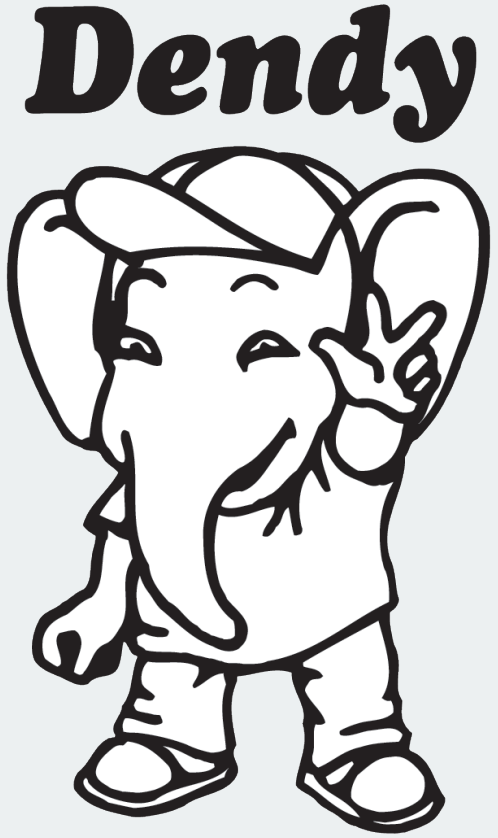
Урок №2

Manipulation



Преподаватель: Кирилл Версетти

Специальность: Backend Developer

Чем отличается атрибут от свойства?

.attr(attributeName [, value])

Возвращает/изменяет значение атрибута у элементов выборки.

.removeAttr(attributeName)

Удаляет указанный атрибут у элементов выборки.

.prop(propertyName [, value])

Возвращает/изменяет значение свойства у элементов выборки.

.removeProp(propertyName)

Удаляет заданное свойство у элементов выборки.

.val([value])

Возвращает/изменяет значение атрибута value у элементов формы.

Как работать с атрибутом class?

.addClass(className)

Добавляет класс(ы) к элементам выборки.

.removeClass([className])

Удаляет все или указанный(е) класс(ы) у элементов выборки.

.hasClass(className)

Возвращает true, если хотя бы один из элементов выборки имеет указанный класс.

.toggleClass(className [, state])

Добавляет или удаляет указанный(е) класс(ы) по принципу переключателя.

Добавляет указанный класс, если state установлен в true.

Удаляет указанный класс, если state установлен в false.

Как задавать значения CSS свойств?

.css(propertyName)

Возвращает значение CSS-свойства первого элемента выборки.

.css(propertyName, value)

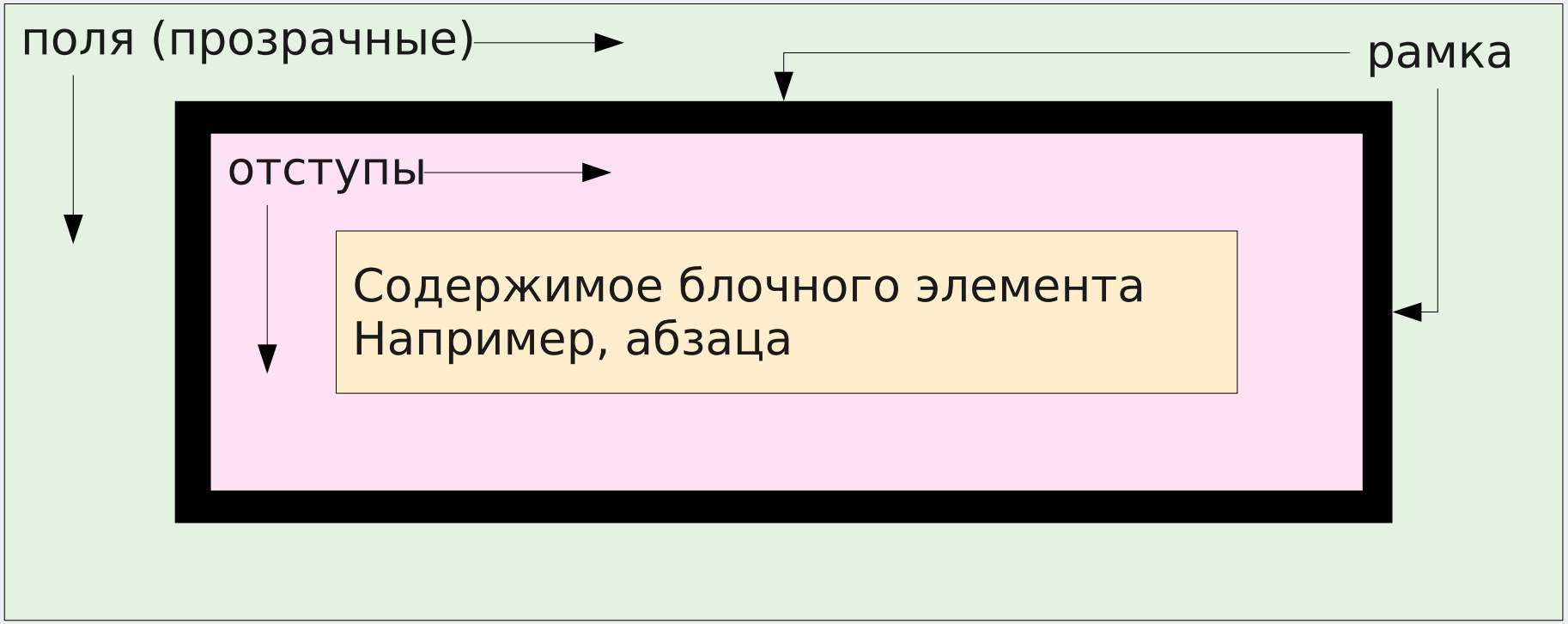
Устанавливает значение одного CSS-свойства для элементов выборки.

