jQuery Elements & Filters

Elements

.before()
.after()
.prepend()
.append()
.remove()
.clone()
.unwrap()
.detach()
.empty()
.add()

Filters

.filter()

.not()

.has()

.is()

:contains()

Elements – inserting new and adding new content

.before()	inserts content before the selected element
.after()	inserts content after the selected element
.prepend()	inserts content inside the selected element after the opening tag
.append()	inserts content inside the selected element before the closing tag

Elements – inserting new and adding new content

Creating new elements

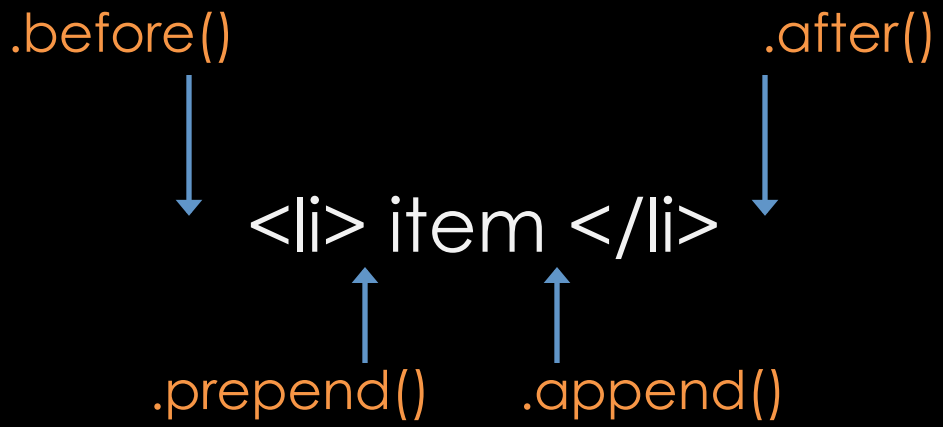
var $newFragment = $('<li>');

creates a variable newFragment to store a jQuery object which contains an empty <li> element

var $newItem = $('<li class="new">item ');

creates a variable newItem to store a jQuery object which contains an <li> element with a class and text

Elements – inserting new and adding new content



Elements – inserting new and adding new content

Example: adding-new-content.html

```
$(function() {
  $('ul').before('<p class="notice">Just updated</p>');
  $('li.hot').prepend('+ ');
  var $newListItem = $('<li><em>gluten-free</em> soy sauce</li>');
  $('li:last').after($newListItem);
});
```

Elements – cut, copy, paste

.remove()	removes matched elements from DOM tree (including any descendants and text nodes)
.detach()	same as .remove but keeps a copy of them in memory
.empty()	removed child nodes and descendants from any elements in matched set
.unwrap()	removes parents of matched set, leaving matched elements
.clone()	creates a copy of the matched set (including any descendants and text nodes)

Elements – cut, copy, paste

Example: cut-copy-paste.html

```
$(function() {
  var $p = $('p');
var $clonedQuote = $p.clone();
  $p.remove();
  $clonedQuote.insertAfter('h2');
  var $moveItem = $('#one').detach();
  $moveItem.appendTo('ul');
});
```

Elements – add and filter

.add	selects al elements that contain the text specified (parameter is case sensitive)
.filter()	finds elements in matched set that in turn match a second selector
.find()	finds descendants of elements in matched set that match the selector
.not()/:not()	finds elements that do not match the selector
.has()/:has()	finds elements from the matched set that have a descendant that matches the selector
:contains()	selects all elements that contain the text specified (parameter is case sensitive)

Elements – add and filter

Example: filters.html

```
$(function() {
  var $listItems = $('li');
  $listItems.filter('.hot:last').removeClass('hot');
  $('li:not(.hot)').addClass('cool');
  $listItems.has('em').addClass('complete');
  $listItems.each( function() {
    var $this = $(this);
    if ($this.is('.hot')) {
      $this.prepend('Priority item: ');
    }});
  $('li:contains("honey")').append(' (local)'); });
```

Elements- exercise

Exercise:

1. Create a simple unordered list with contains 5 items
2. Create a CSS class “priority” (text=em/bold) and apply it to all list items
3. Create a CSS class “basic” (text=smaller) Use .filter() method to find last list item with a class attribute “priority”
4. Remove class “priority” from the last list item
5. Use :not() method to find list item without class “priority”
6. Use .addClass() method to add class “regular” to that list item
7. Use .prepend() method to add < shape before each list item
8. Use .clone() method to copy the entire list and duplicate it after the first list (use <p> to separate the lists)

jQuery Attributes

Attributes

.attr()	gets/sets a specified attribute and its value
.removeAttr()	removes a specified attribute and its value
.addClass()	adds a new value to the existing value of the class attribute
.removeClass()	d=removes a value from a class attribute

Attributes

Example: attributes.html

```
$(function() {
  $('li#three').removeClass('hot');
  $('li.hot').addClass('favorite');
  $('ul').attr('id', 'group');
});
```

CSS properties

.css() gets and sets the values of CSS properties

```
var background = $ ('li').css('background-color');
```

```
$('li').css('background-color', '#272727');
```

sets the bg color of all list items

```
$('li').css('padding-left', '+=20');
```

Increases the value of left padding by 20

CSS properties

Example: css.html

```
$(function() {
var backgroundColor = $('li').css('background-color');
$('ul').append('<p>Color was: ' + backgroundColor + '</p>');
  $('li').css({
    'background-color': '#c5a996',
    'border': '1px solid #fff',
    'color': '#000',
    'text-shadow': 'none',
    'font-family': 'Georgia',
    'padding-left': '+=75'
  });
});
```

CSS properties

Exercise:

Using the previous exercise,
Use .css() methods to add

- Extra padding on the top of each list item
- Change typeface to “helvetica”
- Change the color of the text to white
- Change the background color to dark green

Loop

.each() allows to perform statements on each of the items in the selection of elements

this the current element

```
$('li').each(function() {
    var ids = this.id;
    $(this).append(' <span class="order">' + ids + '</span>');
 });
```

Loop

Example: each.html

```
$(function() {
   $('li').each(function() {
    var ids = this.id;
    $(this).append(' <span class="order">' + ids + '</span>');
  });
});
```

Loop

Exercise:

Continue in the previous exercise file,

Use .each() method to add text decoration = underline to each of the list items