

Лабораторная работа № 32

Тема работы: «Создание анимаций при помощи библиотеки jQuery»

1. Цель работы

Формирование умений работы с методами `animate()`.

2. Задание

Создать HTML-документ, содержащий сценарии JavaScript.

3. Оснащение работы

ПК, текстовый редактор, браузер.

4. Основные теоретические сведения

Эффекты, которых нет в библиотеке jQuery, можно создавать с помощью метода `.animate()`. Интерпретатор браузера динамически, без перезагрузки страницы, изменяет выбранные свойства на указанные значения. Анимация происходит для всех элементов обернутого набора. Чтобы добавить эффекты для конкретного элемента, нужно воспользоваться фильтрами jQuery для отбора.

Метод позволяет анимировать любое CSS-свойство, имеющее числовое значение, например, `font-size`, `opacity`, `border-width`, `margin`, `padding`, `height`, `width`, `background-position` и т.д. При этом имена свойств должны быть указаны слитно — `fontSize`, `paddingLeft`, или должен использоваться CSS-эквивалент свойства — `"font-size"`. Числовые значения свойств не заключаются в кавычки.

Для любого свойства предварительно должно быть установлено начальное значение, а в CSS-объявлении должна использоваться полная запись каждого свойства, т.е., вместо свойства `border` должно быть заданы значения для `border-style`, `border-width` и т.д.

Функция обратного вызова вызывается один раз после завершения анимации. Функции не передается никаких аргументов, но анимации выполняется для элемента, переданного свойству `this` в качестве контекста.

Значениями свойств могут также выступать `hide`, `show` или `toggle`, в результате чего к элементу применится вычисляемое значение — отображение, скрывание или переключение исходных состояний свойств.

Метод `.animate()` позволяет изменять CSS-свойства выбранных элементов с возможностью одновременной анимации нескольких свойств, задавая продолжительность анимации в миллисекундах.

```
$("#div").animate({left: "200px", top: "200px"}, 500);
```

Данная анимация одновременно применяется к свойству `left`, для которого задано значение `200px`, и к свойству `top` со значением `200px`, продолжительность анимации задана `500ms`.

Для элемента можно задавать относительное перемещение при каждом вызове анимации с помощью операторов `+=`, `-=`, `*=`, `/=`, например, `$("#div").animate({left: "+=200", top: "-=200"}, 500);`.

Метод `.animate()` имеет две формы записи. В первой методу передаются четыре аргумента:

- `.animate({свойство1: «значение1», свойство2: «значение2»}, продолжительность, функция перехода, function() {...});`
- `{свойство1: «значение1»}` — объект свойств, которые собираемся анимировать в формате `свойство: «значение»`, перечисленные в фигурных скобках через запятую.

Продолжительность — необязательный параметр продолжительности анимации, задающий время в миллисекундах или с помощью ключевых слов `slow`, `fast`, `normal`.

Функция перехода — необязательное имя функции перехода.

`function() {...}` — необязательная функция обратного вызова.

Вторая форма принимает два аргумента: объект свойств и объект дополнительных функциональных возможностей.

`Animate({свойство1: «значение1», свойство2: значение2}, {duration: «значение», easing: «значение», specialEasing: {`

`Свойство1: «easing1», свойство2: «easing2»}, complete: function() {...}, queue: true, step: callback});`

`{свойство1: «значение1»}` — объект свойств, которые собираемся анимировать в формате `свойство: «значение»`, перечисленные в фигурных скобках через запятую.

`Duration` — число или ключевое слово, значение по умолчанию 400. Устанавливает продолжительность анимации.

`Easing` — имя функции перехода. Значение по умолчанию `swing` — колебательный переход, второе доступное значение — `linear` — линейный переход. Расширенные возможности реализуются с помощью плагинов, смотрите jQuery UI.

`Queue` — помещает анимацию в очередь эффектов, по умолчанию `true`, значение `false` — немедленно воспроизводит анимацию.

`SpecialEasing` — объект, содержащий одно и более свойств и их значений, описывающих функции перехода.

`Step` — функция, которая будет вызываться по окончании каждого этапа анимации и будет применяться для каждого элемента обернутого набора. Функции передается порядковый номер этапа и внутренний объект, описывающий эффект.

`Progress` — функция, которая будет вызываться после каждого шага анимации, только один раз за анимацию независимо от количества анимационных свойств.

`Complete` — функция, которая должна быть вызвана по окончании воспроизведения анимации.

