

## SIMPLIFYING JavaScript - BASICS OF jQuery



### What is our GOAL for this MODULE?

We learned the difference between JavaScript and jQuery and created a simple application to learn about the basics of jQuery.

### What did we ACHIEVE in the class TODAY?

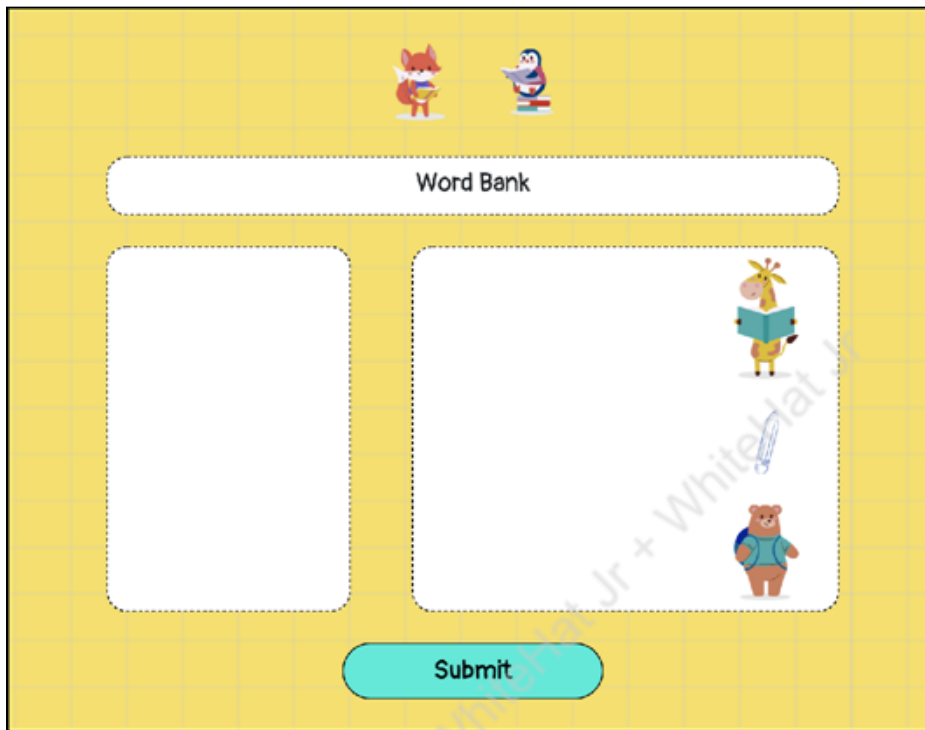
- Understood the difference between JavaScript and jQuery.
- Learned the basics of jQuery.
- Created an online Mad Libs using jQuery.

### Which CONCEPTS/CODING BLOCKS did we cover today?

- jQuery val(), text(), attr(), html(), append(), remove() methods
- jQuery event listeners click and keyup

### How did we DO the activities?

1. Clone the activity from Activity 1 and run the code.



2. Check out the libraries added in index.html.

```
<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />

  <title>MAD LIBS using jQuery</title>

  <!-- jQuery -->
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"
    integrity="sha256-/xUj+303U5yExlq6GSYGSHk7tPXikynS7ogEvEj/m4=" crossorigin="anonymous"></script>

  <!-- Bootstrap 4 -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
    integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxId0hoP+VmmDGMWISt9UJ0Z" crossorigin="anonymous">

  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.min.js"
    integrity="sha384-ltrjvnR4Twt/qOUYxe721u19sVFLVSA4hf/rRt6PrZTmPltdZcI7q7PXQBYTKyf"
    crossorigin="anonymous"></script>

</head>
```

```
<!-- Google Font -->
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Pangolin&display=swap" rel="stylesheet">
```

