Практическая работа

Курс: Разработка интерфейса на JavaScript

Дисциплина: Основы JavaScript

Практическая работа №38: Создание своих плагинов в jQuery

Создание своего плагина

Сперва создаём новое свойство-функцию для объекта jQuery, где именем нового свойства будет имя нашего плагина:

```
jQuery.fn.myPlugin = function() {

  // Тут пишем функционал нашего плагина

};
```

Но постойте, где-же привычный нам значок доллара, который мы все хорошо знаем? Он всё ещё здесь, а чтобы он не конфликтовал с другими библиотеками, которые тоже могут использовать символ доллара, рекомендуется «обернуть» объект jQuery в непосредственно выполняемую функцию-выражение, которое связывает объект jQuery с символом "$", чтобы он не был переопределён другой библиотекой во время выполнения.

```
(function( $ ) {
  $.fn.myPlugin = function() {

      // Тут пишем функционал нашего плагина

  };
})(jQuery);
```

Теперь у нас есть оболочка, внутри которой мы можем начать писать код плагина. Но прежде, чем мы начнём, я хотел бы сказать несколько слов о контексте. В непосредственной области видимости функции нашего плагина ключевое слово «this» ссылается на объект jQuery, для которого был вызван этот плагин. И тут часто ошибаются. Что, в свою очередь, приводит к тому, что разработчики дополнительно оборачивают «this» в функцию jQuery.

```
$(".block").click(function () {
  console.log($(this));
  console.log(this);
});
```

Напишем плагин для скрытия элементов на странице по клику на эти элементы.

```
(function ($) {
  $.fn.hideElement = function () {
    this.fadeOut(4000);
  };
})(jQuery);
```

```
$(".block").click(function () {
  $(this).hideElement();
});
```

Теперь, когда мы понимаем, как работать с контекстом, напишем плагин jQuery, который выполняет полезную работу. Например плагин, который выводит высоту самого наибольшего элемента.

```
(function ($) {
  $.fn.getMaxHeight = function () {
    let max = 0;
    this.each(function (e) {
      max = Math.max(max, $(this).height());
    });
    return max;
  };
})(jQuery);
```

```
$("#btn").click(function () {
  const max_height = $(".container .block").getMaxHeight();
  console.log(max_height);
});
```

Теперь покажу вам плагин для отладки. Который часто используют. Например узнать что за элемент у нас находится.

```
(function ($) {
  $.fn.checkError = function () {
    return this.each(function () {
      alert(`Tag Name: ${$(this).prop("tagName")}`);
    });
  };
})(jQuery);
```

```
$("button").click(function () {
  $(".block").checkError();
});
```

Написать плагин, который по клику на параграф с текстом цвет меняется на красный. Происходит выделение. Реализовать сначала на чистом JavaScript, затем через jQuery.

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

А потом уже через плагин.

```
$.fn.mySimplePlugin = function () {
    $(this).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. Нам нужно передавать параметры

```
(function ($) {
  const default_values = {
    color: "red",
    fontSize: "14px",
    fontStyle: "normal",
  };

  let new_options = {};

  $.fn.changeText = function (params) {
    new_options = $.extend({}, default_values, new_options, params);
    console.log(new_options);
    this.css({
      color: new_options.color,
      fontSize: new_options.fontSize,
      fontStyle: new_options.fontStyle,
    });
  };
})(jQuery);
```

```
$("p").click(function () {
  $(this).changeText({ color: "blue", fontSize: "50px", fontStyle: "italic" });
});
```

Дальше попробуйте реализовать через плагин галерею.