.css({propertyName1: value1, propertyName2: value2, ...})

Устанавливает CSS-свойства для всех элементов выборки, используя объект.

Далее вместо термина объект и строки вида
{propertyName1: value1, propertyName2: value2, ...}
будем использовать термин map

Как выглядит блок?



Основной частью каждого блока элемента является **область содержимого**.

Область содержимого окружена произвольным количеством **отступов (padding)**, **рамок (border)** и **полей (margin)**.

Чему равна ширина блока?

.width(value)

Устанавливает новое значение ширины для выбранных элементов.

1. **width()**

```
width = contentWidth
```

2. **innerWidth()**

```
width = contentWidth + paddingLeft + paddingRight
```

3. **outerWidth()**

```
width = contentWidth + paddingLeft + paddingRight
                     + borderLeft  + borderRight
```

4. **outerWidth(includeMargin)**

```
width = contentWidth + paddingLeft + paddingRight
                     + borderLeft  + borderRight
                     + marginLeft  + marginRight
```

Чему равна высота блока?

.height(value)

Устанавливает новое значение высоты для выбранных элементов.

1. **height()**

```
height = contentHeight
```

2. **innerHeight()**

```
height = contentHeight + paddingTop + paddingBottom
```

3. **outerHeight()**

```
height = contentHeight + paddingTop + paddingBottom
                       + borderTop  + borderBottom
```

4. **outerHeight(includeMargin)**

```
height = contentHeight + paddingTop + paddingBottom
                       + borderTop  + borderBottom
                       + marginTop  + marginBottom
```

Еще немного о стилях

.position()

Возвращает координаты выбранного элемента (top и left), относительно ближайшего родителя с типом позиционирования, отличным от static.

.offset([coordinates])

Возвращает координаты выбранного элемента (top и left), относительно документа (окна браузера).

Изменяет позицию элемента - coordinates - {top:newTop, left:newLeft}.

.offsetParent()

Возвращает ближайшего предка,

у которого задан тип позиционирования, отличный от static.

