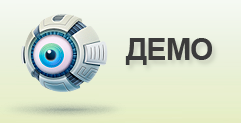
**Эффект приближения для миниатюр**

Создадим эффект приближения для миниатюр с помощью jQuery. Масштаб миниатюр будет изменяться в зависимости от расстояния между ними и курсором мыши.

В уроке используется [плагин jQuery Proximity](https://github.com/padolsey/jQuery-Plugins/tree/master/proximity-event), который написал [James Padolsey](http://james.padolsey.com/).

[](http://ruseller.com/lessons/les1359/demo/index.html)[](http://ruseller.com/lessons/les1359/demo_1359.zip)

**Разметка HTML**

Мы будем использовать неупорядоченный список миниатюр с секцией описания (в первой демонстрации не используется):

|  |  |  |
| --- | --- | --- |
| 01 | <ul id="pe-thumbs" class="pe-thumbs"> | |
| 02 | <li> |

|  |  |
| --- | --- |
| 03 | <a href="#"> |
| 04 | <img src="images/thumbs/1.jpg" /> | |

|  |  |
| --- | --- |
| 05 | <div class="pe-description"> |
| 06 | <h3>Заголовок описания</h3> | |

|  |  |  |
| --- | --- | --- |
| 07 | <p>Текст описания</p> | |
| 08 | </div></a> |

|  |  |
| --- | --- |
| 09 | </li> |
| 10 | <li><!-- ... --></li> | |

|  |  |
| --- | --- |
| 11 | </ul> |

**CSS**

Неупорядоченный список будет центрироваться на странице и иметь тематическое фоновое изображение, просвечивающее через миниатюры с низким значение непрозрачности:

|  |  |
| --- | --- |
| 1 | .pe-thumbs{ |
| 2 | width: 900px; | |

|  |  |
| --- | --- |
| 3 | height: 400px; |
| 4 | margin: 20px auto; | |

|  |  |
| --- | --- |
| 5 | position: relative; |
| 6 | background: transparent url(../images/3.jpg) top center; | |

|  |  |
| --- | --- |
| 7 | border: 5px solid #fff; |
| 8 | box-shadow: 0 1px 2px rgba(0,0,0,0.2); | |

|  |  |
| --- | --- |
| 9 | } |

Мы также слегка затонируем фоновое изображение с помощью псевдо-элемента с фоновым цветом в формате *rgba*:

|  |  |  |
| --- | --- | --- |
| 01 | .pe-thumbs:before { | |
| 02 | content: ""; |

|  |  |
| --- | --- |
| 03 | display: block; |
| 04 | position: absolute; | |

|  |  |
| --- | --- |
| 05 | top: 0px; |
| 06 | left: 0px; | |

|  |  |
| --- | --- |
| 07 | width: 100%; |
| 08 | height: 100%; | |

|  |  |  |
| --- | --- | --- |
| 09 | background: rgba(255,102,0,0.2); | |
| 10 | } |

Список пунктов будет смещаться влево и мы установим относительное положение ссылок и изображений:

|  |  |
| --- | --- |
| 1 | .pe-thumbs li{ |
| 2 | float: left; | |

|  |  |  |
| --- | --- | --- |
| 3 | position: relative; | |
| 4 | } |

|  |  |
| --- | --- |
| 5 | .pe-thumbs li a, |
| 6 | .pe-thumbs li a img{ | |

|  |  |
| --- | --- |
| 7 | display: block; |
| 8 | position: relative; | |

|  |  |
| --- | --- |
| 9 | } |

Каждая миниатюра будет иметь начальную ширину 100 px и непрозрачность 0.2:

.pe-thumbs li a img{

    width: 100px;

    opacity: 0.2;

}

Описание будет скрыто по умолчанию и затем выводиться с помощью JavaScript:

|  |  |  |
| --- | --- | --- |
| 01 | .pe-thumbs li a div.pe-description{ | |
| 02 | width: 200px; |

|  |  |
| --- | --- |
| 03 | height: 100px; |
| 04 | background: rgba(0,0,0,0.8); | |

|  |  |  |
| --- | --- | --- |
| 05 | position: absolute; | |
| 06 | top: 0px; |

|  |  |
| --- | --- |
| 07 | left: -200px; |
| 08 | color: white; |

|  |  |
| --- | --- |
| 09 | display: none; |
| 10 | z-index: 1001; |

|  |  |  |
| --- | --- | --- |
| 11 | text-align: left; | |
| 12 | } |

В завершение определим стили для описания:

