p> <p>App No 1</p>

```
this.style.color = "# FF0000";
```

```
}
```

```
}
```

```
}
```

// naturally, all code should work after loading the whole page

```
document.addEventListener ("DOMContentLoaded", loader, false);
```

This code is not cross-browser, and was written with the aim of once again emphasizing the usability of the library;)

Now we can simplify this code, we include jQuery and we get the following option:

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

We coped with the task, but where is the reuse of the code? Or if we need to repaint in green, and in green? This is where the fun begins, to write a simple plug-in it is enough to expand the \$ .fn object:

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

If we write more competently, then we need to limit the \$ variable only to our plugin, and also return "this" so that we can use call chains (the so-called "chaining"), this is done as follows:

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

I'll make a little explanation about the "magic" happening here, the code (function (\$) {...}) (jQuery) "creates an anonymous function, and immediately calls it,



passing in a jQuery object as a parameter, so we can use inside an anonymous function \$ alias is not afraid of conflicts with other libraries - since now \$ is only in the scope of our function, and we have full control over it. If you have a feeling of deja vu - then everything is correct, I already talked about this

Add an option for choosing a color and get a working plugin:

```
<!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> An example of a plug-in with global settings </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'
      };
      // current settings, global
      var options;
      $.fn.mySimplePlugin = function(params){
// If you call multiple calls, the settings will be retained
// and replaced if necessary
        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> An example of a plug-in with global settings </h1>
  <h2> Start the plug-in with new settings and download all the paragraphs
</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 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>
```

	<pre> &lt;/article&gt; &lt;article&gt;   &lt;/div&gt; &lt;/body&gt; &lt;/html&gt; (function(\$) { // Default - GREEN var defaults = { color:'green' }; // current settings, global var options; \$.fn.mySimplePlugin = function(params){ // If you call multiple calls, the settings will be retained // and replaced if necessary options = \$.extend({ }, defaults, options, params); \$(this).click(function(){ \$(this).css('color', options.color); }); return this; }; })(jQuery); </pre>	
<p><b>End 15 min</b></p>	<p>Reflection:  Reflective Card Method  It was interesting...  I realized that ...  I managed...  I'll tell others at home that ...</p>	<p>Cards</p>
<p><b>Differentiation – how do you plan to give more support?  How do you plan to challenge the more able learners?</b></p>	<p><b>Assessment – how are you planning to check learners' learning?</b></p>	<p><b>Health and Safety</b></p>