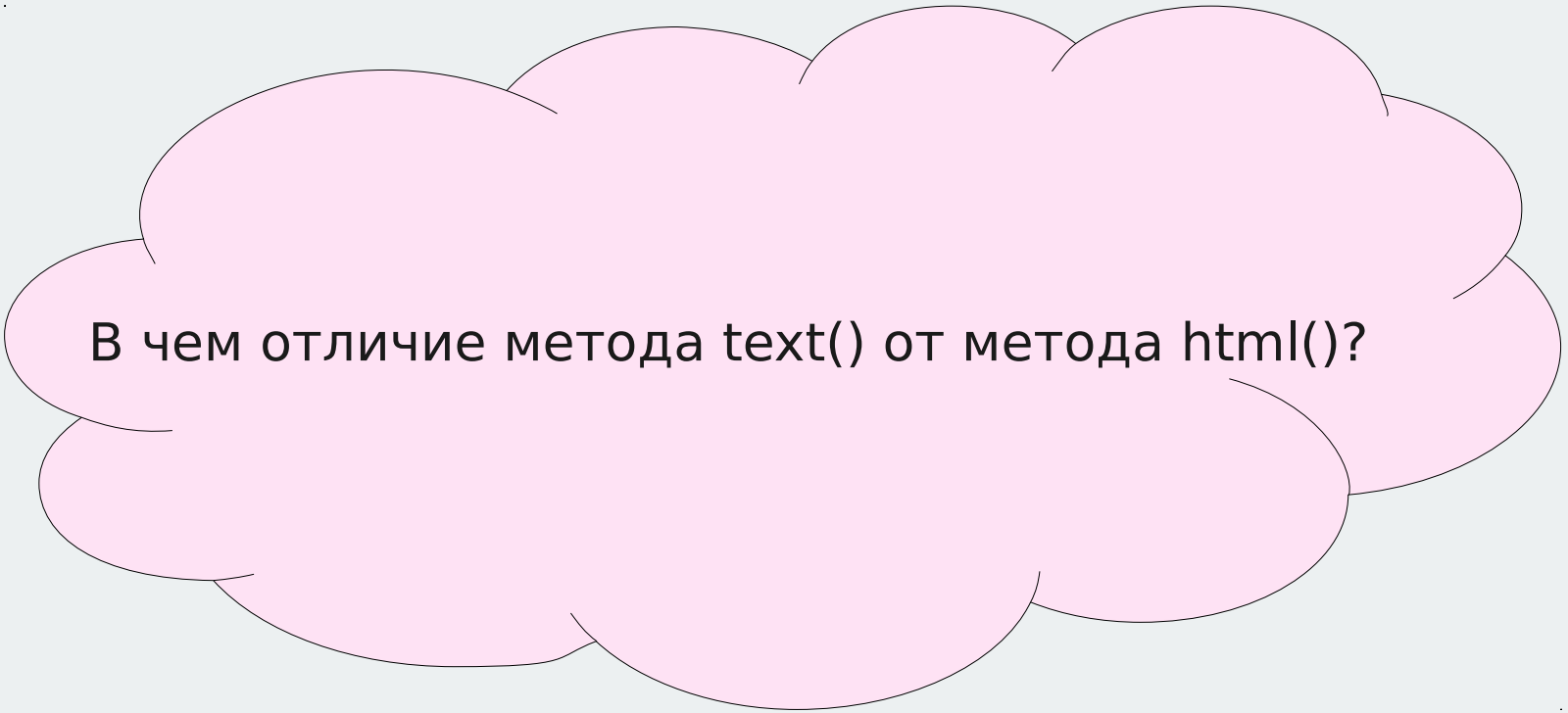
Как изменить содержимое элемента?

.text([textString])

Возвращает/изменяет текст, находящийся в элементах выборки.

.html([htmlString])

Возвращает/изменяет html-содержимое элементов выборки.



Куда вставить элементы DOM?

.before (content [, content])

```
<ul class="vertical-menu">
```

.prepend (content [, content])

```
    <li><a href="#about">Описание</a></li>
    <li><a href="#terms">Условия приема</a></li>
    <li><a href="#program">Программа</a></li>
    <li><a href="#timetable">Расписание</a></li>
```

.append (content [, content])

```
</ul>
```

.after (content [, content])

Как заменить элементы DOM?

.replaceWith(newContent)

Все элементы выборки будут заменены содержимым newContent.

Метод возвращает все удаленные (замененные) jQuery-элементы.

А не поменяться ли местами?

- `elements`[1]`.append(content`[2]`)` => `content`[3]`.appendTo(elements`[1]`)`
- `elements.prepend(content)` => `content.prependTo(elements)`
- `elements.after(content)` => `content.insertAfter(elements)`
- `elements.before(content)` => `content.insertBefore(elements)`
- `elements.replaceWith(content)` => `content.replaceAll(elements)`

1. elements – элементы, над которыми производятся действия
2. content может быть задан html-текстом, jQuery-объектом, DOM-объектом
3. content – новые добавляемые на страницу элементы

Как удалить элементы DOM?

