

# JQUERY

JAVASCRIPT LIBRARY

0100011101010100011000110011  
1101011100010111001100110011  
00100111001110001100110011  
00111000000111001100110011  
10010010000111001100110011  
1001011001100111001100110011  
0101001001100011100110011001  
01011001100110001100110011  
11110110011001100110011001  
00010010011001100110011001  
01110100110011001100110011  
10000111010001100110011001  
0011010100011001100110011001  
01010101010101010101010101  
111101110010001100110011001  
010101100011001100110011001

---

# OUTLINE

- Understanding JQuery
- Download JQuery
- jQuery Syntax
- jQuery selectors
- jQuery Methods
  - DOM Manipulation
  - Attribute Manipulation
  - Traversing Elements
  - CSS Manipulation
  - Animation
- jQuery Events



# WHAT IS JQUERY?

- JQuery is a lightweight, “write less, do more”, JavaScript library
- Included in a single “.js” file
- An open-source library
- Provides many built-in functions to accomplish various tasks easily and quickly
- Provides excellent cross-browser support without writing extra code
- Easy to learn and master



# JQUERY IMPORTANT FEATURES

- DOM Selection
- DOM Manipulation
- Special Effects like fade-in, fade-out, sliding effects, animation etc.
- Event Handling
- AJAX
- Cross-browser support

# DOWNLOAD JQUERY

- Go to <https://www.jquery.com/download> page and download the latest version.

[Download the compressed, production jQuery 3.4.1](#)

[Download the uncompressed, development jQuery 3.4.1](#)

## REFERENCE JQUERY IN HTML PAGE

- You can add the JQuery file using `<script>` tag in your HTML file.

```
<head>  
<script src="/Scripts/jquery-3.4.1.js"></script>  
</head>
```

- Alternatively JQuery library can be included from public CDN

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">  
</script>
```

## USING JQUERY – JQUERY SYNTAX

- Global Function – **jQuery()**.
- Global scope means scope of window.
- \$ - alias of jQuery function
- Below all statements are same.

```
window.jQuery = window.$ = jQuery = $
```

- `$/jQuery()` is a selector function that selects DOM elements

```
jQuery('div') OR window.jQuery('div') OR $('div') OR window.$('div')
```

# THE DOCUMENT READY EVENT

```
$(document).ready(function(){  
    //jQuery methods go here  
})
```

- This is to prevent any jQuery code from running before the document is finished loading, i.e. DOM is loaded fully
- The jQuery ready() method is typically used with an anonymous function. So we can use shorthand notation as well

