

Лабораторная работа 12. Плагин **droppable** - сбрасываем элементы

Плагин **Draggable** позволяет перемещаемые элементы сбрасывать в какой-либо другой элемент (самый распространенный пример - корзина).

У плагина **droppable** есть ряд параметров:

accept - целевой элемент будет принимать только те элементы, которые подходят по селектору. В нашем примере, благодаря этому параметру темные шашки суммируются только, если помещены в лоток для темных шашек, аналогично и для светлых шашек.

Пример:

```
$("#lotok1").droppable({  
  accept: '.t_sh'  
});
```

activeClass - здесь можно указать имя класса, который будет применен к целевому элементу в момент начала перемещения. В нашем примере, при перемещении шашек, фон соответствующего лотка становится желтым.

Пример:

```
$("#lotok1").droppable({  
  activeClass: 'active'  
});
```

hoverClass - здесь можно указать имя класса, который будет применен к целевому элементу в момент, когда перемещаемый элемент

окажется над целевым элементом. В нашем примере, фон соответствующего лотка становится зеленым.

Пример:

```
$("#lotok1").droppable({  
  hoverClass: 'hover'  
});
```

`greedy` - этот параметр используется, если целевой элемент является вложенным в другой целевой элемент. По умолчанию параметр имеет значение `false`. Если установить в `true`, то сброс будет происходить во внутренний целевой элемент, но не в элемент-родитель.

Пример:

```
$("#lotok1").droppable({  
  greedy: true  
});
```

`tolerance` - определяет, когда будет происходить "сброс" элемента, при нахождении его над целевым элементом. Возможны следующие значения:

- `intersect` - перемещаемый элемент перекрыл целевой хотя бы на 50% (значение по умолчанию),
- `fit` - перемещаемый элемент полностью вошел в целевой элемент,
- `pointer` - указатель мыши вошел в целевой элемент,
- `touch` - перемещаемый элемент пересек границу целевого.

Пример:

```
$("#lotok1").droppable({  
  tolerance: 'fit'  
});
```

Следующие пять параметров связаны с событиями:

`activate` - событие наступает с началом перемещения.

`deactivate` - событие наступает в момент окончания перемещения.

`over` - событие наступает при входе перемещаемого элемента в целевой.

`out` - событие наступает при выходе перемещаемого элемента из целевого.

`drop` - событие наступает в момент "сброса" перемещаемого элемента в целевой.

При использовании этих параметров мы можем 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:

- `ui.position.left` - положение перемещаемого элемента относительно родительского элемента слева,
- `ui.position.top` - положение перемещаемого элемента относительно родительского элемента сверху,
- `ui.offset.left` - абсолютное положение перемещаемого элемента слева,

- `ui.offset.top` - абсолютное положение перемещаемого элемента сверху.

Пример:

```
$("#lotok1").droppable({
  deactivate: function(event, ui) {
    var a=event.type;
    var b=ui.position.left;
    var c=ui.position.top;
    $('#res').text('Событие: '+a+' Координаты элемента: '+b+', '+c);
  }
});
```

Также, как и у плагина `draggable`, у плагина `droppable` есть четыре метода, которые позволяют менять его работу после инициализации:

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

Пример:

```
$("#lotok1").droppable('destroy');
```

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

Пример:

```
$("#lotok1").droppable('disable');
```

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

Пример:

```
$("#lotok1").draggable('enable');
```

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

Запись:

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

Задание:

Использовать шахматную доску и шашки, сделанные ранее. Сделать так, чтобы шашками можно было «ходить» (т.е. перемещать их) и дополнить игру двумя лотками, куда будем перемещать «съеденные» шашки.

Суммируются шашки только тогда, когда помещены в предназначенный для них лоток.

Кроме того, подсчитывается количество перемещений шашек, а не их количество в лотке. (Подключаем `custom.min.js`)

```
<div id="board"></div>
<div id='lotki'>
<div id='lotok1'>Для темных шашек
<div id="res1">0</div>
</div>
<div id='lotok2'>Для светлых шашек
<div id="res2">0</div>
</div>
</div>
```

- 1) На странице `script.js` «нарисовать» доску (функция `addBoard`)
- 2) Разместить на ней шашки (функция `addDraughts`).
- 3) Сделать шашки перемещаемыми (плагин `draggable`)
- 4) Определить лотки, как целевые элементы (плагин `droppable`).