

ФУНКЦИИ БИТРИКС JS API (ВЕРСИЯ 1.2 ОТ ESTRIN.PW)

Функция битрикс api	Описание	Аналог в jquery
<u>Базовые функции</u>		
BX(function(){ }); BX.ready(function(){ });	Событие готовности DOM-структуры	\$(function(){ }); \$.ready(function(){ });
<u>Утилиты</u>		
BX.util.array_values(array) BX.util.array_keys(array) BX.util.array_merge(array) BX.util.array_unique(array) BX.util.in_array(array) BX.util.array_search(array) BX.util.trim(array) BX.util.htmlspecialchars(array) BX.util.htmlspecialcharsback(array) BX.util.preg_quote(array) BX.util.str_pad(array) BX.util.strip_tags(array)	Аналоги функций php	-
<u>Функции поиска DOM</u>		
var object = BX('my_id');	Поиск по идентификатору	var array = \$('#my-id');
var array = BX.findChildren(obParent, { tag: 'td', className: 'class-name' }, true); var object = BX.findChild(obParent, { tag: 'td', className: 'class-name' }, true, true);	Множественный рекурсивный поиск в DOM-структуре объекта	var array = arParent.find('td.class-name');
var object = BX.findChild(obParent, { tag: 'div', className: 'class-name'}, true);	Немножественный рекурсивный поиск в DOM-структуре объекта	var array = arParent.find('td.class-name:first'); var array = arParent.find('td.class-name').first;
var object = BX.findChild(obParent, { tag: 'div', className: 'class-name'});	Немножественный нерекурсивный поиск в DOM-структуре объекта	var array = arParent.children('td.class-name:first');
var object = BX.findChild(obParent, { tag: 'div', attribute: { data-id: '167' }});	Поиск по атрибутам (немнож., нерекурс.)	var array = arParent.children('div[data-id=167]');
var object = BX.findChild(obParent, { tag: 'div', property: { data-id: '167' }});	Поиск по свойствам (немнож., нерекурс.)	var array = arParent.children('div[data-id=167]');
var object = BX.findChild(obParent,function(el) { return !!el.tagName && el.tagName.toUpperCase()=='DIV' && el.hasAttribute('data-id'); });	Поиск со сложной логикой (немнож., нерекурс.)	var array = arParent.children('div[data-id]');

Функция битрикс ари	Описание	Аналог в jquery
<code>var parent = BX.findParent(object, {className : 'class-name'});</code>	Поиск среди родительских элементов DOM-структуры объекта	<code>var arParent = array.parents('.class-name:first');</code>
<code>var ob = BX.findNextSibling(object, {className : 'class-name'});</code> <code>var ob = BX.findPreviousSibling(object, {className : 'class-name'});</code>	Поиск предыдущих/следующих элементов DOM-структуры	<code>var ar = array.next();</code> <code>var ar = array.prev();</code>
<u>Управление DOM-структурой</u>		
<code>var object = BX.create({ tag: 'div', props: { name: 'val' }, attrs: { 'data-name': 'val' }, style: { name: 'val' }, events: {}, children: {}, text: "", // or html: "", // or });</code>	Создает объект и устанавливает его параметры	<code>\$('#<div>')</code> <code>.prop(name,val)</code> <code>.data('name','val')</code> <code>.css(name,val)</code> <code>.on(...)</code> <code>.append(...)</code> <code>.text("") // or</code> <code>.html(""); // or</code>
<code>var object = BX.adjust({ node: object, props: { name: 'val' }, style: { name: 'val' }, events: {}, children: {}, text: "", // or html: "", // or });</code>	Устанавливает параметры объекта	<code>\$(object)</code> <code>.prop(name,val)</code> <code>.css(name,val)</code> <code>.on(...)</code> <code>.append(...)</code> <code>.text("") // or</code> <code>.html(""); // or</code>
<code>BX.append(object,parent);</code>	Добавляет объект внутрь другого объекта	<code>\$(object).appendTo(parent);</code>
<code>BX.remove(object);</code>	Удаляет объект из структуры	<code>array.remove();</code>
<code>BX.cleanNode(object);</code>	Очищает объект от всех вложенных элементов	<code>array.empty();</code>
<u>Манипуляции с классами</u>		
<code>BX.addClass(object, 'class-name');</code>	Добавляет класс к элементу	<code>array.addClass('class-name');</code>
<code>BX.removeClass(object, 'class-name');</code>	Удаляет класс у элемента	<code>array.removeClass('class-name');</code>
<code>BX.toggleClass(object, 'class-name');</code>	Переключает класс у элемента	<code>array.toggleClass('class-name');</code>
<code>var bHasClass = BX.hasClass(object, 'class-name');</code>	Проверяет наличие класса у элемента	<code>var bHasClass = array.hasClass('class-name');</code>
<u>Data-атрибуты</u>		
<code>var value = BX.data(object,'name');</code>	Получает значение Data-атрибута	<code>var value = \$(object).data('name');</code>

