STEP
IT ACADEMY

Chapter 11

Introduction to jQuery

Lecturer Mr. Sok Dina

Content

1. Using jQuery

    1.1. What Is jQuery?

    1.2. Adding jQuery

2. Accessing Page Elements Using the $ Function

3. The Concept of a Selector

    3.1. CSS Selectors

    3.2. Form Selectors

    3.3. Features of Selectors in jQuery

4. Methods Managing jQuery Styles

    4.1. Traversing. DOM Traversing Methods

    4.2. Event Handling in jQuery

    4.3. Assign One Handler for Multiple Events

    4.4. Animation Methods in jQuery

5. Delegating Events

    5.1. Canceling Event Handlers

6. jQuery and AJAX

STEP IT ACADEMY

1.1. What Is jQuery?

- jQuery: a JavaScript library written in JavaScript to manipulate DOM elements, handle events, create animation, and load or get AJAX data.
- The developer of jQuery is John Resig who published the first version of jQuery in 2006.
- jQuery:

  + Access to elements using selectors and various filters.

  + Cross-browser, meaning it worked equally among browsers

  + Create animation effects

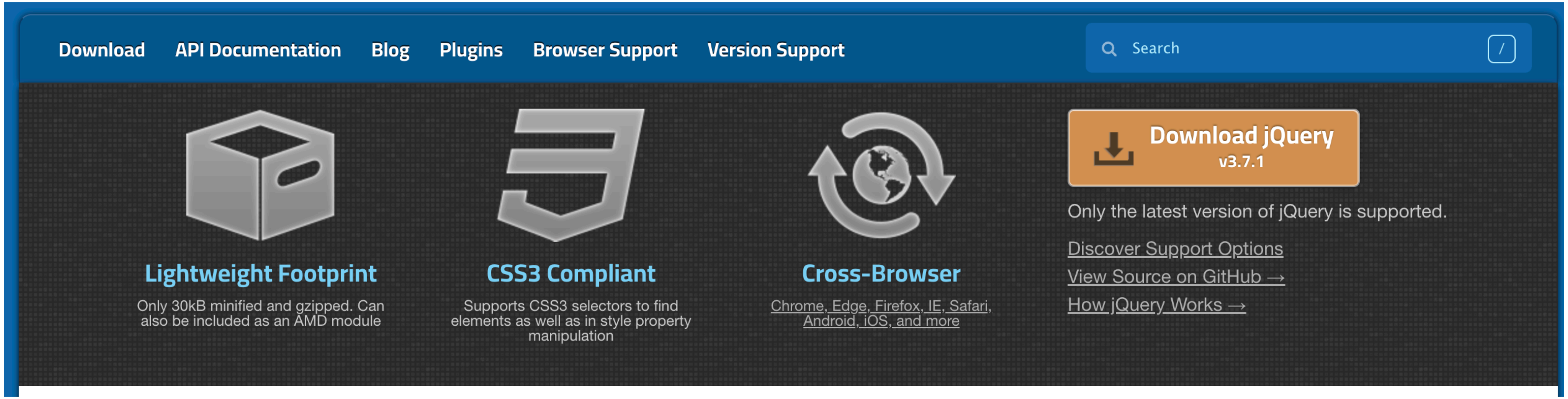
1.2. Adding jQuery

- Go website at https://jquery.com/.
- Click on the Download jQuery link that let you download in several options:

+ **Compressed, production jQuery**: a compressed, minified file is used for website pages.

+ **Uncompressed, development jQuery** is an uncompressed file for plugin developers with comments to functions

+ **Slim build** means a minified version of jQuery without all the animation methods and AJAX methods.



Lightweight Footprint

Only 30kB minified and gzipped. Can also be included as an AMD module

CSS3 Compliant

Supports CSS3 selectors to find elements as well as in style property manipulation

Cross-Browser

Chrome, Edge, Firefox, IE, Safari, Android, iOS, and more

Only the latest version of jQuery is supported.

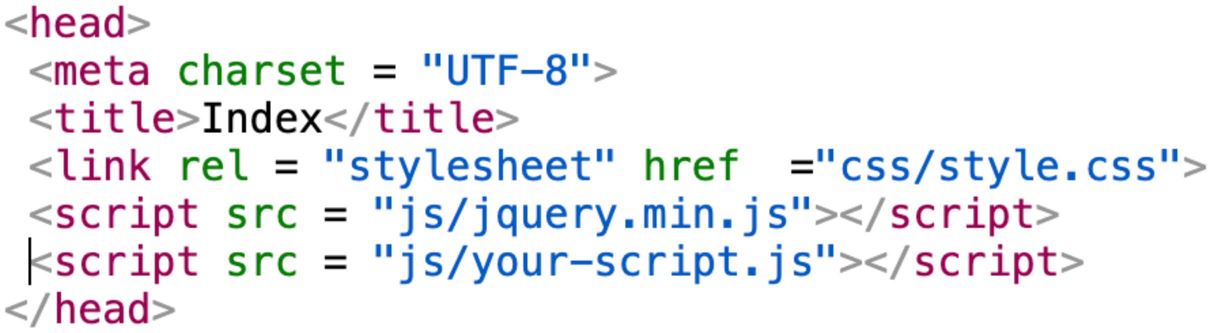
Discover Support Options

View Source on GitHub →

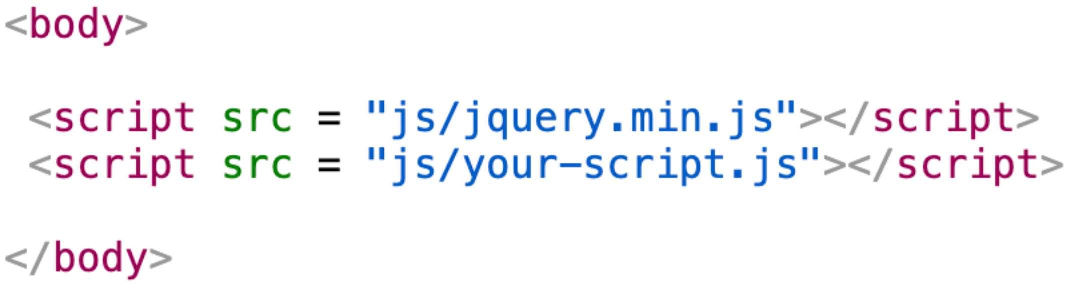
How jQuery Works →

1.2. Adding jQuery

- Download the “Download the compressed, production jQuery 3.7.1”
- Click on the link, press CTRL+S, and save the file to the project folder
- To add jQuery to the project using “src” attribute of the <script> tag in the <head> or <body> block:

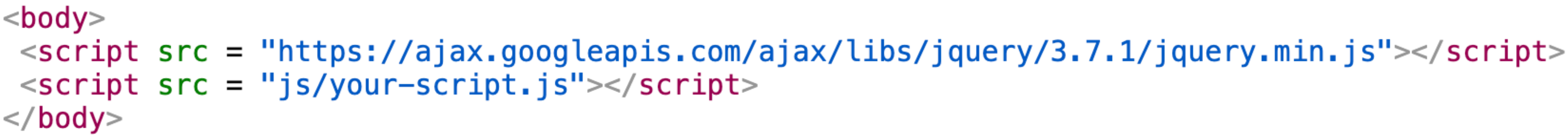


```
<head>
 <meta charset = "UTF-8">
 <title>Index</title>
 <link rel = "stylesheet" href  ="css/style.css">
 <script src = "js/jquery.min.js"></script>
 <script src = "js/your-script.js"></script>
</head>
```



```
<body>

 <script src = "js/jquery.min.js"></script>
 <script src = "js/your-script.js"></script>

</body>
```

- In this case, adding jQuery without download by using Google CDN:



```
<body>
 <script src = "https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
 <script src = "js/your-script.js"></script>
</body>
```

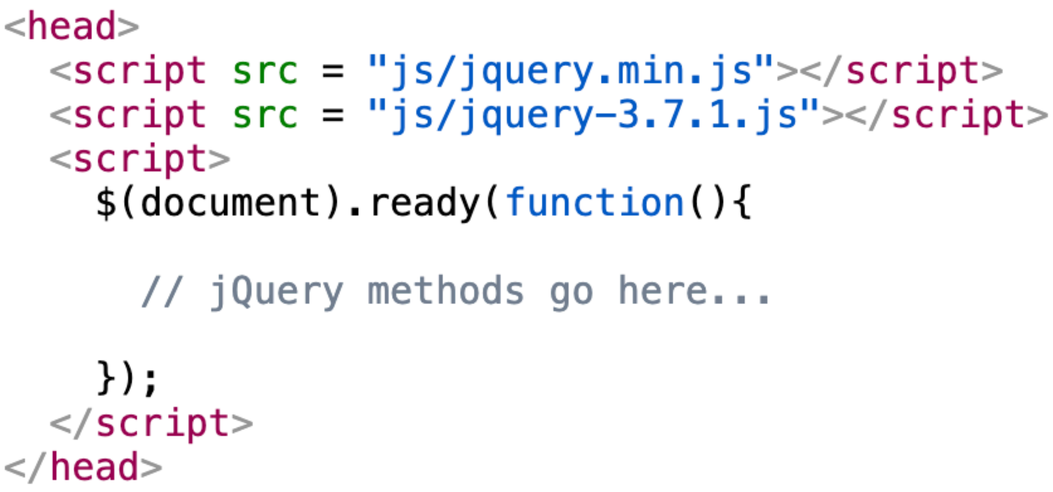
2. Accessing Page Elements Using the $ Function

- In jQuery, any object must be referenced using a “jQuery” keyword or “$”.
- Syntax: **$(*selector*).*action*(**function(){ // your code }**)**

  jQuery(selector).action(function(){ // your code });

The jQuery syntax is for **selecting** HTML elements and performing some **action** on the element(s).

- “Ready” Event: prevent any jQuery code from running before the document is finished loading (is ready).

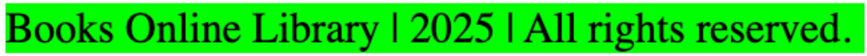


```
<head>
  <script src = "js/jquery.min.js"></script>
  <script src = "js/jquery-3.7.1.js"></script>
  <script>
    $(document).ready(function(){

      // jQuery methods go here...

    });
  </script>
</head>
```

3.1. CSS Selectors

- A selector in jQuery is a string that specifies criteria for searching DOM elements on a page.
- Syntax: **jQuery(selector) or.** $('selector')

```
<!DOCTYPE html><html>
<head>
  <style> .intro{ background-color: yellow;  } </style>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  <script>
    $(document).ready(function() {
               $('#datenow').text(new Date().getFullYear());
               $(".intro").css("background","lime");
    });
  </script>
</head>
<body>
  <p class="intro">Books Online Library | <span id ="datenow"></span> | All rights reserved.</p>
</body>
</html>
```



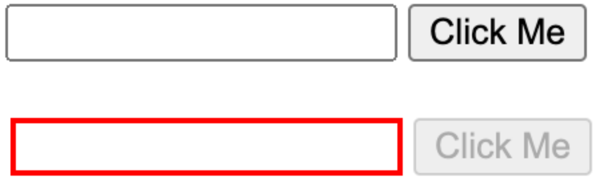
Books Online Library | 2025 | All rights reserved.

- $('#date-now') access to the element id="date-now" attribute
- The $(".intro") selector access elements of the .intro class