```
$(function(){  
    //jQuery methods go here  
})
```

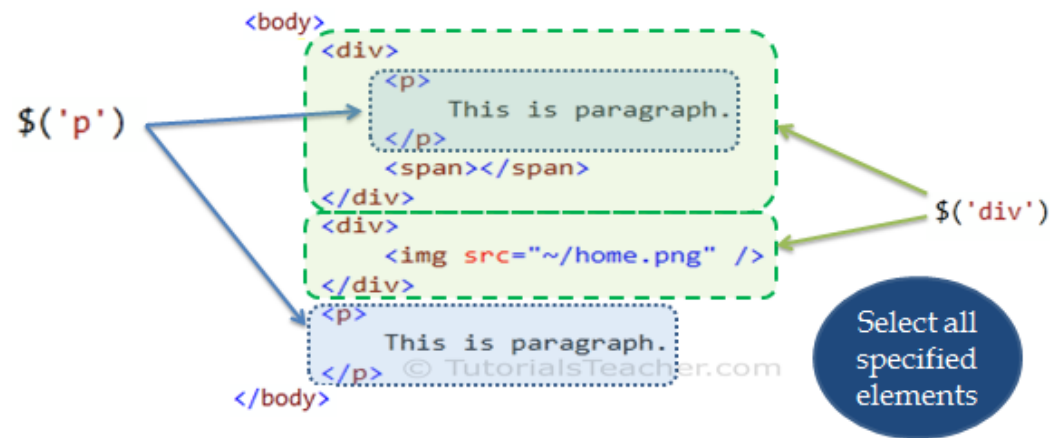


# JQUERY SELECTORS

- jQuery selectors allow you to select and manipulate HTML elements
- With jQuery selectors you can find elements based on their id, classes, types, attributes, values of attributes and much more.
- All types of selectors start with the dollar sign and parenthesis : \$()
- Types of jQuery selectors:
  - Element selectors
  - Id selectors
  - Class selectors

# ELEMENT SELECTOR

- The element selector selects element based on their tag names.
- Specifying an element name as string e.g. `$('div')` will return an array of all the `<div>` elements in the web page.
- The following figure shows which DOM element will be returned from `$('p')` and `$('div')`





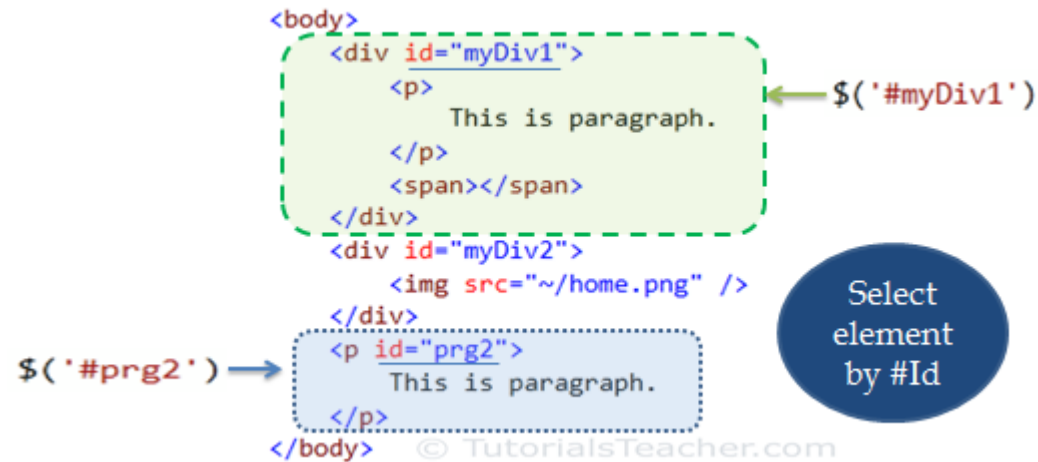
## ELEMENT SELECTOR

- Example of using Element selector. The following code will append text to all p elements.

```
$('p').append("This is a paragraph");
```

# ID(#) SELECTOR

- The #id selector uses the id attribute of HTML tag to find the specified element.
- The following figure shows which DOM elements will be returned from \$('#myDiv1') and \$('#prg2')



## EXAMPLE – ID SELECTOR

```
$(function(){  
  $('#impPrg').append('This element\'s id is "impPrg"');  
  
  $('#myDiv2').append('This element\'s id is "myDiv2"');  
});
```

## CLASS (.) SELECTOR

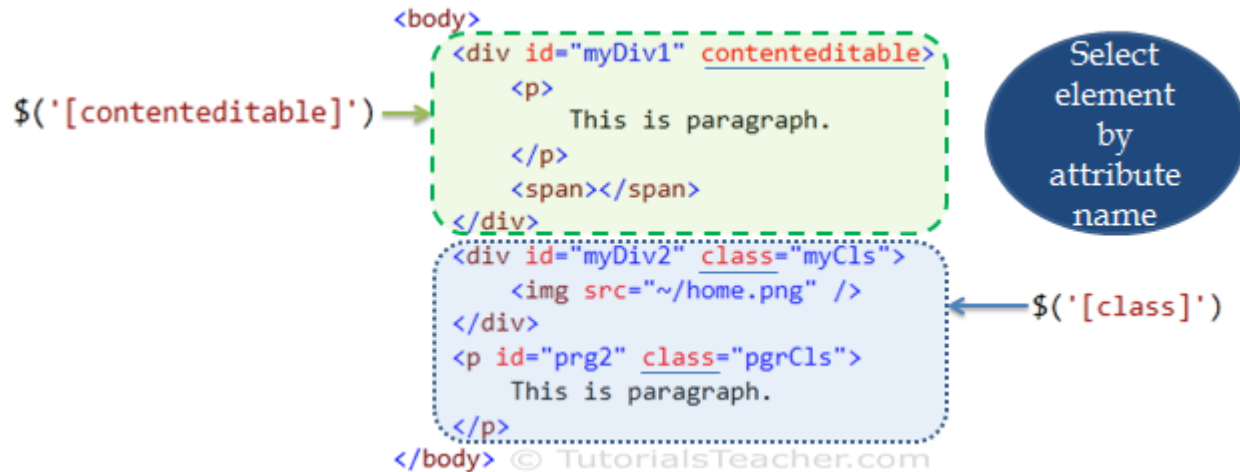
- The jQuery class selector finds elements with a specific class.
- Example:

