

Лабораторная работа 11. jQuery UI. Плагин **draggable** - перетаскиваем элементы

Для работы с набором этих плагинов нам понадобится файл `ui.core.js`, который можно скачать на официальном сайте по адресу <http://jqueryui.com>.

Draggable-плагин позволяет перетаскивать элементы страницы с помощью мыши.

Создайте блок

```
<div id="les10_ex1">Перетащи меня</div>
```

Стиль блоку придайте сами, на странице `script.js` необходимо указать, что этот блок будет перемещаемым:

```
$(document).ready(init);  
function init(){  
    $("#les10_ex1").draggable();  
}
```

Перемещение элемента можно настраивать, для этого существует ряд параметров. Например:

axis - ограничивает перетаскивание элемента либо по горизонтали (x) либо по вертикали (y) оси координат.

Пример:

```
$(document).ready(init);  
function init(){  
    $("#les10_ex2").draggable({  
        axis: 'x'  
    });  
}
```

Теперь блок можно перетаскивать только по горизонтали.

`containment` - ограничивает область перетаскивания. Возможные значения: `parent` (родительский элемент), `document` (текущий документ), `window` (окно), `[x1, y1, x2, y2]` - координаты левого верхнего и правого нижнего угла допустимой области.

Пример:

```
$(document).ready(init);  
function init(){  
    $("#les10_ex3").draggable({  
        containment: 'parent'  
    });  
}
```

Теперь блок можно перетаскивать только в пределах родительского элемента (в пределах рамки).

`cursor` - определяет вид курсора при перетаскивании.

Пример:

```
$(document).ready(init);  
  
function init(){  
    $("#les10_ex4").draggable({  
        cursor: 'crosshair'  
    });  
}
```

Теперь при перетаскивании курсор принимает вид перекрестья в виде прицела. Другие возможные значения:

- `e-resize` - двунаправленная горизонтальная стрелка,
- `n-resize` - двунаправленная вертикальная стрелка,
- `ne-resize` - двунаправленная диагональная стрелка (из левого нижнего угла в верхний правый угол),
- `nw-resize` - двунаправленная диагональная стрелка (из правого нижнего угла в верхний левый угол),
- `move` - перекрестье из четырёх стрелок,
- `text` - обычный текстовый курсор,
- `wait` - песочные часы Windows.

`delay` - определяет время в миллисекундах, по истечении которого, начнется перетаскивание (полезно для предотвращения случайного перетаскивания).

Пример:

```
$(document).ready(init);

function init(){
    $("#les10_ex5").draggable({
        delay: 2000
    });
}
```

Теперь перетаскивание будет начинаться через 2 секунды после нажатия и удержания левой кнопки мыши.

`distance` - определяет расстояние в пикселах, которое указатель мыши должен пройти после того, как вы нажали на кнопку мыши, прежде чем начнется перетаскивание (так же, как и предыдущий параметр полезен для предотвращения случайного перетаскивания).

Пример:

```
$(document).ready(init);  
function init(){  
    $("#les10_ex6").draggable({  
        distance: 50  
    });  
}
```

Теперь перетаскивание будет начинаться только после того, как указатель мыши пройдет 50 пикселей, причем неважно в какую сторону.

`handle` - определяет, за какой элемент внутри данного осуществляется перетаскивание.

Пример:

```
$(document).ready(init);  
  
function init(){  
    $("#les10_ex7").draggable({  
        handle: 'p'  
    });  
}
```

```
}
```

Теперь перетаскивать блок можно только, ухватив за абзац (за текст).

`revert` - определяет, будет ли элемент возвращен на свое место после перетаскивания (`true` - да).

Пример:

```
$(document).ready(init);  
function init(){  
    $("#les10_ex8").draggable({  
        revert: true  
    });  
}
```

Теперь после перетаскивания блок будет возвращаться на свое место.

`revertDuration` - определяет, за какое время элемент будет возвращен на свое место после перетаскивания (если предыдущий параметр определен, как `true`).

Пример:

```
$(document).ready(init);  
function init(){  
    $("#les10_ex9").draggable({  
        revert: true,  
        revertDuration: 3000  
    });  
}
```

Теперь после перетаскивания блок будет возвращаться на свое место за 3 секунды.

`zIndex` - определяет значение свойства `z-index` (номер слоя) перемещаемого элемента.

Пример:

```
$(document).ready(init);  
function init(){  
    $("#les10_ex10").draggable({  
        zIndex: 1  
    });  
    $("#les10_ex11").draggable({  
        zIndex: 2  
    });  
    $("#les10_ex12").draggable({  
        zIndex: 3  
    });  
}
```

z-index: 1

z-index: 2

z-index: 3

Теперь в какой бы последовательности мы не перетаскивали наши блоки, они всегда будут располагаться соответственно своим слоям (самый верхний 3, под ним 2, в самом низу 1).

А сейчас мы рассмотрим три параметра, связанных с событиями:

`start` - событие наступает с началом перемещения.

`drag` - событие происходит в течение всего времени перемещения.

`stop` - событие наступает при завершении перемещения.

При использовании любого из этих параметров мы должны определить функцию, которая принимает два аргумента – `event` (объект события) и `ui` (специальный объект, содержащий информацию о положении перемещаемого объекта).

Используя свойства этих аргументов, можно узнать координаты объекта в любой момент при перемещении и после него. Например:

```
<div id="les10_ex13"></div>
<textarea cols="70" rows="3" wrap="off"
id="res"></textarea>
```

Функция:

```
$(document).ready(init);
function init(){
start: function(event, ui) {
var a=event.type;
$('#res').text('Событие: '+a);
},
stop: function(event, ui) {
var a=event.type;
var c=ui.position.left;
var d=ui.position.top;
var e=ui.offset.left;
var f=ui.offset.top;
$('#res').text('Событие: '+a+
```

```
'\n Расстояние слева и сверху относительно  
родительского элемента: '+c+' '+d+'  
'\n Абсолютные значения расстояния слева и сверху:  
'+e+' '+f);  
}  
}
```

Всего у плагина `draggable` более 20 различных параметров, полный перечень можете посмотреть на сайте.

У плагина `draggable` есть четыре метода, которые позволяют **менять работу плагина после его инициализации**:

`destroy` - удаляет всю функциональность плагина `Draggable`.

Пример:

```
$("#les").draggable('destroy');
```

`disable` - временно запрещает работу всей функциональности плагина `Draggable`.

Пример:

```
$("#les").draggable('disable');
```

`enable` - разрешает работу всей функциональности плагина `Draggable` (если она была запрещена).

Пример:

```
$("#les").draggable('enable');
```


`option` - позволяет установить значения параметров плагина после его инициализации.

Запись:

```
$("#les").draggable('option', optionName, [value]);
```

Пример:

```
<div id="les10_ex14"></div>
<input type="button" value="Метод option" id="metOpt">
<input type="button" value="Метод disable"
id="metDis">
<input type="button" value="Метод enable" id="metEn">
```

Функция:

```
$(document).ready(init);
function init(){
$("#les10_ex14").draggable({
  revert: true,
  cursor: 'crosshair'
});
$("#metOpt").click(function(){
$("#les10_ex14").draggable("option", "revertDuration",
[5000]);
});
$("#metDis").click(function(){
$("#les10_ex14").draggable("disable");
```

```
});  
$("#metEn").click(function() {  
    $("#les10_ex14").draggable("enable");  
});  
}
```