3.2. Form Selectors

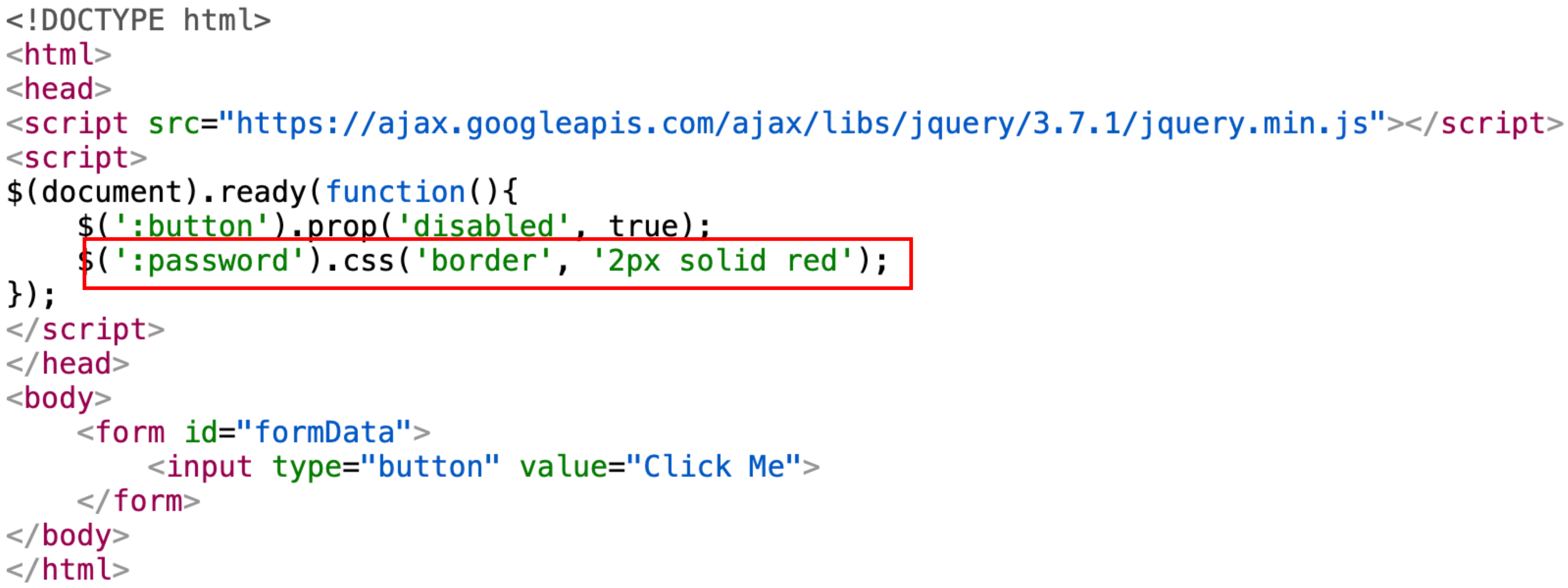
- Selectors: control form elements like input fields, radio buttons, checkboxes, dropdowns, and buttons.
  Syntax: **$(‘:selector’) or.** $(’:selector')

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $(':button').prop('disabled', true);
    $(':password').css('border', '2px solid red');
});
</script>
</head>
<body>
    <form id="formData">
        <input type="password">
        <input type="button" value="Click Me">
    </form>
</body>
</html>
```

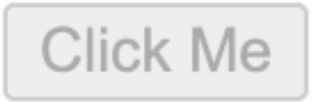


3.3. Features of Selectors in jQuery

- Using selectors in jQuery always returns an object than error if incorrectly specify the selector or no such selectors.
- Pros: jQuery code is executed normally without error
- Cons harder to find what wrong because no errors in the console, and the code still executed



```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $(':button').prop('disabled', true);
    $(':password').css('border', '2px solid red');
});
</script>
</head>
<body>
    <form id="formData">
        <input type="button" value="Click Me">
    </form>
</body>
</html>
```

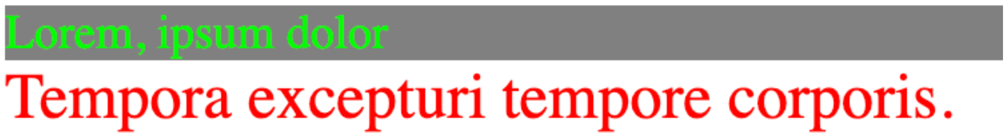


4. Methods Managing jQuery Styles

- jQuery allows to either add a style attribute to the specified selector (css() method) or control the addition/removal of the class attribute or values
- Syntax: $('#elementId').css(color:’red'); or $(‘.elementClass’).css(color:’red’);

  $(‘#elementId’).addClass(“className”);

```
<head>
<style>
 .intro { font-size: 150%; color: red;}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $('#block').css({
      color: 'lime',
      fontSize: '18px'
    });
    $('.example').css({
      'background-color':'grey',
    });
    $('#menu').addClass("intro");
});
</script>
</head>
<body>
    <div id="block" class = "example" style = "color: red"> Lorem, ipsum dolor </div>
    <div id="menu">Tempora excepturi tempore corporis. </div>
</body>
```

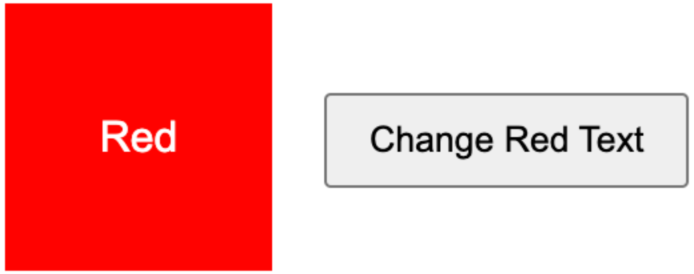


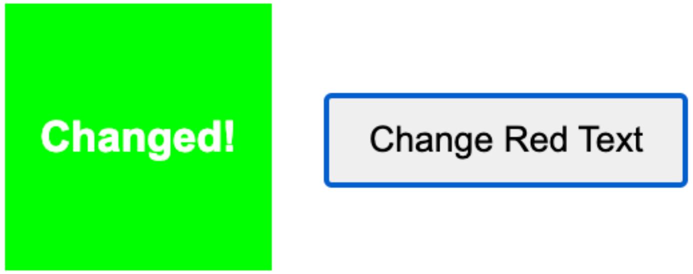
Lorem, ipsum dolor

Tempora excepturi tempore corporis.

Class Practice

```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
        body { font-family: Arial, sans-serif; max-width: 800px; margin: 0 auto; padding: 20px; }
        .box { width: 100px; height: 100px; margin: 10px; display: inline-block; text-align: center; line-height: 100px;  color: white;}
        .red { background-color: red; }
        button { padding: 8px 15px; margin: 5px; cursor: pointer; }
        .highlight { background-color: lime; font-weight: bold; }
  </style>
</head>
<body>
  <div class="exercise">
        <div class="box red">Red</div>
        <button id="change-red">Change Red Text</button>
  </div>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script>

