jQuery делает тривиальным добавление простых эффектов на страницу. Эффекты могут использовать встроенные настройки или подгонять продолжительность анимации индивидуально. Вы также можете создавать собственную анимацию произвольных свойств CSS.

Встроенные эффекты

Часто используемые эффекты встроены в jQuery как методы, которые вы можете вызвать для любого объекта jQuery:

.show() — показать выбранные элементы;

.hide() — скрыть выбранные элементы;

.fadeIn() — анимация прозрачности выбранных элементов до 0%;

.fadeOut() — анимация прозрачности выбранных элементов до 100%;

.fadeTo() — анимация прозрачности выбранных элементов до n%;

.slideDown() — отображение выбранных элементов с помощью вертикального скользящего движения;

.slideUp() — сокрытие выбранные элементы с помощью вертикального скользящего движения;

.slideToggle() — показать или скрыть выбранные элементы с вертикальным скользящим движением в зависимости от того, видны элементы в данный момент или нет.

После создания выборки мы просто применяем к ней эффект.

```
$( '.hidden' ).show();
```

Вы также можете указать длительность встроенных эффектов. Есть два способа сделать это: можете задать время в миллисекундах

```
$( '.hidden' ).show( 300 );
```

или использовать одну из предустановленных скоростей:

```
$( '.hidden' ).show( 'slow' );
```

Предустановленные скорости указаны в объекте jQuery.fx.speeds. Вы можете изменить этот объект, чтобы переопределить значения по умолчанию или расширить его новыми именами:

```
// Переустановить имеющуюся предустановленную скорость
jQuery.fx.speeds.fast = 50;
// Создать новую предустановленную скорость
jQuery.fx.speeds.turtle = 3000;
// Поскольку мы переустановили скорость 'fast', то теперь анимация
// длится 50 миллисекунд
$( '.hidden' ).hide( 'fast' );
// После их создания мы можем использовать пользовательские скорости
// подобно встроенным
$( '.other-hidden' ).show( 'turtle' );
```

Часто вам требуется сделать что-то после завершения анимации — если вы попытаетесь сделать это до окончания анимации, то это может повлиять на качество анимации или привести к удалению элементов, которые являются частью анимации.

Вы можете предоставить функцию обратного вызова для методов анимации, если желаете указать, что должно произойти после завершения эффект. Внутри этой функции this указывает на исходный элемент DOM, к которому применялся эффект. Подобно событиям мы можем превратить его в объект jQuery через $(this).

```
$( 'p.old' ).fadeOut( 300, function() {
  $( this ).remove();
});
```

Произвольные эффекты с .animate()

Если встроенные анимации не подходят, вы можете использовать .animate() для создания произвольной анимации большинства свойств CSS. Учтите, что

вы не можете анимировать свойство color, но есть плагин, который делает его возможным.

У метода .animate() есть три аргумента:

- объект, определяющий свойства для анимации;
- продолжительность анимации в миллисекундах;
- функция обратного вызова, которая будет вызываться после окончания анимации.

Метод .animate() может анимировать до указанного конечного значения или увеличить существующее значение.

```
$( '.funtimes' ).animate({
   left: '+=50', // увеличить на 50
   opacity: 0.25,
   fontSize: '12px'
 },
 300,
 function() {
   // выполняется, когда анимация завершена
 }
);
```

Обратите внимание, что если вы хотите анимировать свойство CSS, название которого включает дефис, то нужно использовать CamelCase вариант написания свойства, если вы не возьмёте его в кавычки. К примеру, свойство font-size указывается как fontSize.

Управление анимацией

jQuery предлагает два важных метода для управления анимацией.

.stop() — останавливает выполняемую в данное время анимацию для выбранных элементов.

```
$('.panel').stop();            // останавливаем выполнение текущей анимации
$('.panel').stop(true);        // останавливаем выполнение текущей анимации и всех последующих (чистим очередь)
```

$('.panel').stop(true, true); // останавливаем выполнение текущей анимации и всех последующих, но применяем результат текущей

$('.panel').stop(false, true); // останавливаем выполнение только текущей анимации, и применяем её результат

.delay() — пауза перед выполнением следующей анимации. В качестве аргумента передаётся желаемое время ожидания в миллисекундах.

Практическое задание

Встроить к себе в страницу анимацию, используя предопределенные методы и метод animate. Расширить или изменить объект fx.speeds.