3. Add a document ready event with a function displayStory(), which will show the text content of the story.

```
<script>
  /* jQuery CODE GOES HERE! */

  $(document).ready(function () {
    displayStory();
  })

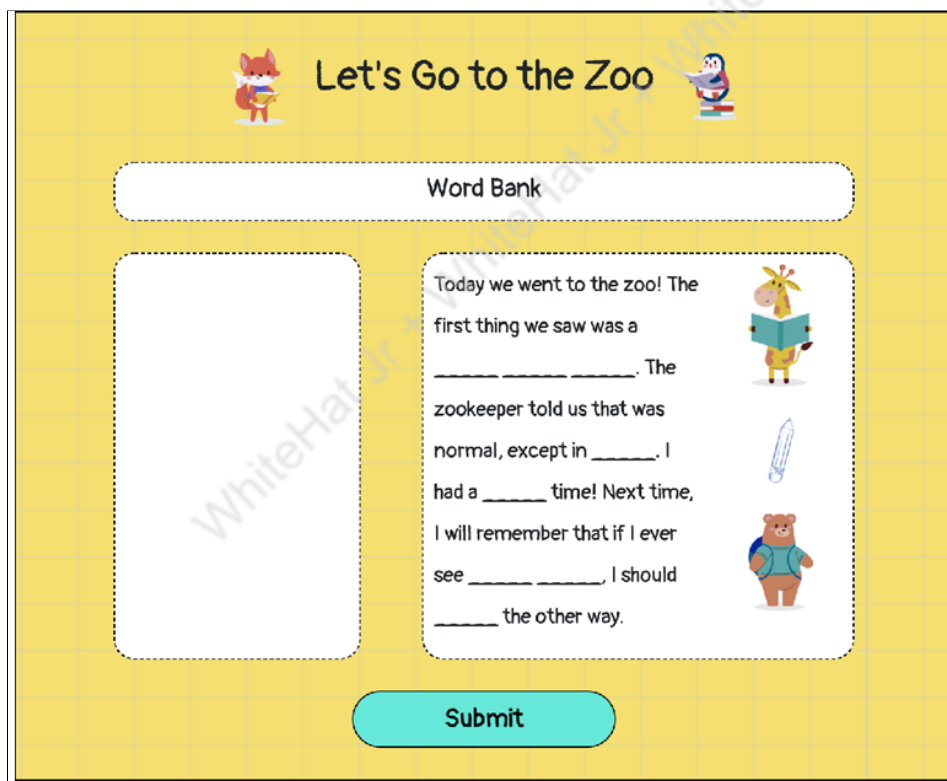
  function displayStory() {
  }
</script>
```

4. Observe the stories array in the boilerplate code with blanks for inputs created using <span> tag.

```
let stories = [
  {
    "inputs": 8,
    "title": "Let's Go to the Zoo",
    "story": "Today we went to the zoo! The first thing we saw was a <span class='rep_input'>____</span> <span class='rep_input'>____</span> <span class='rep_input'>____</span> <span class='rep_input'>____</span>.",
    "words": ["Black", "Gorilla", "Dancing", "Madagascar", "Nice", "White", "Tigers", "Move"]
  },
  {
    "inputs": 7,
    "title": "Picnic Time",
    "story": "On <span class='rep_input'>____</span> <span class='rep_input'>____</span> we are going on a picnic! I'm going with my <span class='rep_input'>____</span> <span class='rep_input'>____</span> and my favourite pet <span class='rep_input'>____</span>.",
    "words": ["Sunday", "Aunt", "Dog", "Burgers", "Soft Drinks", "Nice", "Cards"]
  },
  {
    "inputs": 12,
    "title": "Silly Animal Tale",
    "story": "There once was a <span class='rep_input'>____</span> <span class='rep_input'>____</span> <span class='rep_input'>____</span> <span class='rep_input'>____</span> from <span class='rep_input'>____</span> <span class='rep_input'>____</span>. Nobody knew he was <span class='rep_input'>____</span>.",
    "words": ["Smelly", "Cat", "California", "Cat", "Blue", "3", "Fishes", "Dance", "Songs", "Sad", "Childishly", "Happy"]
  }
]
```

5. Update the **displayStory()** function to use the **html()** method to set the story title and the story content and see the output.

```
function displayStory() {  
  
    //Get random story  
    randomNumber=parseInt(Math.random() * stories.length)  
    const story = stories[randomNumber];  
  
    //Set the story title  
    $("#story_title").html(story.title)  
  
    //Set the story content  
    $("#story_text").html(story.story)  
  
}
```



The screenshot shows a web application with a yellow background and a grid pattern. At the top, there's a title "Let's Go to the Zoo" flanked by a fox and a penguin. Below the title is a "Word Bank" section with a dashed border. To the left of the word bank is a large empty box for writing. To the right is a text area with a story template: "Today we went to the zoo! The first thing we saw was a \_\_\_\_\_. The zookeeper told us that was normal, except in \_\_\_\_\_. I had a \_\_\_\_\_ time! Next time, I will remember that if I ever see \_\_\_\_\_, I should \_\_\_\_\_ the other way." To the right of the text area are three cartoon animals: a giraffe reading a book, a pencil, and a bear. At the bottom is a blue "Submit" button.