  </script>
</body>
</html>
```





4.1. Traversing. DOM Traversing Methods

- jQuery has methods for traversing DOM elements.

- ”parent()”: returns the immediate parent node of the specified element.

  Syntax: `$(”elementName").parent().css({"color": "red", "border": "2px solid red"});`

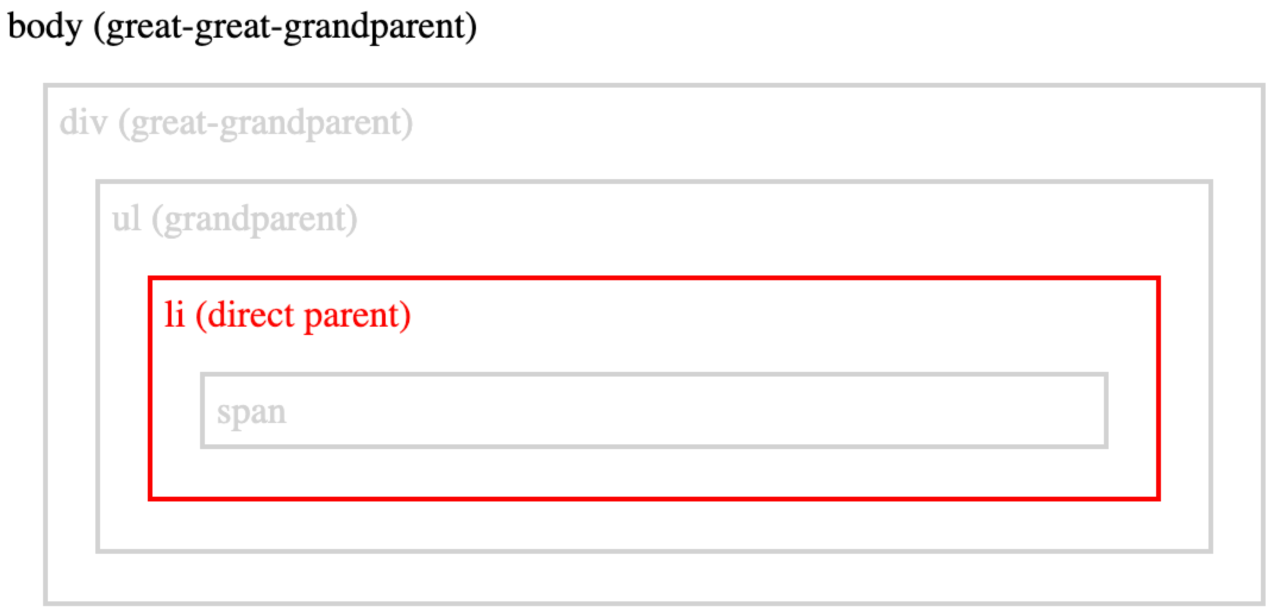
- ”parents()”: returns all ancestor elements of the selected element.

  Syntax: `$(”elementName").parents().css({"color": "red", "border": "2px solid red"});`

- ”find()”: all descendant elements that match a specified selector..

  Syntax: `$(”elementName").find(“elementName”).css({"color": "red"});`

```
<!DOCTYPE html>
<html>
<head>
            <style>.ancestors * { display: block; border: 2px solid lightgrey;  color: lightgrey; padding: 5px; margin: 15px;}</style>
            <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
            <script>
                $(document).ready(function(){
                        $("#childSpan").parent().css({"color": "red", "border": "2px solid red"});
                        //$("span").parent().css({"color": "red", "border": "2px solid red"});
                        //$("span").parents().css({"color": "red", "border": "2px solid red"});
                        //$("ul").find("span").css({"color": "red", "border": "2px solid red"});
                });
            </script>
</head>
<body class="ancestors">body (great-great-grandparent)
  <div style="width:500px;">div (great-grandparent)
      <ul>ul (grandparent)
              <li>li (direct parent)
                        <span id="childSpan">span</span>
              </li>
      </ul>
  </div>
</body></html>
```



body (great-great-grandparent)

div (great-grandparent)

ul (grandparent)

li (direct parent)

span

4.2. Event Handling in jQuery

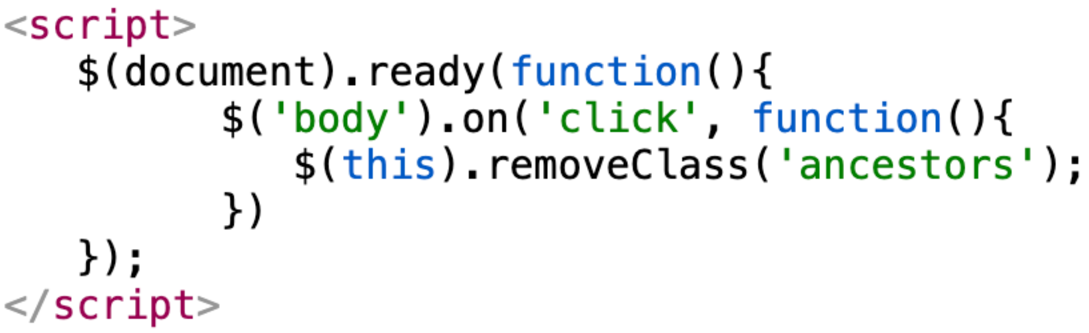
- Changes in styles/classes for HTML elements be assigned with handling events. In order to assign an event handler, use the on method:

- Syntax: $('selector').on(click, function(){ body function });

  $('selector').click(function(){ body function });

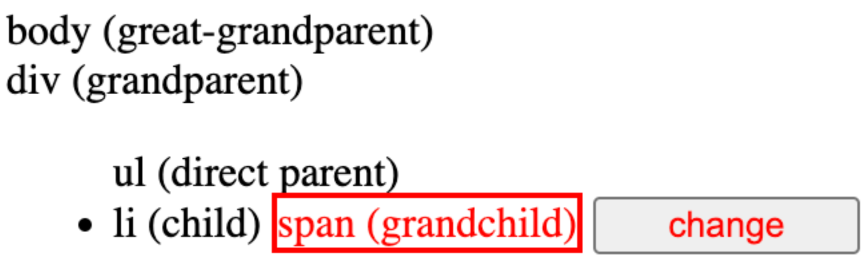
- Syntax: $('selector').on(mousedown, function(){ body function });

  $('selector').mousedown(function(){ body function });



```
<script>
   $(document).ready(function(){
         $('body').on('click', function(){
            $(this).removeClass('ancestors');
         })
   });
</script>
```

```
<!DOCTYPE html>
<html>
<head>
        <style>.ancestors * { display: block; border: 2px solid lightgrey;  color: lightgrey; padding: 5px; margin: 15px;}</style>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
        <script>
         $(document).ready(function(){
            $("ul").find("span").css({"color": "red", "border": "2px solid red"});
            $('input').on('click', function(){    $("body").removeClass('ancestors');})
         });
        </script>
</head>
<body class="ancestors">body (great-great-grandparent)
  <div style="width:500px;">div (great-grandparent)
     <ul>ul (grandparent)
          <li>li (direct parent)
                 <span id="childSpan">span (grandchild)</span>
                  <input id="buttonChange" type="button" value="change" style="width:100px; color:red"/>
          </li>
     </ul>
  </div>
</body></html>
```

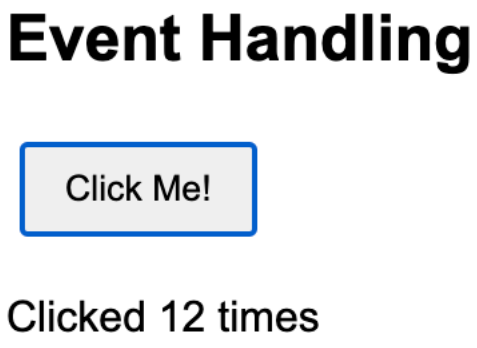


body (great-grandparent)
div (grandparent)

ul (direct parent)
- li (child) span (grandchild)

Class Practice