```
<script>
$(function(){
$('.myclass').append('This element\'s class is "myclass"');
});
</script>

<div class="myclass" id="myDiv1">
</div>
```

# ATTRIBUTE ([ ]) SELECTOR

- jQuery allows to find an element based on attributes set on it.
- In the below example, jQuery returns all the elements that have class or contenteditable attribute irrespective of any value.



## EXAMPLE: ATTRIBUTE SELECTOR

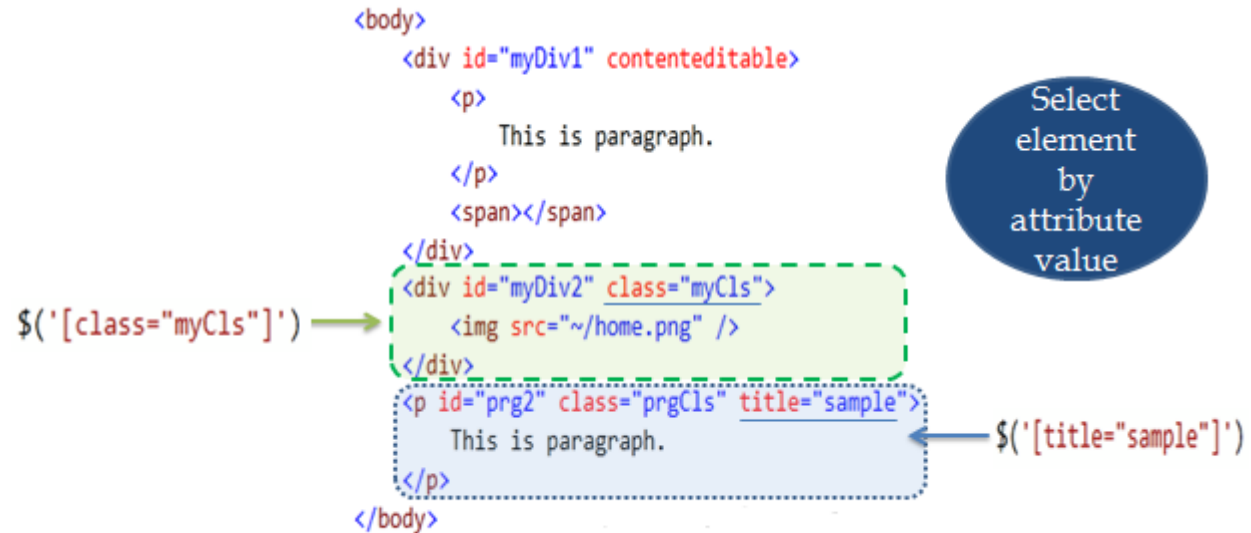
```
<script>
    $(document).ready(function () {
        $('[name]').val('My Text');
    });
</script>
```

```
<input type=password name=t2>
```

```
<input type=text name=t1 value="My Value">
```



# ATTRIBUTE VALUE SELECTOR



## EXAMPLE

```
<script>
    $(document).ready(function () {
        $('[name="t1"]').val('My Text');
    });
</script>
```