`Start` — функция, которая будет вызвана в начале анимации.

`Done` — функция, вызванная по завершении анимации.

`Fail` — функция, вызываемая в случае невозможности завершения анимации.

`Always` — функция, которая будет вызвана в случае остановки или неполного завершения анимации.

Метод `.fadeIn()` управляет прозрачностью, показывая скрытый элемент, при этом свойство `opacity` выбранного элемента изменяется от 0 до 1. Для этого на странице появляется необходимое пространство для элемента, при этом остальные элементы могут сдвинуться с места.

`.fadeIn(длительность, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта с помощью ключевых слов "fast", "normal", "slow" или числовых значений. По умолчанию используется значение "normal", равное 400 миллисекундам.

Функция по завершении анимации — необязательный параметр, задает функцию, которая будет вызвана после проявления элемента.

`.fadeIn(объект свойств)`

Объект свойств — представляет собой пары дополнительных опций, указанных в формате свойство: «значение»

`.fadeIn(длительность, функция перехода, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта.

Функция перехода — необязательный параметр, задает функцию перехода. Значение по умолчанию `swing` — колебательный переход, второе доступное значение — `linear` — линейный переход.

Функция по завершении анимации — необязательный параметр, задает функцию, которая будет вызвана после проявления элемента.

Метод `.fadeOut()` заставляет элемент исчезнуть, сделав его прозрачным, сохраняя на странице место, занимаемое элементом. Свойство `opacity` выбранного элемента изменяется от 1 до 0.

`.fadeOut(длительность, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта с помощью ключевых слов "fast", "normal", "slow" или числовых значений. Значение по умолчанию — 400 миллисекунд.

Функция по завершении анимации — необязательный параметр, задает функцию, которая будет вызвана после проявления элемента.

`.fadeOut(объект свойств)`

Объект свойств — представляет собой пары дополнительных опций, указанных в формате свойство: «значение».

`.fadeOut(длительность, функция перехода, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта.

Функция перехода — необязательный параметр, задает функцию перехода.

Функция по завершении анимации — необязательный параметр, задает функцию, которая будет вызвана после проявления элемента.

Метод `.fadeTo()` позволяет изменить степень прозрачности до заданного значения, например, `$("#element").fadeTo("normal",.50);`.

`.fadeTo(длительность, прозрачность, функция по завершении анимации)`

Длительность — задает скорость проявления эффекта.

Прозрачность — число от 0 до 1, задающее прозрачность элемента.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

`.fadeTo(длительность, прозрачность, функция перехода, функция по завершении анимации)`

Длительность — задает скорость проявления эффекта.

Прозрачность — число от 0 до 1, задающее прозрачность элемента.

Функция перехода — необязательный параметр, задает функцию перехода.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

Метод `.fadeToggle()` если элемент скрыт, то он появляется на экране, если виден, то исчезает.

`.fadeToggle(длительность, функция перехода, функция по завершении анимации)`

Длительность — задает скорость проявления эффекта.

Функция перехода — необязательный параметр, задает функцию перехода.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

`.fadeToggle(объект свойств)`

Объект свойств — представляет собой пары дополнительных опций, указанных в формате свойство: «значение».

Метод `.hide()` скрывает видимый элемент. При установлении скорости элемент исчезает, как бы сужаясь. Чтобы замедлить действие, нужно передать значение продолжительности.

`.hide()` метод указывается без параметров.

`.hide(длительность, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

`.hide(объект свойств)`

Объект свойств — представляет собой пары дополнительных опций, указанных в формате свойство: «значение».

`.hide(длительность, функция перехода, функция по завершении анимации)`

Длительность — задает скорость проявления эффекта.

Функция перехода — необязательный параметр, задает функцию перехода.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

Метод `.show()` показывает ранее скрытый элемент. Если не задано значение скорости, то элемент появляется моментально, если скорость задана, то элемент появляется от верхнего левого к нижнему левому углу.

`.show()` метод указывается без параметров.

`.show(длительность, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

`.show(объект свойств)`

Объект свойств — представляет собой пары дополнительных опций, указанных в формате свойство: «значение».

`.show(длительность, функция перехода, функция по завершении анимации)`

Длительность — задает скорость проявления эффекта.

Функция перехода — необязательный параметр, задает функцию перехода.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

Метод `.toggle()` одиночный метод, переключает выбранный элемент из одного состояния в другое, в зависимости от его текущего состояния, скрывая или отображая его.

