

 Warsaw University of Technology	Advanced Internet Programming
Project 7 jQuery	Maciej Iwańczyk 311258
Date: 12.04.2024	Grade:

## 1. Introduction:

In today's project I had to learn about jQuery and apply it in my Project.

## 2. What is jQuery?:

jQuery is a fast, small, and feature-rich JavaScript library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. It provides a simple and efficient way to manipulate the Document Object Model (DOM), handle events, and make asynchronous HTTP requests.

## 3. Link to the website (later it will be uploaded to UNIX server):

The website: <https://zstarwarss.github.io/AIPprojectMI/>

The files and the codes: <https://github.com/ZstarwarsS/AIPprojectMI>

## 4. jQuery implementation:

### 4.1. Description on hover:

```
<div id="smoothmenu1" class="ddsmoothmenu">
  <ul>
    <li><a href="#" class="page-link2" data-page="1" data-description="Project 1 Tables and input manipulation using JavaScript">Project 1</a></li>
    <li><a href="#" class="page-link2" data-page="2" data-description="Project 2 Styling a table using CSS and use of Tableless Tables">Project 2</a></li>
    <li><a href="#" class="page-link2" data-page="3" data-description="Project 3 Sorting using JavaScript">Project 3</a></li>
    <li><a href="#" class="page-link2" data-page="4" data-description="Project 4 Creating responsive websites">Project 4</a></li>
    <li><a href="#" class="page-link2" data-page="6" data-description="Project 6 JSON Technology and its Applications">Project 6</a></li>
  </ul>
  <br style="clear: left" />
</div>

<!-- Link Description -->
<div id="link-description"></div>
```

I have added a "data-description" data attribute for the descriptions of each project so that I can use them in a function.

```
48 $(document).ready(function() {
49     $('.page-link2').hover(function() {
50         var description = $(this).data('description');
51         $('#link-description').text(description);
52     }, function() {
53         $('#link-description').text('');
54     });
55 }
```

`$('.page-link2').hover(function() {...}, function() {...});` This line adds a hover event handler to all elements with the class `.page-link2`. The first function is executed when the mouse enters the element, and the second function is executed when the mouse leaves the element.

`var description = $(this).data('description');` Within the hover function, it retrieves the value of the `data-description` attribute of the hovered element using `$(this).data('description')`.

`$('#link-description').text(description);` It sets the text of the `#link-description` element to the retrieved description when hovering over a project link. Conversely, when the mouse leaves the link, it clears the text of the `#link-description` element.

## 4.2. Dynamically loading animated content:

```
function my_updateContent(content2, pageNumber) {  
    $('.content2').fadeOut('slow', function() {  
        if (pageNumber == 4) {  
            // Just fade in slowly for Project 4  
            $(this).html(content2).fadeIn('slow');  
        } else if (pageNumber == 6) {  
            // Slide from bottom to top for Project 6  
            $(this).html(content2).css('top', '100%').css('display', 'block').animate({'top': '0'}, 'slow');  
        } else {  
            // Slide from left to right for other projects  
            $(this).html(content2).css('left', '100%').css('display', 'block').animate({'left': '0'}, 'slow');  
        }  
    });  
}
```

In order to implement jQuery easily into my webpage I have come up with an idea of adding jQuery into my “my\_updateContent” function that updates the content of elements with the class `.content2` with the fetched content.

`$('.content2').fadeOut('slow', function() { ... });` This line selects all elements with the class `content2` and fades them out slowly. The `fadeOut` function takes a speed parameter ('slow' in this case) and a callback function to execute after the animation completes.

Inside the callback function:

a. `if (pageNumber == 4) { ... }`: If the `pageNumber` parameter is equal to 4, it means it's updating for Project 4. In this case, the content is immediately replaced (`html(content2)`) within the element, and then it fades in slowly (`fadeIn('slow')`).

b. `else if (pageNumber == 6) { ... }`: If the `pageNumber` parameter is equal to 6, it means it's updating for Project 6. Here, the content is replaced immediately, and then it's animated to slide from the bottom to the top. This is achieved by setting the initial CSS property `top` to 100% (to position it at the bottom), then setting `display` to `block` (to ensure it's visible), and finally animating the `top` property to 0 (to slide it to the top).

c. `else { ... }`: For any other project, the content is replaced immediately, and then it's animated to slide from the left to the right. Similar to the previous case, it sets initial CSS property `left` to 100%, sets `display` to `block`, and then animates the `left` property to 0 to slide it from left to right.