```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
        body {font-family: Arial, sans-serif; max-width: 800px; margin: 0 auto; padding: 20px; }
        button {padding: 8px 15px;  margin: 5px; cursor: pointer;}
  </style>
</head>
<body>
  <div class="exercise">
  <h2>Event Handling</h2>
  <button id="click-me">Click Me!</button>
  <p id="click-counter">Clicked 0 times</p>
  <!-- Load jQuery -->
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script>
  </script>
</body>
</html>
```



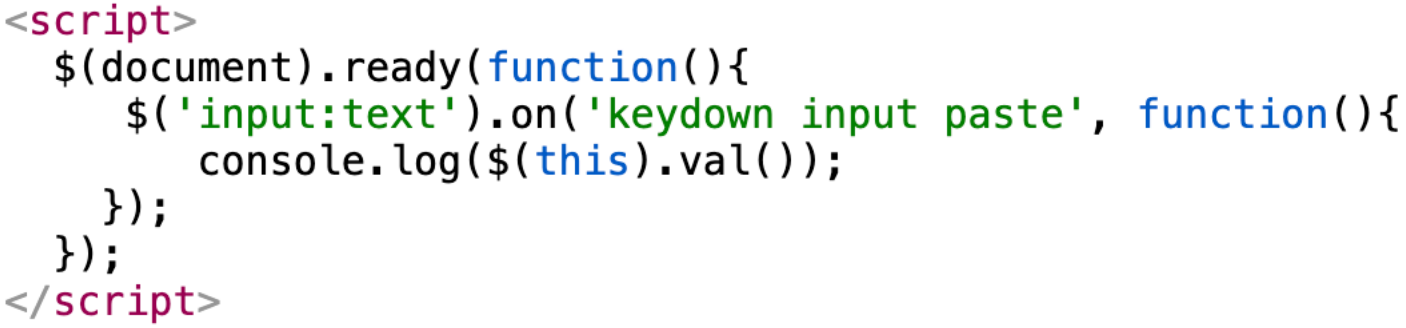
Event Handling

Clicked 12 times

4.3. Assign One Handler for Multiple Events

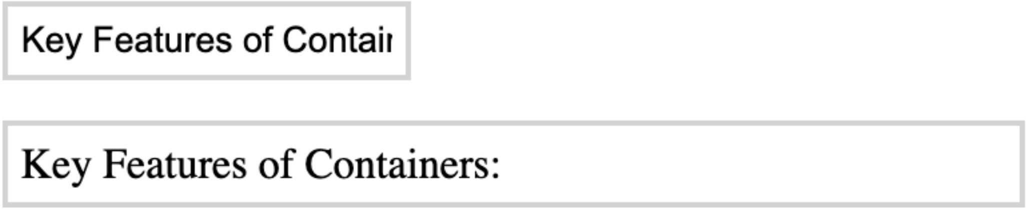
- The on() method can assign one function to handle multiple events at once.

- Syntax: $('selector').on(event1 event2 event3',, function(){ body function });



```
<script>
  $(document).ready(function(){
     $('input:text').on('keydown input paste', function(){
        console.log($(this).val());
    });
  });
</script>
```

```
<!DOCTYPE html>
<html>
<head>
        <style>.ancestors * { display: block; border: 2px solid lightgrey;  color: black; padding: 5px; margin: 15px;}</style>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
        <script>
          $(document).ready(function(){
                $('input:text').on('keydown input paste', function(){
                        document.getElementById("out").innerHTML = $(this).val();
                 });
          });
        </script>
</head>
<body class="ancestors">
        <input id="textInput" type="text"/>
        <p id="out"></p>
</body>
</html>
```



Key Features of Contaiı

Key Features of Containers:

4.4. Animation Methods in jQuery

jQuery allows setting animation effects for various elements in a simple way.

- show([time])/hide([time]): elements changing their height and width over time to show or hide
- toggle([time]): toggles the elements between visible/invisible.
- slideUp()/slideDown(): animated hiding/showing of element while changing its height.
- fadeIn(time): gradually change the opacity of elements from hidden to visible.
- fadeOut(time): gradually change the opacity of elements from visible to hidden.
- animate(): create custom animations on elements by changing their CSS properties over a specified duration.

Syntax: $(selector).animate({styles}, [duration], [easing], [complete]);

+ duration: 'fast' (200ms) or 'slow' (600ms). The default duration is 400ms.

+ easing: ‘swing’ or ’linear’

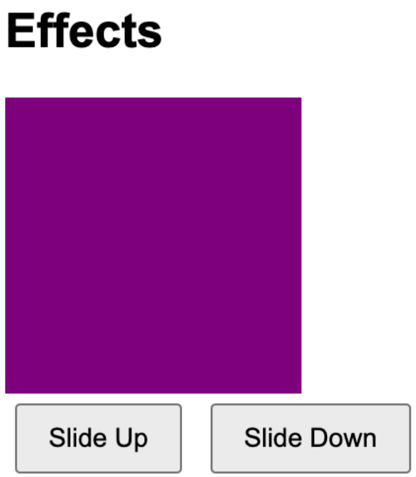
+ complete (optional): a callback function

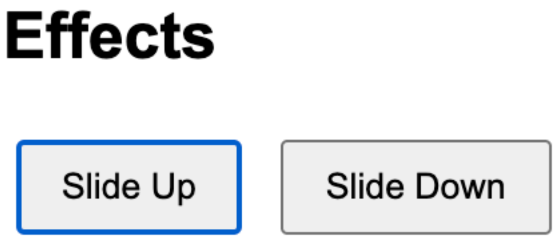
```
<!DOCTYPE html><html>
<head>
   <style>.ancestors * { display: block; border: 2px solid lightgrey;  color: black; padding: 5px; margin: 15px;}</style>
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
   <script>
                $(document).ready(function(){
                    $("#header").css("width", "200px");
                    $("#header").css("height", "50px");
                    $("#header").slideUp(1000);
                    $("#header").hide("slow");  // Animate the hide
                    $("#header").slideDown(1000);
                    $("#header").show("slow"); // Animate the show
                    //$("#header").css("visibility", "visible");  // hidden or visible
                    // $("#header").fadeOut(1000);
                     //$("#header").fadeIn(1000,0);
                });
   </script>
</head>
<body class="ancestors">
   <h2 id="header">IT Step Academy</h2>
</body> </html>
```

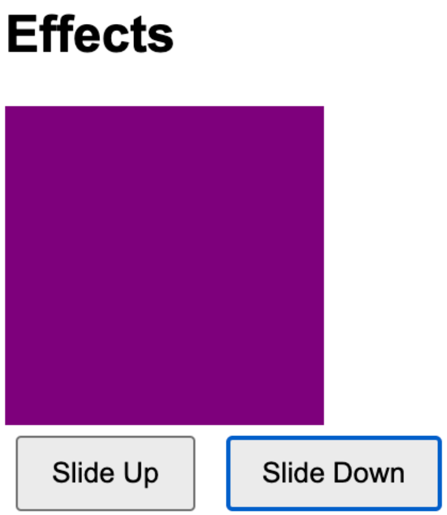
```
<!DOCTYPE html><html>
<head>
  <style>.ancestors * { display: block; border: 2px solid lightgrey;  color: black; padding: 5px; margin: 15px;}</style>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  <script>
		$(document).ready(function(){
			$("#myDiv").animate({ left: '50px', opacity: '0.5', height: '+=50px', width: '+=50px' }, 5000, 'linear', function() { alert("Animation complete!"); });
		});
  </script>
</head>
<body class="ancestors">
   <div id="myDiv">IT Step Academy</div>
</body>
</html>
```

Class Practice