|  |  |
| --- | --- |
| 01 | .pe-description h3{ |
| 02 | padding: 10px 10px 0px 10px; | |

|  |  |
| --- | --- |
| 03 | line-height: 20px; |
| 04 | font-family: 'BebasNeueRegular', 'Arial Narrow', Arial, sans-serif; | |

|  |  |  |
| --- | --- | --- |
| 05 | font-size: 22px; | |
| 06 | margin: 0px; |

|  |  |
| --- | --- |
| 07 | } |
| 08 | .pe-description p{ | |

|  |  |  |
| --- | --- | --- |
| 09 | padding: 10px 0px; | |
| 10 | margin: 10px; |

|  |  |
| --- | --- |
| 11 | font-size: 11px; |
| 12 | font-style: italic; | |

|  |  |  |
| --- | --- | --- |
| 13 | border-top: 1px solid rgba(255,255,255,0.3); | |
| 14 | } |

**JavaScript**

Сначала рассчитаем размер контейнер с описанием и его положение при наведении курсора мыши на миниатюру. Данные значения зависят от максимального масштаба миниатюры и ее расположения в основном контейнере. Например, когда миниатюра находится рядом с правым краем, описание должно появляться слева от миниатюры.

Затем надо привязать событие приближения к изображению. Масштаб изображения будет изменяться в соответствии с положением курсора мыши. Как только масштаб получит максимальное значение, установим большой *z-index*, чтобы миниатюра выводилась поверх всего, и выведем соответствующее описание:

|  |  |
| --- | --- |
| 001 | // Список миниатюр |
| 002 | var $list       = $('#pe-thumbs'), | |

|  |  |
| --- | --- |
| 003 | // Ширина и смещение влево списка. |
| 004 | // Данные значения будут использоваться для получения положения контейнера описания. | |

|  |  |
| --- | --- |
| 005 | listW       = $list.width(), |
| 006 | listL       = $list.offset().left, | |

|  |  |
| --- | --- |
| 007 | // Изображения |
| 008 | $elems      = $list.find('img'), | |

|  |  |
| --- | --- |
| 009 | // Контейнер описания |
| 010 | $descrp     = $list.find('div.pe-description'), | |

|  |  |
| --- | --- |
| 011 | // maxScale : максимально значение масштаба, которое может иметь изображение |
| 012 | // minOpacity / maxOpacity : минимальное (устанавливается в CSS) и максимальное значение непрозрачности изображения |

|  |  |
| --- | --- |
| 013 | settings    = { |
| 014 | maxScale    : 1.3, | |

|  |  |
| --- | --- |
| 015 | maxOpacity  : 0.9, |
| 016 | minOpacity  : Number( $elems.css('opacity') ) | |

|  |  |
| --- | --- |
| 017 | }, |
| 018 | init        = function() { | |

|  |  |
| --- | --- |
| 019 |  |
| 020 | // minScale устанавливается в CSS | |

|  |  |  |
| --- | --- | --- |
| 021 | settings.minScale = \_getScaleVal() || 1; | |
| 022 | // Предварительная загрузка миниатюр |

|  |  |  |
| --- | --- | --- |
| 023 | \_loadImages( function() { | |
| 024 |  |

|  |  |
| --- | --- |
| 025 | \_calcDescrp(); |
| 026 | \_initEvents(); |

|  |  |
| --- | --- |
| 027 |  |
| 028 | }); | |

|  |  |
| --- | --- |
| 029 |  |
| 030 | }, | |

|  |  |  |
| --- | --- | --- |
| 031 | // Получаем значение масштаба из CSS с помощью JavaScript: | |
| 032 | \_getScaleVal= function() { |

|  |  |
| --- | --- |
| 033 |  |
| 034 | var st = window.getComputedStyle($elems.get(0), null), | |

|  |  |  |
| --- | --- | --- |
| 035 | tr = st.getPropertyValue("-webkit-transform") || | |
| 036 | st.getPropertyValue("-moz-transform") || |

|  |  |  |
| --- | --- | --- |
| 037 | st.getPropertyValue("-ms-transform") || | |
| 038 | st.getPropertyValue("-o-transform") || |

|  |  |  |
| --- | --- | --- |
| 039 | st.getPropertyValue("transform") || | |
| 040 | "fail..."; |

|  |  |
| --- | --- |
| 041 |  |
| 042 | if( tr !== 'none' ) { | |

|  |  |
| --- | --- |
| 043 |  |
| 044 | var values = tr.split('(')[1].split(')')[0].split(','), | |