```
<input type=password name=t2>
```

```
<input type=text name=t1 value="My Value">
```

# SELECTOR PATTERNS

| Category                 | Selector               | Description   |
|--------------------------|------------------------|---|
| Find by Element          | \$('p')                | Find all <p> elements   |
|                          | \$('p, div, code')     | Find <p>, <div> and <code> elements   |
| Find Descendant Elements | \$('div p')            | Find all <p> elements which are descendants of <div>                                |
|                          | \$('div > p')          | Find <p> which is child of <div>  |
|                          | \$('*')                | Find all elements   |
| Find by Id               | \$('#myDiv')           | Find element whose id is <i>myDiv</i>   |
|                          | \$( <i>p</i> myPrg')   | Find <p> element whose Id is <i>myPrg</i>   |
|                          | \$('#myDiv1, #myDiv2') | Find multiple elements by id separated by comma.                                    |
| Find by Attributes       | \$('[class]')          | Find all the elements with the <i>class</i> attribute(whatever the value).          |
|                          | \$('div[class]')       | Find all the <div> elements that have a <i>class</i> attribute(whatever the value). |

# SELECTOR PATTERNS

| Category           | Selector                                      | Description  |
|--------------------|---|--|
| Find by CSS Class  | <code>\$('.myCSSClass')</code>                | Find all the elements with <i>class=myCSSClass</i> .   |
|                    | <code>\$('.myCSSClass1, .myCSSClass2')</code> | Finds all elements whose class attribute is set to <i>myCSSClass1</i> or <i>myCSSClass2</i>                                |
|                    | <code>\$('div.myCSSClass')</code>             | Finds all <code>&lt;div&gt;</code> elements with <i>class=myCSSClass</i>   |
|                    | <code>\$('p:first-child')</code>              | Find all <code>&lt;p&gt;</code> elements, which is the first child of its parent element. (parent element can be anything) |
| Find by Input type | <code>\$(":button")</code>                    | Find all input whose type is button.   |
|                    | <code>\$(':input[type="radio"]')</code>       | Find all radio input types.  |
| Even-Odd rows      | <code>\$('tr:odd')</code>                     | Find all odd rows. (1,3,5,7..)   |
|                    | <code>\$('tr:even')</code>                    | Find all even rows.(0,2,4,6..)   |

# SELECTOR PATTERNS

| Category                              | Selector                                  | Description   |