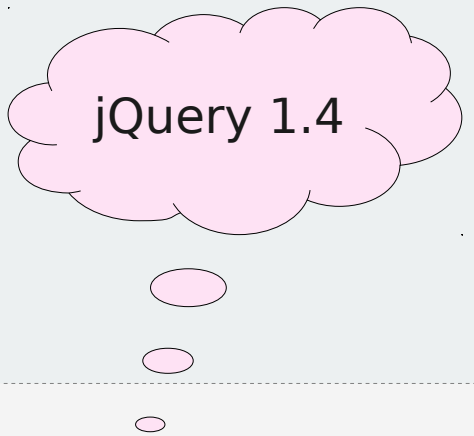
.empty()

Удаляет содержимое выбранных элементов.

.remove([selector])

Удаляет выбранные элементы.

selector - удаляются только те элементы, которые ему соответствуют.



.detach([selector])

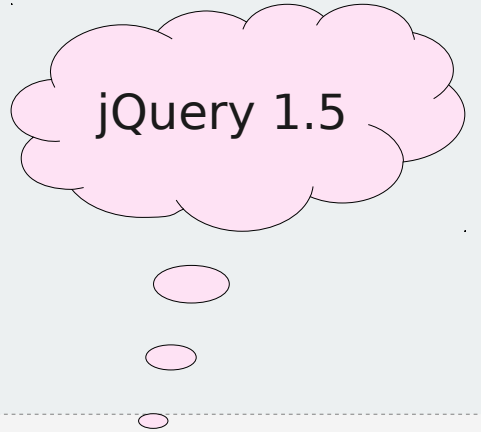
Метод работает аналогично .remove(), но не удаляет информацию об элементе и поэтому его можно восстановить.

Как клонировать элементы DOM?

.clone([withDataAndEvents])

Возвращает копию выбранных элементов страницы.

Если withDataAndEvents установлен в true, то будут скопированы все данные и обработчики событий. По-умолчанию false.



.clone([withDataAndEvents] [, deepWithDataAndEvents])

deepWithDataAndEvents указывает, нужно ли скопировать все данные и обработчики событий у дочерних элементов.

По-умолчанию равен первому аргументу.

Как «обернуть» элементы DOM?

.wrap(wrappingElement) и .wrapAll(wrappingElement)

Выбранные элементы будут "обернуты" wrappingElement[1].

.wrap() обертывает каждый элемент выборки.

.wrapAll() обертывает все выбранные элементы в один wrappingElement.

.wrapInner(wrappingElement)

Обертывает дочерние элементы каждого элемента из выборки.

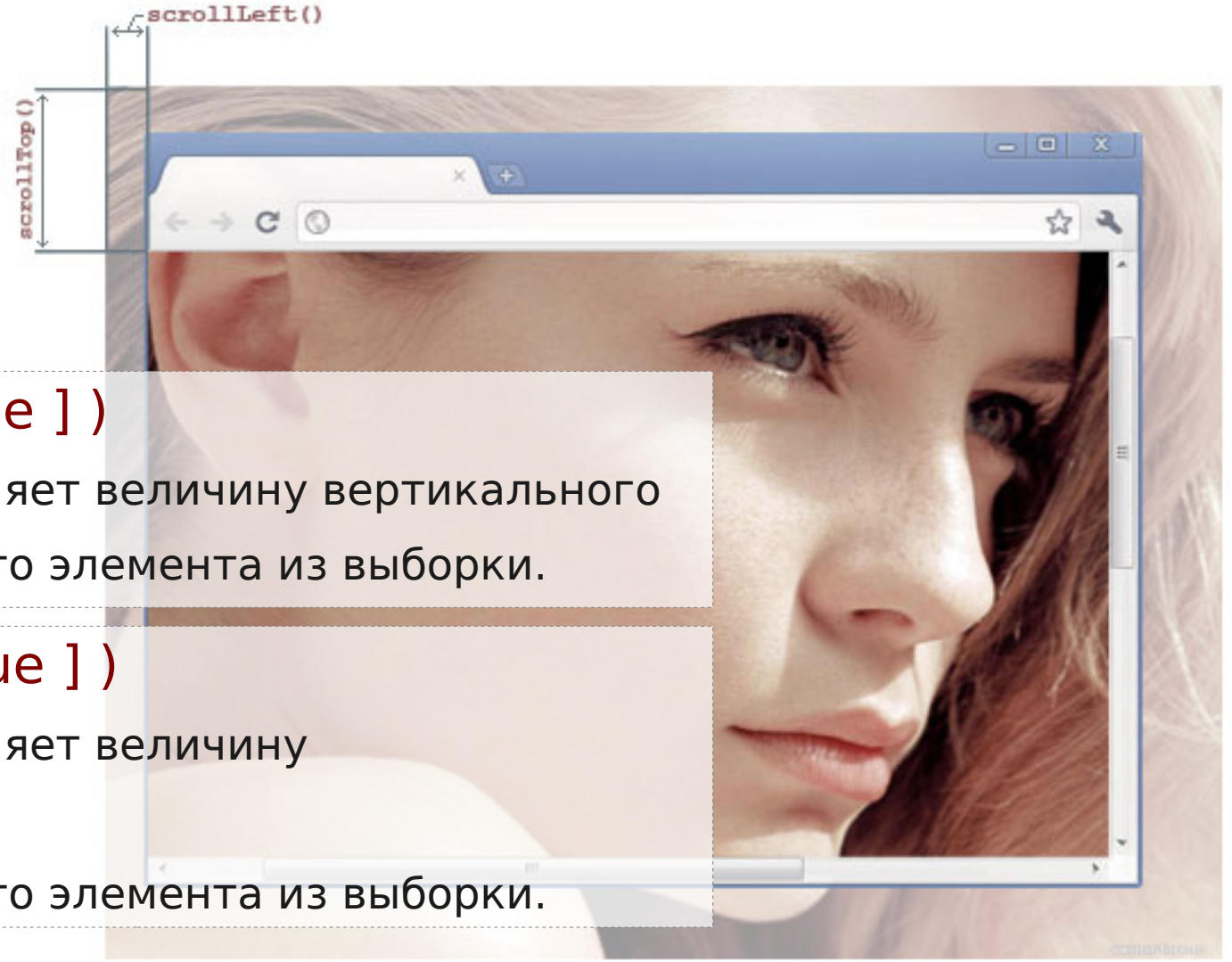
.unwrap()

