

jQuery Fundamental



Bok, Jong Soon
javaexpert@nate.com
<https://github.com/swacademy/jquery>

Manipulation



html()

- *Get* the HTML contents of the first element in the set of matched elements.
- Return
 - **String**
- Syntax
 - **.html()**

```
var htmlString = $("div#intro").html();
```

html() (Cont.)

- *Set* the HTML contents of each element in the set of matched elements.
- Return
 - jQuery
- Syntax
 - `.html(htmlString)`

```
$ ("p#first") .html ("<strong>변경후</strong>") ;
```

text()

- *Get* the combined text contents of each element in the set of matched elements, including their descendants.
- Return
 - **String**
- Syntax
 - **.text()**

```
var textString = $("div#intro").text();
```

text() (Cont.)

- *Set* the content of each element in the set of matched elements to the specified text.
- Return
 - `jQuery`
- Syntax
 - `.text(text)`
 - `text` : `String` or `Number` or `Boolean`

```
$ ("p").text("Hello, World!");
```

prepend()

- *Insert* content, specified by the parameter, to the beginning of each element in the set of matched elements.
- Return
 - **jQuery**
- Syntax
 - **.prepend(content[, content])**
 - **content** : **htmlString** or **Element** or **Text** or **Array** or **jQuery**.

```
$ ("div") .prepend ("<p>반갑습니다.</p>") ;
```

prependTo()

- *Insert* every element in the set of matched elements to the beginning of the target.
- Return
 - **jQuery**
- Syntax
 - **.prepend(target)**
 - **target** : **Selector** or **htmlString** or **Element** or **Array** or **jQuery**.

```
$ ("strong") .prependTo ("p") ;
```


append()

- *Insert* content, specified by the parameter, to the end of each element in the set of matched elements.
- Return
 - **jQuery**
- Syntax
 - **.append(content [, content])**
 - **content** : **htmlString** or **Element** or **Text** or **Array** or **jQuery**.

```
$ ("div") .append ("<p>반갑습니다.</p>") ;
```

appendTo()

- *Insert* every element in the set of matched elements to the end of the target.
- Return
 - **jQuery**
- Syntax
 - **.appendTo(target)**
 - **target** : **Selector** or **htmlString** or **Element** or **Array** or **jQuery**.

```
$ ("strong") .appendTo ("p") ;
```

before()

- *Insert* content, specified by the parameter, before each element in the set of matched elements.
- Return
 - **jQuery**
- Syntax
 - **.before(content [, content])**
 - **content** : **htmlString** or **Element** or **Text** or **Array** or **jQuery**.

```
$ ("p#first").before("<h1> 태그 앞에 삽입 </h1>");
```

insertBefore()

- *Insert* every element in the set of matched elements before the target.
- Return
 - jQuery
- Syntax
 - `.insertBefore(target)`
 - `target` : `Selector` or `htmlString` or `Element` or `Array` or `jQuery`.

```
$("h1").insertBefore("p");
```

after()

- *Insert* content, specified by the parameter, after each element in the set of matched elements.
- Return
 - `jQuery`
- Syntax
 - `.after(content [, content])`
 - `content` : `htmlString` or `Element` or `Text` or `Array` or `jQuery`.

```
$ ("p#first").after("<h1> 태그 뒤에 삽입 </h1>");
```

InsertAfter()

- *Insert* every element in the set of matched elements after the target.
- Return
 - `jQuery`
- Syntax
 - `.insertAfter(target)`
 - `target` : `Selector` or `htmlString` or `Element` or `Array` or `jQuery`.

```
$ ("h1") .insertAfter ("p") ;
```

remove()

- *Remove* the set of matched elements from the DOM.
- Return
 - `jQuery`
- Syntax
 - `.remove([selector])`
 - `selector : String`

```
$("p strong").remove();
```

empty()

- *Remove* all child nodes of the set of matched elements from the DOM.
- Return
 - *jQuery*
- Syntax
 - *.remove()*

```
$("p:first").empty();
```


replaceWith()

- *Replace* each element in the set of matched elements with the provided new content and return the set of elements that was removed.
- Return
 - `jQuery`
- Syntax
 - `.replaceAll(newContent)`
 - `newContent` : `htmlString` or `Element` or `Array` or `jQuery`.

```
$ ("p") .replaceWith ("<h1> 변경후</h1>") ;
```

wrap()

- *Wrap* an HTML structure around each element in the set of matched elements.
- Return
 - `jQuery`
- Syntax
 - `.wrap(wrappingElement)`
 - `wrappingElement` : `Selector` or `htmlString` or `Element` or `jQuery`.

```
$ ("strong") .wrap ("<h1></h1>") ;
```

wrapAll()