`.toggle(длительность, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

`.toggle(объект свойств)`

Объект свойств — представляет собой пары дополнительных опций, указанных в формате свойство: «значение».

`.toggle(длительность, функция перехода, функция по завершении анимации)`

Длительность — задает скорость проявления эффекта.

Функция перехода — необязательный параметр, задает функцию перехода.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

`.toggle(display)`

Display — если параметр `display` установлен, то элемент будет показан.

Метод `.slideDown()` заставляет скрытый элемент появиться на веб-странице. Элемент проявляется постепенно — сначала его верхняя часть, и по мере проявления остальной части то, что находилось под элементом, сдвигается вниз. Поэтому для того, чтобы контент не перемещался по странице, можно использовать абсолютное позиционирование элемента `{position:absolute;}`. Если необходимо разместить данный элемент относительно другого, то задайте относительное позиционирование `{position:relative;}` для элемента, который окружает абсолютно позиционированный элемент.

`.slideDown(длительность, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

`.slideDown(объект свойств)`

Объект свойств — представляет собой пары дополнительных опций, указанных в формате свойство: «значение».

`.slideDown(длительность, функция перехода, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта.

Функция перехода — необязательный параметр, задает функцию перехода.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

Метод `.slideUp()` изменяет свойство `height` элемента, пока она не станет равной 0, после скрывает элемент `display: none;`. При этом удаление начинается снизу, и если для элемента не задано позиционирование, то контент, находившийся ниже, перемещается вверх.

`.slideUp(длительность, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

`.slideUp(объект свойств)`

Объект свойств — представляет собой пары дополнительных опций, указанных в формате свойство: «значение».

`.slideUp(длительность, функция перехода, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта.

Функция перехода — необязательный параметр, задает функцию перехода.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

Метод `.slideToggle()` скрывает видимый элемент и показывает скрытый, т.е. может использоваться в качестве переключателя, позволяющего как отображать, так и скрывать элемент. При этом элемент проявляется постепенно, сверху вниз.

`.slideToggle(длительность, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

`.slideToggle(объект свойств)`

Объект свойств — представляет собой пары дополнительных опций, указанных в формате свойство: «значение».

`.slideToggle(длительность, функция перехода, функция по завершении анимации)`

Длительность — необязательный параметр, задает скорость проявления эффекта.

Функция перехода — необязательный параметр, задает функцию перехода.

Функция по завершении анимации — необязательный параметр, описывает функцию, которая будет вызвана после проявления элемента.

Управление очередью анимации

Анимация элемента реализуется только с помощью методов, создающих анимационные эффекты. Если анимация реализована цепочкой методов, каждый эффект выполняется последовательно, друг за другом. Например, `$("#div").slideDown().fadeOut()`; сначала начнется скольжение блока вниз, а выцветание поставится в очередь "fx" и будет вызвано только при завершении скольжения.

Чтобы добавить пользовательские эффекты, создаются функции, которые также добавляются в очередь "fx". Очередь представляет массив функций, существующих на уровне элемента и хранящихся в `jQuery.data`. Каждый элемент может иметь одну или несколько очередей функций, но обычно используется только одна очередь по умолчанию "fx".

Функции включаются в очередь с помощью метода `.queue()`, без него функции выполняться не будут. Каждая функция по своему завершению должна вызывать метод `.dequeue()`, чтобы передать управление следующей функции в очереди.

```
$("#div").slideUp();
$("#div").queue(function() {
    document.write("Hellow");
    $(this).dequeue();
});
```

Метод `.queue()` метод показывает текущую очередь анимации или манипулирует очередью функций, создающих анимационные эффекты, для элемента/элементов набора.

`.queue(имяОчереди)`

ИмяОчереди — строка, содержащая имя очереди. По умолчанию текущая очередь имеет зарезервированное имя "fx". Возвращает массив функций, находящихся в очереди.

`.queue(имяОчереди, функция)`

ИмяОчереди — строка, содержащая имя очереди.

Функция — функция обратного вызова, которая добавляется в конец очереди с именемОчереди для всех элементов обернутого набора.

`.queue(имяОчереди, имяНовойОчереди)`

ИмяОчереди — строка, содержащая имя очереди.

ИмяНовойОчереди — заменяет текущую очередь всех элементов набора на новуюОчередь, состоящую из массива функций.