|---------------------------------------|---|---|
| Find by containing value of attribute | <code>\$('div[class=myCls]')</code>       | Find all the <code>&lt;div&gt;</code> elements whose class attributes are equal to <i>myCls</i> .   |
|                                       | <code>\$('div[class =myCls]')</code>      | Find all the <code>&lt;div&gt;</code> elements whose class attributes are either equal to <i>myCls</i> or starting with <i>myCls</i> string followed by a hyphen (-). |
|                                       | <code>\$('div[class *= "myCls"]')</code>  | Selects <code>&lt;div&gt;</code> elements whose class attributes contain <i>myCls</i> .   |
|                                       | <code>\$('div[class~=myCls]')</code>      | Selects <code>div</code> elements whose class attributes contain <i>myCls</i> , delimited by spaces.  |
|                                       | <code>\$('div[class\$=myCls]')</code>     | Selects <code>&lt;div&gt;</code> elements whose class attributes value ends with <i>myCls</i> . The comparison is case sensitive.                                     |
|                                       | <code>\$('div[class!=myCls]')</code>      | Selects <code>&lt;div&gt;</code> elements which does not have <i>class</i> attribute or value does not equal to <i>myCls</i> .  |
|                                       | <code>\$('div[class^=myCls]')</code>      | Selects <code>&lt;div&gt;</code> elements whose <i>class</i> attribute value starts with <i>myCls</i> .   |
|                                       | <code>\$('div:contains("mydata")')</code> | Find all <code>&lt;div&gt;</code> elements that contains the text <i>'mydata'</i>   |



# JQUERY METHODS

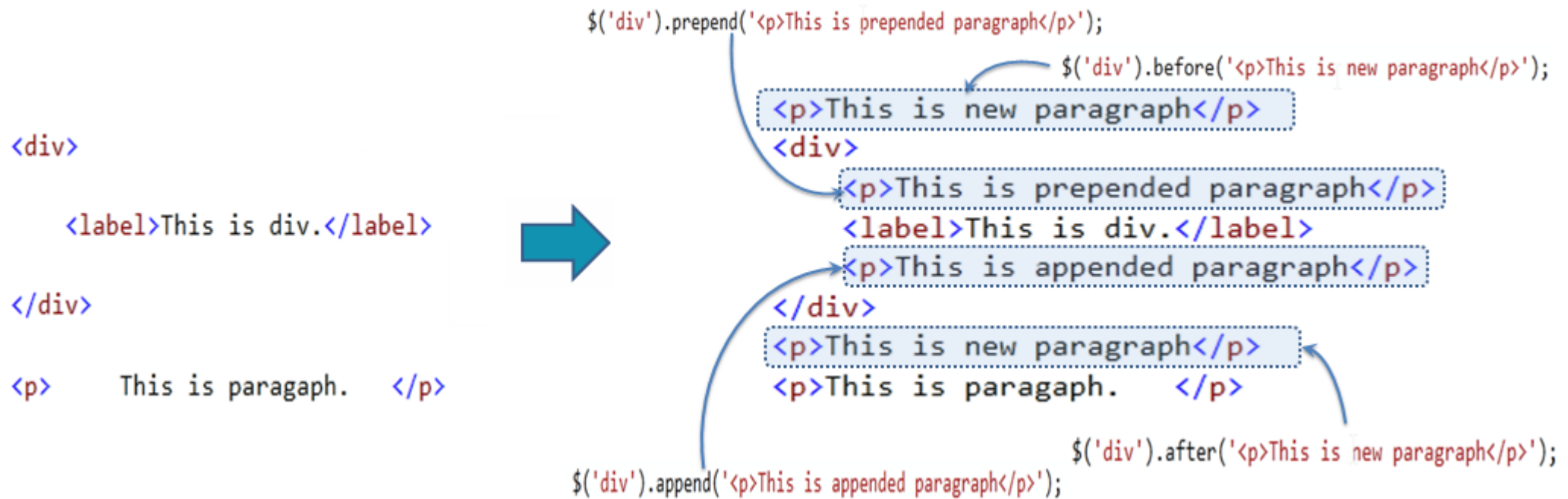




# JQUERY METHODS

- jQuery provides various methods for different tasks such as
  - Manipulate DOM
  - Events
  - AJAX etc.

# DOM MANIPULATION





# DOM MANIPULATION

| jQuery Method | Description   |
|---------------|---|
| append()      | Inserts content to the end of element(s) which is specified by a selector.                            |
| before()      | Inserts content (new or existing DOM elements) before an element(s) which is specified by a selector. |
| after()       | Inserts content (new or existing DOM elements) after an element(s) which is specified by a selector.  |
| prepend()     | Insert content at the beginning of an element(s) specified by a selector.                             |
| remove()      | Removes element(s) from DOM which is specified by selector.   |

# MANIPULATE ATTRIBUTES