|  |  |
| --- | --- |
| 045 | a = values[0], |
| 046 | b = values[1], |

|  |  |
| --- | --- |
| 047 | c = values[2], |
| 048 | d = values[3]; |

|  |  |
| --- | --- |
| 049 |  |
| 050 | return Math.sqrt( a \* a + b \* b ); | |

|  |  |
| --- | --- |
| 051 |  |
| 052 | } | |

|  |  |  |
| --- | --- | --- |
| 053 |  | |
| 054 | }, |

|  |  |  |
| --- | --- | --- |
| 055 | // Вычисляем значения для контейнера описания | |
| 056 | // на основе установочных переменных |

|  |  |  |
| --- | --- | --- |
| 057 | \_calcDescrp = function() { | |
| 058 |  |

|  |  |  |
| --- | --- | --- |
| 059 | $descrp.each( function(i) { | |
| 060 |  |

|  |  |
| --- | --- |
| 061 | var $el     = $(this), |
| 062 | $img    = $el.prev(), | |

|  |  |
| --- | --- |
| 063 | img\_w   = $img.width(), |
| 064 | img\_h   = $img.height(), | |

|  |  |
| --- | --- |
| 065 | img\_n\_w = settings.maxScale \* img\_w, |
| 066 | img\_n\_h = settings.maxScale \* img\_h, |

|  |  |
| --- | --- |
| 067 | space\_t = ( img\_n\_h - img\_h ) / 2, |
| 068 | space\_l = ( img\_n\_w - img\_w ) / 2; |

|  |  |
| --- | --- |
| 069 |  |
| 070 | $el.data( 'space\_l', space\_l ).css({ | |

|  |  |  |
| --- | --- | --- |
| 071 | height  : settings.maxScale \* $el.height(), | |
| 072 | top     : -space\_t, |

|  |  |  |
| --- | --- | --- |
| 073 | left    : img\_n\_w - space\_l | |
| 074 | }); |

|  |  |
| --- | --- |
| 075 |  |
| 076 | }); | |

|  |  |
| --- | --- |
| 077 |  |
| 078 | }, | |

|  |  |  |
| --- | --- | --- |
| 079 | \_initEvents = function() { | |
| 080 |  |

|  |  |  |
| --- | --- | --- |
| 081 | $elems.on('proximity.Photo', { max: 80, throttle: 10, fireOutOfBounds :true }, function(event, proximity, distance) { | |
| 082 |  |

|  |  |
| --- | --- |
| 083 | var $el         = $(this), |
| 084 | $li         = $el.closest('li'), | |

|  |  |
| --- | --- |
| 085 | $desc       = $el.next(), |
| 086 | scaleVal    = proximity \* ( settings.maxScale - settings.minScale ) + settings.minScale, | |

|  |  |  |
| --- | --- | --- |
| 087 | scaleExp    = 'scale(' + scaleVal + ')'; | |
| 088 |  |

|  |  |  |
| --- | --- | --- |
| 089 | // Изменяем z-index элемента, как только будет достигнуто максимальное значение масштаба. | |
| 090 | // Также показываем контейнер описания |

|  |  |  |
| --- | --- | --- |
| 091 | if( scaleVal === settings.maxScale ) { | |
| 092 |  |

|  |  |  |
| --- | --- | --- |
| 093 | $li.css( 'z-index', 1000 ); | |
| 094 |  |

|  |  |  |
| --- | --- | --- |
| 095 | if( $desc.offset().left + $desc.width() > listL + listW ) { | |
| 096 |  |

|  |  |  |
| --- | --- | --- |
| 097 | $desc.css( 'left', -$desc.width() - $desc.data( 'space\_l' ) ); | |
| 098 |  |

|  |  |  |
| --- | --- | --- |
| 099 | } | |
| 100 |  |

|  |  |  |
| --- | --- | --- |
| 101 | $desc.fadeIn( 800 ); | |
| 102 |  |

|  |  |
| --- | --- |
| 103 | } |
| 104 | else { | |

|  |  |
| --- | --- |
| 105 |  |
| 106 | $li.css( 'z-index', 1 ); | |

|  |  |
| --- | --- |
| 107 |  |
| 108 | $desc.stop(true,true).hide(); | |

|  |  |
| --- | --- |
| 109 |  |
| 110 | } | |

|  |  |
| --- | --- |
| 111 |  |
| 112 | $el.css({ | |

|  |  |  |
| --- | --- | --- |
| 113 | '-webkit-transform' : scaleExp, | |
| 114 | '-moz-transform'    : scaleExp, |