Удаляет родительские элементы, при этом, их содержимое останется на своих местах.

---

1. wrappingElement может быть задан html-текстом, jQuery-объектом, DOM-объектом

Как прокручивать содержимое?



.scrollTop([value])

Возвращает/изменяет величину вертикального скроллинга первого элемента из выборки.

.scrollLeft([value])

Возвращает/изменяет величину горизонтального скроллинга первого элемента из выборки.


Effects
digitalblasphemy.com

Пара слов об аргументах

```
jQuery( exp ).show( [ duration ] [, callback ] );
```

duration - продолжительность выполнения анимации.

Значения аргумента:

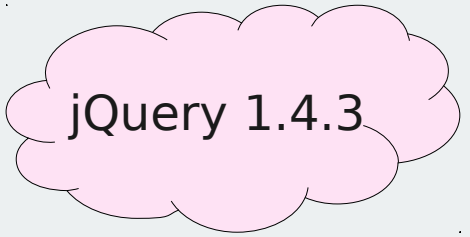
- миллисекунды (1000 миллисекунд - 1 секунда);
- fast (200 миллисекунд);
- slow (600 миллисекунд).

По-умолчанию 400 миллисекунд.

callback - функция-обработчик завершения анимации.

- не имеет аргументов;
- this будет содержать DOM-объект анимируемого элемента;
- обработчик вызывается для каждого элемента набора.

Пара слов об аргументах




```
jQuery(exp).show([duration] [, easing] [, callback])
```

easing - динамика выполнения анимации:

- замедление;
- ускорение;
- равномерно.

Задается с помощью функции. В стандартном jQuery таких две:

- linear - равномерная;
- swing - с ускорением (по умолчанию).

Другие варианты можно найти в плагинах.

Например, jQuery UI предоставляет более 30 новых динамик.

Какие эффекты есть в jQuery?

Базовые

- show([duration] [, easing] [, callback])
- hide([duration] [, easing] [, callback])
- toggle([duration] [, easing] [, callback] | showOrHide)

Появление/затухание

- fadeIn([duration] [, easing] [, callback])
- fadeOut([duration] [, easing] [, callback])
- fadeTo(duration, opacity [, easing] [, callback])
- fadeToggle([duration] [, easing] [, callback])

Скольжение

- slideDown([duration] [, easing] [, callback])
- slideUp([duration] [, easing] [, callback])
- slideToggle([duration] [, easing] [, callback])

Конец

"jQuery is not always the answer to all JavaScript problems.
There are many chases when simply no library is acceptable."