```
$('#myDiv').attr('class')  $('#myDiv').prop('class')  
  
<div id="myDiv" class="divCls">  
  <p style="background-color:yellow;width:100%">  
    This is paragraph.  
  </p>  
</div>  
<div id="firstNameDiv">  
  <label>First Name</label><input type="text" value="John" />  
</div>  
<input type="button" value="Get Value" id="addBtn" style="width:100px" />
```

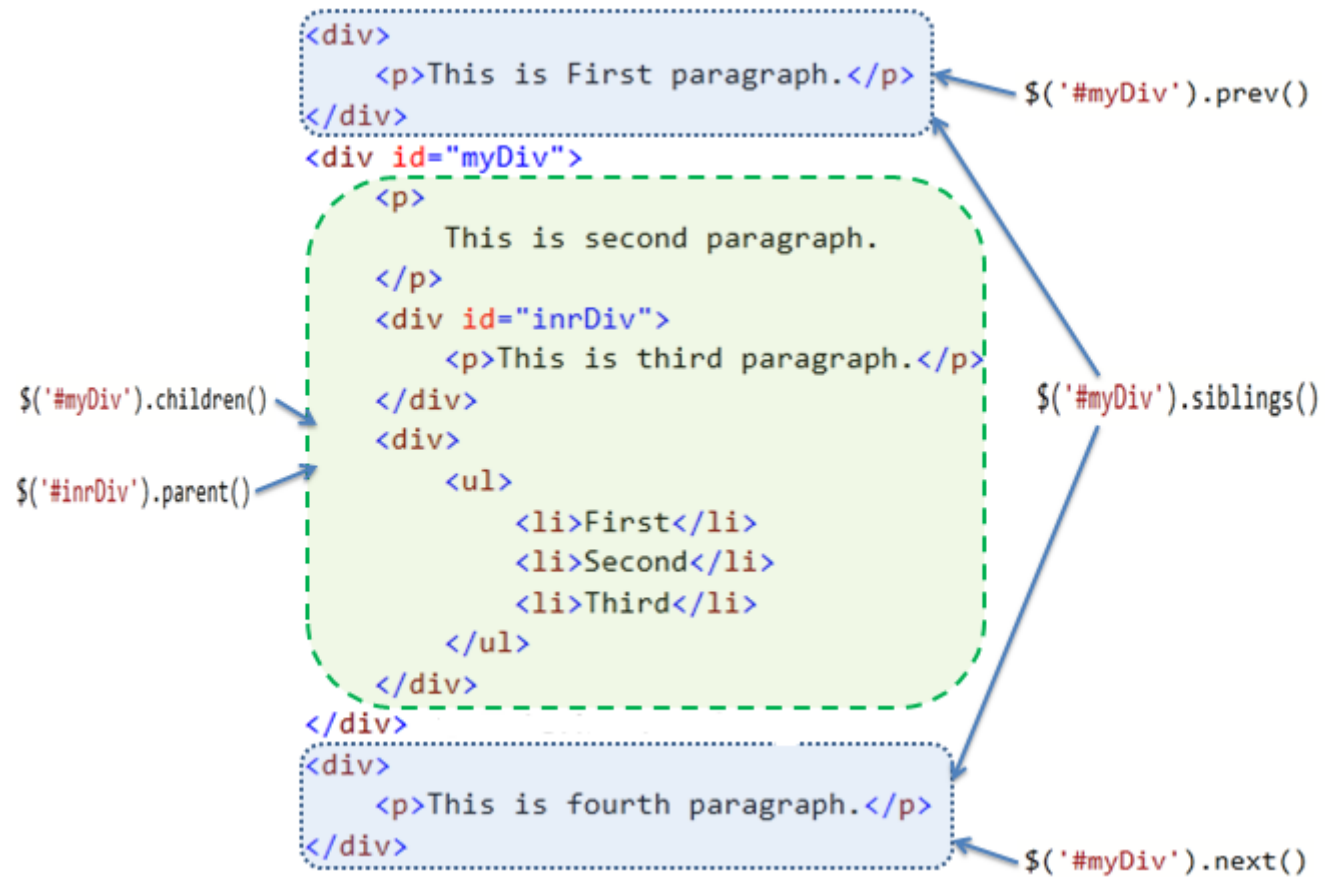
Diagram illustrating jQuery attribute manipulation methods and their corresponding HTML elements:

- `$('#myDiv').attr('class')` and `$('#myDiv').prop('class')` point to the `class="divCls"` attribute in the `<div>` tag.
- `$('#myDiv').html()` points to the entire content of the `<div id="myDiv">` element.
- `$('#myDiv').text()` points to the text content "This is paragraph." inside the `<p>` tag.
- `$('#input:text').val()` points to the `value="John"` attribute of the text input.
- `$('#label').text()` points to the text "First Name" inside the `<label>` tag.
- `$('#input:button').val()` points to the `value="Get Value"` attribute of the button input.
- `$('#input:button').prop('style').width` points to the `width:100px` style attribute of the button input.

# MANIPULATE ATTRIBUTES

| jQuery Method | Description  |
|---------------|--|
| attr()        | Get or set the value of specified attribute                |
| prop()        | Get or set value of specified property                     |
| html()        | Get or set html content to the specified target element    |
| text()        | Get or set text for the specified target element           |
| val()         | Get or set value property of the specified target element. |

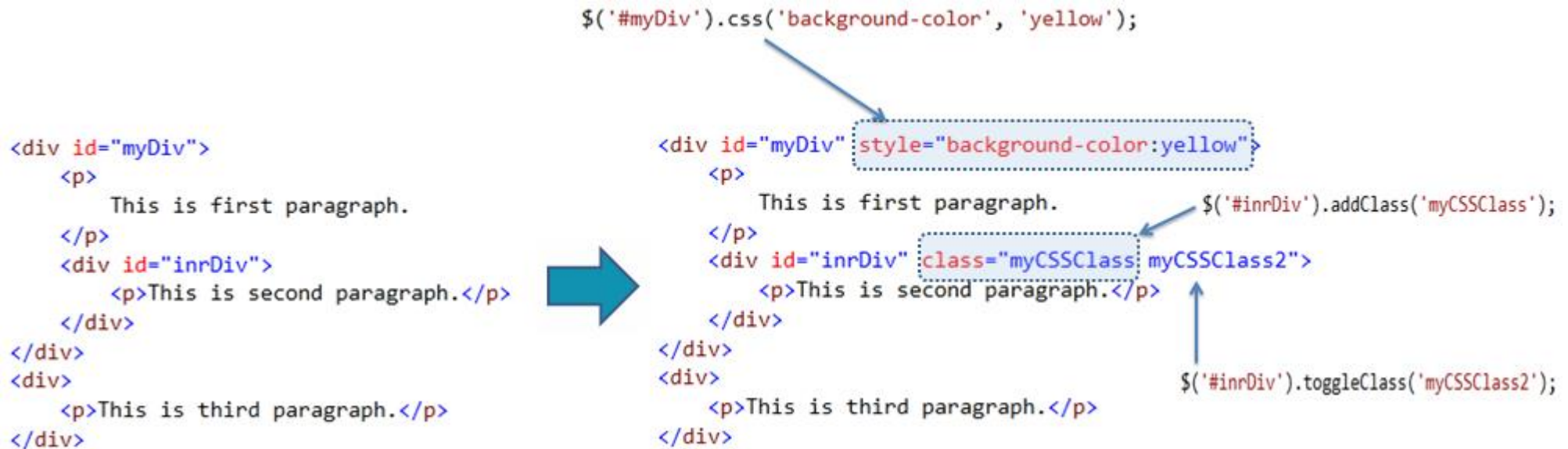
# TRAVERSING ELEMENTS



# TRAVERSING DOM ELEMENTS

| jQuery Method | Description   |
|---------------|---|
| children()    | Get all the child elements of the specified element                                       |
| each()        | Iterate over specified elements and execute specified call back function for each element |
| find()        | Get all the specified child elements of each specified element                            |
| first()       | Get the first occurrence of the specified element   |
| next()        | Get the immediately following sibling of the specified element                            |
| parent()      | Get the parent of the specified element(s)  |
| prev()        | Get the immediately preceding sibling of the specified element.                           |
| siblings()    | Get the siblings of each specified element(s)   |

# CSS MANIPULATION



# CSS MANIPULATION

| jQuery Method | Description  |
|---------------|--|
| css()         | Get or set style properties to the specified element(s).                               |
| addClass()    | Add one or more class to the specified element(s).                                     |
| hasClass()    | Determine whether any of the specified elements are assigned the given CSS class.      |