Метод `.dequeue()`. Ставит следующую анимацию в очередь. Это означает, что если выполняется анимация, когда этот метод вызывается, то новая анимация будет начата сразу после того, как текущая закончит свое выполнение. Выполняет первую функцию очереди для всех элементов набора, после выполнения функция удаляется из очереди. Обработчики события будут выполнять и удалять из очереди поочередно последующие функции. Если метод вызывается внутри функции, добавляемой в очередь, это вызовет поочередное исполнение всех функций, входящих в очередь.

`.dequeue(имяОчереди)`

ИмяОчереди — строка, содержащая имя очереди.

Метод `.clearQueue()` удаляет все функции анимации из очереди, не только текущую, но и все последующие, без их выполнения. Может использоваться для удаления любых очередей.

`.clearQueue(имяОчереди)`

ИмяОчереди — строка, содержащая имя очереди.

Метод `.delay()`

Позволяет установить отсрочку для запуска эффектов, ожидая определенное количество миллисекунд, прежде чем запустить следующий эффект в очереди.

`.delay(число или строка, имяОчереди)`

Число или строка — устанавливает продолжительность задержки в миллисекундах или с помощью ключевых слов — `fast` (эквивалентно `200ms`) и `slow` (эквивалентно `600ms`).

ИмяОчереди — необязательный параметр, указывает на очередь, для которой устанавливается задержка, по умолчанию — это очередь `"fx"`.

Метод `.stop()`

Останавливает текущую анимацию в очереди сразу после его запуска для элементов набора. Если при вызове метода передаются некоторые аргументы, то вы также можете очистить очередь и определить, должны ли элементы при остановке анимации остаться на месте, или они должны вернуться к первоначальному состоянию. По умолчанию оба параметра имеют значение `false`.

`.stop(логическое значение1, логическое значение2)`

Логическое значение1 — значение `true` останавливает все анимационные эффекты, находящиеся в очереди.

Логическое значение2 — значение `true` останавливает все эффекты, кроме текущего.

Метод `.finish()` останавливает текущую анимацию, удаляет все очереди анимации и завершает все анимации для соответствующих элементов.

`.finish(имяОчереди)`

ИмяОчереди — задает имя очереди анимации, которую нужно остановить. По умолчанию — очередь `"fx"`.

5. Порядок выполнения работы

1. Создайте несколько параграфов с элементами `strong` и `em`. Задайте обработчики событий наведения мышью на эти элементы: при наведении, цвет элементов должен отличаться от обычного, а размер плавно увеличиваться.

2. Создать метод у объекта `jQuery` — `parabola()`, который принимает аргументы `a`, `b` и `c` и анимирует элементы по параболе

3. При наведении на элемент с идентификатором `#box`, поворачивать элемент с классом `.some` на `90` градусов, при событии выведения мыши с элемента возвращать `.some` в исходную позицию

5. Назначить обработчики событий для кнопок, такие, чтобы после нажатия на кнопку элемент должен двигаться на `100px` в направлении, указанном кнопкой

6. При нажатии на любую точку страницы, перемещать с анимацией этот элемент к нужному месту.
7. Создать эффект переключателя с использованием библиотеки jQuery
8. Для изображений создать обработчики событий, чтобы работа скрипта была похожа на работу слайдера
9. Создать эффект лайтбокса средствами jQuery
10. Создать эффект карусели
11. Создать эффект галереи
12. Показать текущую очередь анимации, реализованную для слайдера
13. По нажатию на кнопку «Стоп» остановить выполнение всей анимации на странице.

6. Форма отчета о работе

Лабораторная работа № _____

Номер учебной группы _____

Фамилия, инициалы учащегося _____

Дата выполнения работы _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы: _____

7. Контрольные вопросы

1. Опишите способ создания собственных эффектов с помощью метода .animate().
2. Какие анимационные эффекты jQuery вы знаете?
3. Дайте характеристику метода .slideDown()
4. За что отвечает метод .dequeue()?

8. Рекомендуемая литература

1. **Макфарланд, Д.** Новая большая книга CSS / Дэвид Макфарланд. – СПб.: Питер, 2016. – 720с.
2. **Никсон, Р.** Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – 4-е изд. – СПб.: ООО «ПИТЕР М», 2017. – 768 с.
3. **Прохоренок, Н.А.** HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н.А.Прохоренок. – СПб.: БХВ-Петербург, 2010. – 912с.
4. **Робсон, Э.** Изучаем HTML, XHTML и CSS / Э. Робсон. – 2-е изд. – СПб.: ООО «ПИТЕР М», 2017. – 720 с.
5. **Фрейн, Б.** HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Бен Фрейн. – СПб.: Питер Пресс, 2017. — 272с.