```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
        body { font-family: Arial, sans-serif; max-width: 800px; margin: 0 auto;     padding: 20px;}
        .box { width: 100px; height: 100px; margin: 10px; display: inline-block; text-align: center; line-height: 100px; color: white; }
        button { padding: 8px 15px; margin: 5px; cursor: pointer; }
        #animated-box {width: 150px; height: 150px; background-color: purple; }
  </style>
</head>
<body>
  <div class="exercise">
        <h2>Effects</h2>
        <div id="animated-box"></div>
        <button id="slide-up">Slide Up</button>
        <button id="slide-down">Slide Down</button>
  </div>
  <!-- Load jQuery -->
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <!-- Your jQuery code -->
  <script>

  </script>
</body>
</html>
```





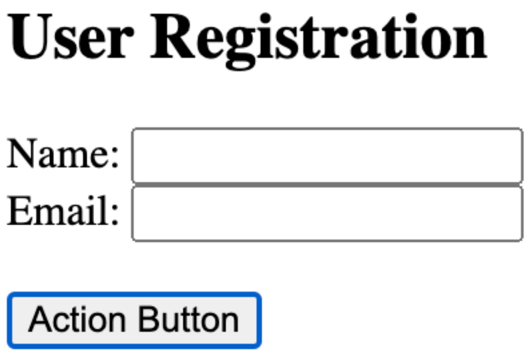


5. Delegating Events

- Delegating events: an event handler is assigned to the parent element, and the event is handled by child elements.
- Syntax: $('parent selector').on('event', 'child selector', function(){ ... });
- Ex: the click handling for elements like <button class="btn-plus"> which can be added to the form:

```
$('form').on('click', '.btn-plus', function(){
        console.log('button clicked');
});
```

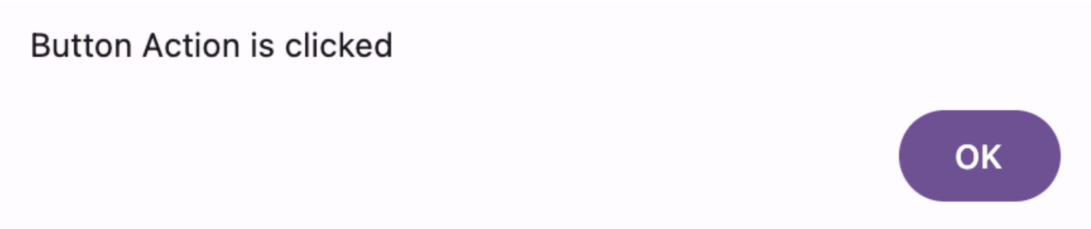
```
<!DOCTYPE html>
<html lang="en">
  <head>
            <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  </head>
  <body>
            <div id="form-container">
              <form id="main-form" class="user-form">
                        <h2>User Registration</h2>
                        Name:  <input type="text" id="name" name="name" required> <br>
                        Email: <input type="email" id="email" name="email" required> <br>
              </form>
              <button id="add-form" class="btn-action">Action Button</button>
            </div>
            <script>
              $(document).ready(function() {
                        // Delegated button click to form
                        $('#form-container').on('click', '.btn-action', function() {
                                    alert("Button Action is clicked");
                        });
              });
            </script>
</body></html>
```



User Registration

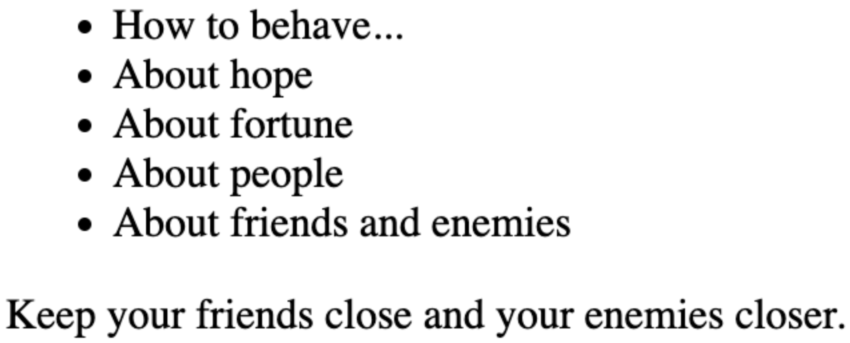
Name:

Email:



Button Action is clicked