| removeClass() | Remove a single class, multiple classes, or all classes from the specified element(s). |
| toggleClass() | Toggles between adding/removing classes to the specified elements                      |

# JQUERY ANIMATION

| jQuery Method | Description  |
|---------------|--|
| animate()     | Perform custom animation using element's style properties.                         |
| queue()       | Show or manipulate the queue of functions to be executed on the specified element. |
| stop()        | Stop currently running animations on the specified element(s).                     |
| fadeIn()      | Display specified element(s) by fading them to opaque.                             |
| fadeOut()     | Hides specified element(s) by fading them to transparent.                          |
| fadeTo()      | Adjust the opacity of the specified element(s)                                     |
| fadeToggle()  | Display or hide the specified element(s) by animating their opacity.               |
| hide()        | Hide specified element(s).   |



# JQUERY ANIMATION

| jQuery Method | Description   |
|---------------|---|
| show()        | Display specified element(s).                             |
| toggle()      | Display hidden element(s) or hide visible element(s).     |
| slideUp()     | Hide specified element(s) with sliding up motion.         |
| slideDown()   | Display specified element(s) with sliding down motion.    |
| slideToggle() | Display or hide specified element(s) with sliding motion. |



# JQUERY - EVENTS



# EVENT HANDLING

- Use selector to get the reference of DOM element and invoke appropriate event method

```
<script>
$( '#saveBtn' ).click(function () {
    alert('Save button clicked');
});
</script>
```

```
<input type="button" value="Save" id="saveBtn" />
```

## EVENT BINDING USING 'ON'

- jQuery allows to attach an event handler for one or more events to the selected element using **on** method
- The on() method gives you more flexibility in event binding

- Syntax

on(types, selector, data, fn )

- Types = One or more space-separated event types
- Selector = selector string
- Data = data to be passed to the handler in event
- Fn = a function to execute when event is triggered

## EXAMPLE

```
<script>
    $('#saveBtn').on('click',function () {
        alert('Save Button clicked');
    });
</script>

<input type="button" value="Save" id="saveBtn" />
```

## EXAMPLE: EVENT BINDING USING ON & SELECTORS

```
<script>
  $(function(){
    $('#myDiv').on('click', ':button', function () {
      alert('Button clicked');
    });
  });
</script>

<div id="myDiv" >
  <input type="button" value="Save" id="saveBtn" />
  <input type="button" value="Add" id="addBtn" />
</div>

<input type="button" value="Delete" id="delBtn" />
```

- The selector is :button
- Click event triggered by buttons in <div> tag whose id is 'myDiv', will only be handled.