- *Wrap* an HTML structure around all elements in the set of matched elements.
- Return
 - `jQuery`
- Syntax
 - `.wrapAll(wrappingElement)`
 - `wrappingElement` : `Selector` or `htmlString` or `Element` or `jQuery`.

```
$ ("strong") .wrapAll ("<h1></h1>") ;
```

wrapInner()

- *Wrap* an HTML structure around the content of each element in the set of matched elements.
- Return
 - `jQuery`
- Syntax
 - `.wrapAll(wrappingElement)`
 - `wrappingElement` : `Selector` or `htmlString` or `Element` or `jQuery`.

```
$ ("p") .wrapInner ("<strong></strong>") ;
```

Traversing



get()

- *Retrieve* one of the elements matched by the jQuery object.
- Return
 - **Element**
- Syntax
 - **.get(index)**
 - **index : Integer**

```
var str = $('#google').get(0).tagName;
```

children()

- *Get* the children of each element in the set of matched elements, optionally filtered by a selector.
- Return
 - `jQuery`
- Syntax
 - `.children([selector])`
 - `selector : Selector`

```
$("div#container").children("p").css("color", "blue");
```

siblings()

- *Get* the siblings of each element in the set of matched elements, optionally filtered by a selector.
- Return
 - `jQuery`
- Syntax
 - `.siblings([selector])`
 - `selector` : `Selector`

```
$("div#container").siblings().css("color", "blue");
```


find()

- *Get* the descendants of each element in the current set of matched elements, filtered by a selector, jQuery object, or element.
- Return
 - `jQuery`
- Syntax
 - `.find([selector])`
 - `selector : Selector`

```
$("div#container").find("p").css("color", "blue");
```

next()

- *Get* the immediately following sibling of each element in the set of matched elements.
- Return
 - `jQuery`
- Syntax
 - `.next([selector])`
 - `selector : Selector`

```
$("div#container").next().css("color", "blue");
```

prev()

- *Get* the immediately preceding sibling of each element in the set of matched elements.
- Return
 - **jQuery**
- Syntax
 - **.prev([selector])**
 - **selector : Selector**

```
$("div#container").prev().css("color", "blue");
```

slice()

- *Reduce* the set of matched elements to a subset specified by a range of indices.
- Return
 - `jQuery`
- Syntax
 - `.slice(start [, end])`
 - `start : Integer`
 - `end : Integer`

```
$('input').slice(1, 3).addClass("yellow");
```

filtering

■ `.first()`

- *Reduce* the set of matched elements to the first in the set.

■ `.last()`

- *Reduce* the set of matched elements to the final one in the set.

■ `.eq(index)`

- *Reduce* the set of matched elements to the one at the specified index.

■ `.filter(selector)`

- *Reduce* the set of matched elements to those that match the selector or pass the function's test.

■ `.not(selector)`

- *Remove* elements from the set of matched elements.

each()

- *Iterate* over a jQuery object, executing a function for each matched element.
- Return
 - `jQuery`
- Syntax
 - `.each(function)`

```
$('p').each(  
    function(index) {  
        alert($(this).html());  
    }  
);
```

Forms



:checked Selector

- *Matches* all elements that are *checked* or *selected*.
- Syntax
 - `jQuery (":checked")`

```
alert ( (':checked') .val() ) ;
```


:selected Selector

- *Selects* all elements that are *selected*.

- Syntax

 - `jQuery(":selected")`

```
var size = $(':selected').size();
```

:input Selector

- *Selects* all input, textarea, select and button elements.
- Syntax
 - `jQuery(":input")`

```
alert($(' :input' ).size() );
```

:text Selector

- *Selects* all input elements of type text.

- Syntax

- `jQuery(":text")`

```
$(":text").focus(function() {  
    $(this).after("<strong>필수 항목 입니다</strong>");  
});
```

:radio Selector

- *Selects* all elements of type radio.

- Syntax

- `jQuery(":radio")`

```
$(":radio").change(function() {  
  
});
```

:checkbox Selector

- *Selects* all elements of type checkbox.

- Syntax

 - `jQuery (":checkbox")`

```
$ (":checkbox") .attr ("checked" , "checked" ) ;
```

:submit Selector

- *Selects* all elements of type submit.
- Syntax
 - `jQuery(":submit")`

```
$(":submit").mouseover(function() {  
  
}
```

:reset Selector

- *Selects* all elements of type reset.

- Syntax

- `jQuery(":reset")`

```
$(":reset").one("click",function(){  
  
})
```

:button Selector

- *Selects* all button elements and elements of type button.
- Syntax
 - `jQuery(":button")`

```
$(":button").click(function() {  
  
}
```


:file Selector

- *Selects* all elements of type file.