|  |  |
| --- | --- |
| 115 | '-o-transform'      : scaleExp, |
| 116 | '-ms-transform'     : scaleExp, | |

|  |  |
| --- | --- |
| 117 | 'transform'         : scaleExp, |
| 118 | 'opacity'           : ( proximity \* ( settings.maxOpacity - settings.minOpacity ) + settings.minOpacity ) | |

|  |  |  |
| --- | --- | --- |
| 119 | }); | |
| 120 |  |

|  |  |  |
| --- | --- | --- |
| 121 | }); | |
| 122 |  |

|  |  |
| --- | --- |
| 123 | }, |
| 124 | \_loadImages = function( callback ) { | |

|  |  |
| --- | --- |
| 125 |  |
| 126 | var loaded  = 0, | |

|  |  |  |
| --- | --- | --- |
| 127 | total   = $elems.length; | |
| 128 |  |

|  |  |  |
| --- | --- | --- |
| 129 | $elems.each( function(i) { | |
| 130 |  |

|  |  |  |
| --- | --- | --- |
| 131 | var $el = $(this); | |
| 132 |  |

|  |  |  |
| --- | --- | --- |
| 133 | $('<img/>').load( function() { | |
| 134 |  |

|  |  |
| --- | --- |
| 135 | ++loaded; |
| 136 | if( loaded === total ) | |

|  |  |  |
| --- | --- | --- |
| 137 | callback.call(); | |
| 138 |  |

|  |  |  |
| --- | --- | --- |
| 139 | }).attr( 'src', $el.attr('src') ); | |
| 140 |  |

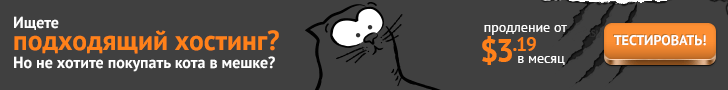
|  |  |  |
| --- | --- | --- |
| 141 | }); | |
| 142 |  |

|  |  |  |
| --- | --- | --- |
| 143 | }; | |
| 144 |  |

|  |  |
| --- | --- |
| 145 | return { |
| 146 | init    : init | |

|  |  |
| --- | --- |
| 147 | }; |

Данный урок подготовлен для вас командой сайта [ruseller.com](http://ruseller.com/)  
Источник урока: [tympanus.net/codrops/2012/01/04/thumbnail-proximity-effect/](http://ruseller.com/tympanus.net/codrops/2012/01/04/thumbnail-proximity-effect/)  
Перевел: Сергей Фастунов  
Урок создан: 2 Февраля 2012  
Просмотров: 43929  
[Правила перепечатки](http://ruseller.com/page.php?doc=reprint)+

[](https://hostiq.com.ua/virtual-hosting/?utm_source=ruseller.com&utm_medium=sponsored&utm_campaign=ruseller-banner-flash)

**5 последних уроков рубрики "jQuery"**

* http://ruseller.com/lessons/les2788/thumbnail.png

[**Вычисление позиции курсора мыши**](http://ruseller.com/lessons.php?rub=32&id=2788)

Данный сниппет поможет вам определить координаты пользовательского курсора практически во всех браузерах.

* http://ruseller.com/lessons/les2787/thumbnail.png

[**Отключение прокрутки на Google картах**](http://ruseller.com/lessons.php?rub=32&id=2787)

Скорее всего каждый из нас хоть один раз вставлял Google карты на свои страницы. Тогда вы не могли не столкнуться с проблемой: во время прокрутки страницы, если курсор попадал на карту Google, то вместо смещения страницы, начинала масштабироваться карта. Мы ответим на вопрос о том, как отключить прокрутку в Google картах.

* http://ruseller.com/lessons/les2781/thumbnail.jpeg

[**16 jQuery плагинов для создания “каруселей”**](http://ruseller.com/lessons.php?rub=32&id=2781)

Данные jQuery плагины позволят быстро и безболезненно создать слайдер контента.

* http://ruseller.com/lessons/les2780/thumbnail.jpg

[**Извлечение GET параметров через JavaScript**](http://ruseller.com/lessons.php?rub=32&id=2780)

Параметры в URL могут содержать кучу полезной информации: данные о товаре, пользовательские настройки, идентификатор партнёра. В этой статье посмотрим как через JS можно извлечь данную информацию.

* http://ruseller.com/lessons/les2777/thumbnail.jpg

[**Удаление элементов через JavaScript**](http://ruseller.com/lessons.php?rub=32&id=2777)

Существует несколько методов удаления элементов через JavaScript: традиционный, перебирая элементы и новый - прямой метод DOM4.