Функция битрикс ари	Описание	Аналог в jquery
<code>BX.data(object,'name','value');</code>	Устанавливает значение Data-атрибута	<code>\$(object).data('name','value');</code>
<u>Манипуляции со стилями</u>		
<code>BX.style(object,'name','value');</code>	Устанавливает css свойство у элемента	<code>\$(object).css('name','value');</code>
<u>Функции обработки форм</u>		
<code>BX.submit(obForm);</code>	Отправляет форму на сервер	<code>arForm.submit();</code>
<code>BX.focus(object);</code>	Устанавливает фокус на поле ввода	<code>\$(object).focus();</code>
<u>События</u>		
<code>BX.bind(object, 'click', function() {});</code>	Устанавливает функцию обработки события для элемента	<code>\$(object).on('click',function({}));</code>
<u>Проxy-функции</u>		
<code>BX.proxy(function(params){ }, object)</code>	Инициализирует функцию с кастомной переменной this	<code>\$.proxy(function(params){}, object)</code>
<code>BX.bindDelegate(object, 'click', {tagName : 'a'}, BX.proxy(onClick, this));</code> <code>var onClick = function() {</code> <code> var target = BX.proxy_context;</code> <code>}</code>	Пример получения исходного объекта в Проxy-функции	<code>\$(object).on('click',</code> <code>\$.proxy(onClick,this));</code> <code>var onClick = function(event) {</code> <code> var target = event.target;</code> <code>}</code>
<u>Функции AJAX</u>		
<code>BX.ajax.post(url,{},function(res) {});</code>	Отправляет POST запрос	<code>\$.post(url,{},function(res) {});</code>
<code>BX.ajax.loadJSON(url,{},function(res) {});</code>	Отправляет POST запрос и получает результат в виде объекта	<code>\$.getJSON(url,{},function(res) {});</code>
<code>BX.ajax.InsertToNode(url,object);</code>	Отправляет запрос и вставляет результат в объект (можно указать id)	-
<u>Система кастомных событий</u>		
<code>BX.addCustomEvent(window, 'onEvent', function_name);</code>	Инициализирует глобальный обработчик события	<code>\$(document).bind('my.onEvent', function_name);</code>
<code>BX.removeCustomEvent(window, 'onEvent', function_name);</code>	Отключает глобальный обработчик события	<code>\$(document).unbind('my.onEvent', function_name);</code>
<code>BX.onCustomEvent(window, 'onEvent',[arg1,arg2,arg3]);</code>	Выполняет глобальный обработчики события	<code>\$.event.trigger('my.onEvent', [arg1,arg2,arg3]);</code>
<code>BX.addCustomEvent(object, 'onEvent', function_name);</code>	Инициализирует обработчик события для объекта	<code>object.on('my.onEvent', function_name);</code>
<code>BX.removeCustomEvent(object, 'onEvent', function_name);</code>	Отключает обработчик события для объекта	<code>object.off('my.onEvent', function_name);</code>
<code>BX.onCustomEvent(object, 'onEvent',[arg1,arg2,arg3]);</code>	Выполняет обработчики события для объекта	<code>object.trigger('my.onEvent', [arg1,arg2,arg3]);</code>
© estrin.pw - модули для битрикс		