```
<!DOCTYPE html><html>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  <script>
      $(()=>{
            const output = $('#output');
            $('ul#proverbs').on('mouseenter', 'li', (event)=>{
                        //alert($(event.target.value));
                        output.text($(event.target).data('text'));
            }).on('mouseleave', 'li', (event)=>{
                        output.text('');
            });
      })
  </script>
</head>
<body>
      <ul id = "proverbs">
            <li data-text="When in Rome, do as the Romans.">How to behave...</li>
            <li data-text="Hope for the best, but prepare for the worst.">About hope</li>
            <li data-text="Fortune favors the bold.">About fortune</li>
            <li data-text="Birds of a feather flock together.">About people</li>
            <li data-text="Keep your friends close and your enemies closer.">About friends and enemies</li>
     </ul>
     <div id="output"/>
</body> </html>
```



- How to behave...
- About hope
- About fortune
- About people
- About friends and enemies

Keep your friends close and your enemies closer.

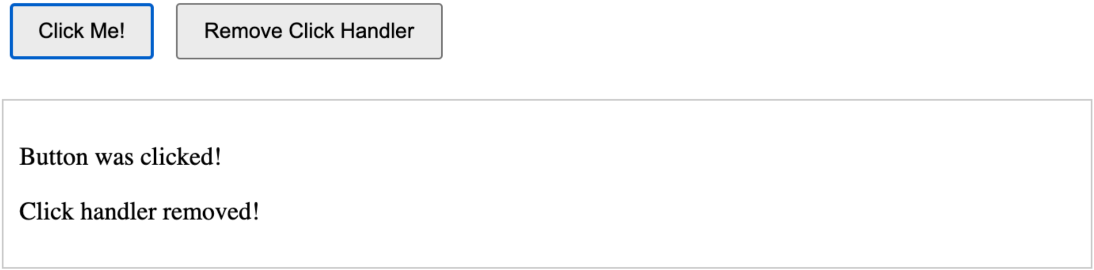
5.1. Canceling Event Handlers

- Event handler to be removed at some point, the off() method is used.

- Syntax:

```
// assign an event handler
$('selector').on('event', someFunc);

// cancel the event handler
$('selector').off('event', someFunc);
```

```
<!DOCTYPE html> <html lang="en">
<head>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <style>
        button { padding: 8px 16px; margin: 5px; cursor: pointer;}
        #message { margin-top: 20px; padding: 10px; border: 1px solid #ccc; min-height: 50px; }
  </style>
</head>
<body>
  <button id="clickMe">Click Me!</button>
  <button id="removeHandler">Remove Click Handler</button>
   <div id="message"></div>
<script>
   $(document).ready(function() {
        const clickHandler = function() {
                $('#message').append('<p>Button was clicked!</p>');
        };
        $('#clickMe').on('click', clickHandler);
        $('#removeHandler').on('click', function() {
                $('#clickMe').off('click', clickHandler);
                $('#message').append('<p>Click handler removed!</p>');
        });
  });
</script></body></html>
```



Button was clicked!

Click handler removed!

6. jQuery and AJAX

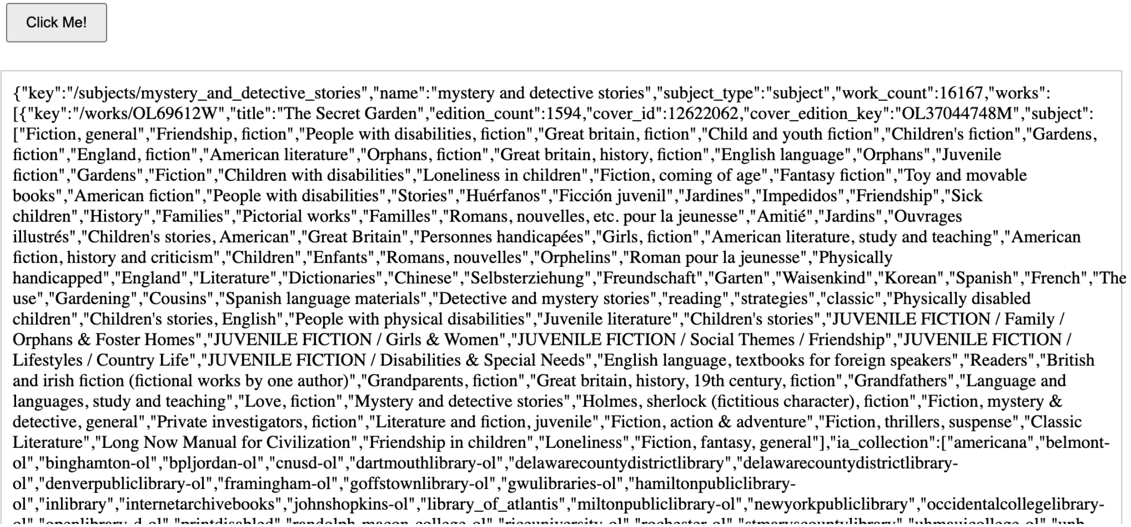
- jQuery provides methods for working with AJAX for asynchronous data load and transfer between the server and HTML page using JavaScript.

- API (*Application Programming Interface*) allows users of a website to receive/send information with a server-side.

- Syntax:

```
$.ajax({ url: ’api_url',    dataType: "json",}).done(function(data) {
        // the code that triggers if the form is successfully submitted to the server
).fail(function() {
        // the code that will run in case of error
});
```