- Syntax

- `jQuery(":file")`

```
$(":file").mouseover(function() {
```

```
}
```

:image Selector

- *Selects* all elements of type image.
- Syntax
 - `jQuery(":image")`

```
$(":image").mouseover(function() {  
  
}
```

focus()

- *Bind* an event handler to the "*focus*" JavaScript event, or trigger that event on an element.
- Return
 - **jQuery**
- Syntax
 - **.focus(handler)**
 - **handler : Function**

```
$ ("input").val ("입력해 주세요").css ("color", "#CCC")  
  .focus (function () {  
    $ (this).val ("").css ("color", "#000");  
  });
```

blur()

- *Bind* an event handler to the "*blur*" JavaScript event, or trigger that event on an element.
- Return
 - **jQuery**
- Syntax
 - **.blur(handler)**
 - **handler : Function**

```
$ ("input").val ("입력해 주세요").css ("color", "#CCC")  
.blur (function () {  
    if ($ (this).val () == "")  
        $ (this).val ("입력해 주세요").css ("color", "#000") ;  
});
```

change()

- *Bind* an event handler to the "*change*" JavaScript event, or trigger that event on an element.
- Return
 - **jQuery**
- Syntax
 - **.change(handler)**
 - **handler : Function**

```
$ ("select").change(function() {  
    $ ("span").text($ (this).val());  
});
```

submit()

- *Bind* an event handler to the "*submit*" JavaScript event, or trigger that event on an element.
- Return
 - **jQuery**
- Syntax
 - **.submit(handler)**
 - **handler : Function**

```
$ ("form") .submit (function () {  
  
}
```

Events



click()

- *Bind* an event handler to the "*click*" JavaScript event, or trigger that event on an element.
- Return
 - **jQuery**
- Syntax
 - **.click(handler)**
 - **handler : Function**

```
$ ("button") .click (function () {  
  
})) ;
```


dblclick()

- *Bind* an event handler to the "*dblclick*" JavaScript event, or trigger that event on an element.
- Return
 - *jQuery*
- Syntax
 - *.dblclick(handler)*
 - *handler : Function*

```
$("button").dblclick(function() {  
  
});
```

mousedown()

- *Bind* an event handler to the "*mousedown*" JavaScript event, or trigger that event on an element.
- Return
 - *jQuery*
- Syntax
 - *.mousedown(handler)*
 - *handler : Function*

```
$("a").mousedown(function() {  
  
});
```

mouseup()

- *Bind* an event handler to the "*mouseup*" JavaScript event, or trigger that event on an element.
- Return
 - *jQuery*
- Syntax
 - *.mouseup(handler)*
 - *handler : Function*

```
$ ("a") .mouseup (function () {  
  
})) ;
```

mouseover()

- *Bind* an event handler to the "*mouseover*" JavaScript event, or trigger that event on an element.
- Return
 - `jQuery`
- Syntax
 - `.mouseover(handler)`
 - `handler : Function`

```
$ ("img") .mouseover (function () {  
  
    }) ;
```

mousemove()

- *Bind* an event handler to the "*mousemove*" JavaScript event, or trigger that event on an element.
- Return
 - *jQuery*
- Syntax
 - *.mousemove(handler)*
 - *handler : Function*

```
$ ("img") .mousemove (function () {  
  
})) ;
```

one()

- *Attach* a handler to an event for the elements.
- The handler is executed at most *once* per element per event type.
- Return
 - `jQuery`
- Syntax
 - `.one(events [, data], handler)`
 - `events : String`
 - `data : PlainObject`
 - `handler : Function`

```
$("a").one("click", function() {  
  
});
```

hover()

- *Bind two* handlers to the matched elements, to be executed when the mouse pointer enters and leaves the elements.
- Return
 - `jQuery`
- Syntax
 - `.hover(handlerIn, handlerOut)`
 - `handlerIn : Function`
 - `handlerOut : Function`

```
$('table tr:gt(0)').hover(  
    function() { $(this).addClass('hover'); },  
    function() { $(this).removeClass('hover'); }  
);
```

bind()

- *Attach* a handler to an event for the elements.
- Return
 - `jQuery`
- Syntax
 - `.bind(eventType [, eventData], handler)`
 - `eventType : String`
 - `eventData : Anything`
 - `handler : Function`

```
$('#btnClick').bind('click', function() {  
    alert("Hello, World");  
});
```


unbind()

- *Remove* a previously-attached event handler from the elements.
- Return
 - **jQuery**
- Syntax
 - **.unbind(eventType [, handler])**
 - **eventType : String**
 - **handler : Function**

```
$ ("a") .unbind ("click") ;
```