```
<!DOCTYPE html> <html lang="en">
<head>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <style>
    button { padding: 8px 16px; margin: 5px; cursor: pointer;}
    #message { margin-top: 20px; padding: 10px; border: 1px solid #ccc; min-height: 50px; }
  </style>
</head>
<body>
  <button id="clickMe">Click Me!</button>
  <div id="message"></div>
  <script>
    $(document).ready(function() {
                const clickHandler = function() {
                  var category_url = "https://openlibrary.org/subjects/mystery_and_detective_stories.json";
                  $.ajax({url: category_url,  dataType: "json"
                  }).done(function (data) {
                              $('#message').append(JSON.stringify(data));
                  }).fail(function(data) {
                              alert("Error from mail!!!" + JSON.stringify(data));
                  });
                };
                $('#clickMe').on('click', clickHandler);
  });
</script></body></html>
```

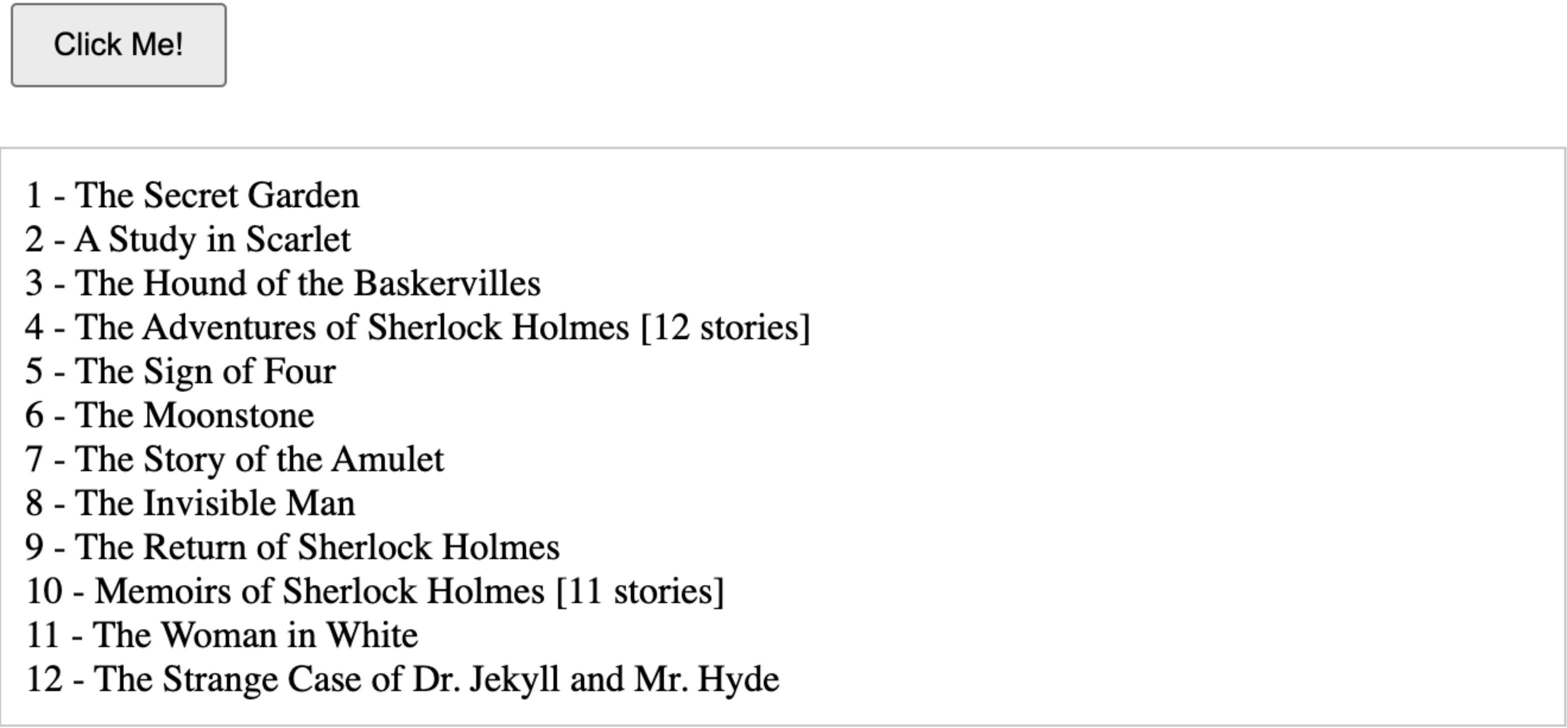


```
{"key":"/subjects/mystery_and_detective_stories","name":"mystery and detective stories","subject_type":"subject","work_count":16167,"works":
[{"key":"/works/OL69612W","title":"The Secret Garden","edition_count":1594,"cover_id":12622062,"cover_edition_key":"OL37044748M","subject":
["Fiction, general","Friendship, fiction","People with disabilities, fiction","Great britain, fiction","Child and youth fiction","Children's fiction","Gardens,
fiction","England, fiction","American literature","Orphans, fiction","Great britain, history, fiction","English language","Orphans","Juvenile
fiction","Gardens","Fiction","Children with disabilities","Loneliness in children","Fiction, coming of age","Fantasy fiction","Toy and movable
books","American fiction","People with disabilities","Stories","Huérfanos","Ficción juvenil","Jardines","Impedidos","Friendship","Sick
children","History","Families","Pictorial works","Familles","Romans, nouvelles, etc. pour la jeunesse","Amitié","Jardins","Ouvrages
illustrés","Children's stories, American","Great Britain","Personnes handicapées","Girls, fiction","American literature, study and teaching","American
fiction, history and criticism","Children","Enfants","Romans, nouvelles","Orphelins","Roman pour la jeunesse","Physically
handicapped","England","Literature","Dictionaries","Chinese","Selbsterziehung","Freundschaft","Garten","Waisenkind","Korean","Spanish","French","The
use","Gardening","Cousins","Spanish language materials","Detective and mystery stories","reading","strategies","classic","Physically disabled
children","Children's stories, English","People with physical disabilities","Juvenile literature","Children's stories","JUVENILE FICTION / Family /
Orphans & Foster Homes","JUVENILE FICTION / Girls & Women","JUVENILE FICTION / Social Themes / Friendship","JUVENILE FICTION /
Lifestyles / Country Life","JUVENILE FICTION / Disabilities & Special Needs","English language, textbooks for foreign speakers","Readers","British
and irish fiction (fictional works by one author)","Grandparents, fiction","Great britain, history, 19th century, fiction","Grandfathers","Language and
languages, study and teaching","Love, fiction","Mystery and detective stories","Holmes, sherlock (fictitious character), fiction","Fiction, mystery &
detective, general","Private investigators, fiction","Literature and fiction, juvenile","Fiction, action & adventure","Fiction, thrillers, suspense","Classic
Literature","Long Now Manual for Civilization","Friendship in children","Loneliness","Fiction, fantasy, general"],"ia_collection":["americana","belmont-
ol","binghamton-ol","bpljordan-ol","cnusd-ol","dartmouthlibrary-ol","delawarecountydistrictlibrary","delawarecountydistrictlibrary-
ol","denverpubliclibrary-ol","framingham-ol","goffstownlibrary-ol","gwulibraries-ol","hamiltonpubliclibrary-
ol","inlibrary","internetarchivebooks","johnshopkins-ol","library_of_atlantis","miltonpubliclibrary-ol","newyorkpubliclibrary","occidentalcollegelibrary-
ol","openlibrary_d-ol","printdisabled","randolph-macon-college-ol","riceuniversity-ol","rochester-ol","stmaryscountylibrary","uhmauicollege-ol","unh
```

Classwork

- Using jQuery to get only book title with API bellow

- API: "https://openlibrary.org/subjects/mystery_and_detective_stories.json";



1 - The Secret Garden
2 - A Study in Scarlet
3 - The Hound of the Baskervilles
4 - The Adventures of Sherlock Holmes [12 stories]
5 - The Sign of Four
6 - The Moonstone
7 - The Story of the Amulet
8 - The Invisible Man
9 - The Return of Sherlock Holmes
10 - Memoirs of Sherlock Holmes [11 stories]
11 - The Woman in White
12 - The Strange Case of Dr. Jekyll and Mr. Hyde