6. Add **input boxes** for all the blanks(dashes) on the left side of the story content:
- Loop for the number of inputs in a given story.
  - Define a **<input>** type variable and set class, id and placeholder attribute.
  - Set the HTML **#input\_field** using **jQuery append()** method for all the boxes one after the other.

```
function displayStory() {
    //Get random story
    randomNumber=parseInt(Math.random() * stories.length)
    const story = stories[randomNumber];

    //Set the story title
    $("#story_title").html(story.title)

    //Set the story content
    $("#story_text").html(story.story)

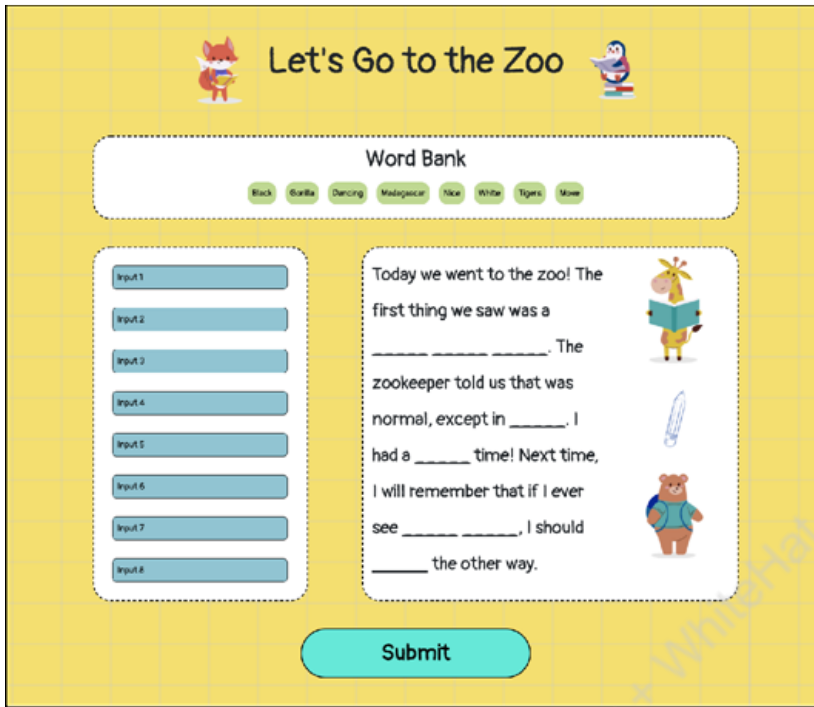
    //Make sure the input blanks(dashes) are empty when the story loads
    $("#input_fields").empty();

    //Set input boxes
    for (let i = 0; i < story.inputs; i++) {
        let input_html = `<input type="text" class="input_field" id="input_${i}" placeholder="Input ${i + 1}"/>`
    }
}

//Set input boxes
for (let i = 0; i < story.inputs; i++) {
    let input_html = `<input type="text" class="input_field" id="input_${i}" placeholder="Input ${i + 1}"/>`
    $("#input_fields").append(input_html)
}
```

7. Render words (for hints to fill in the blanks) that we have in our stories:
- Loop for the **words** array in the story.
  - Define a **<button>** type variable and set class attribute.
  - Set the HTML **#bank\_words** using **jQuery append()** method for all the words in the array one after the other.

```
$("#bank_words").empty();
for (let i = 0; i < story.words.length; i++) {
    let html = `<button class="word_bank_button">${story.words[i]}</button>`
    $("#bank_words").append(html)
}
```



- Listen to a **keyup** event on all the input fields by creating a listener for keyup event.

```
$(function () {
  $(".input_field").keyup(function () {
    let id = $(this).attr("id");
    let input_number = id.split("_")[1]
  })
})
```

- Update the HTML with the value of the input field.

```
$(function () {
  $(".input_field").keyup(function () {
    let id = $(this).attr("id");
    let input_number = id.split("_")[1]
    $(".rep_input").eq(input_number).html($(this).val());
  })
})
```

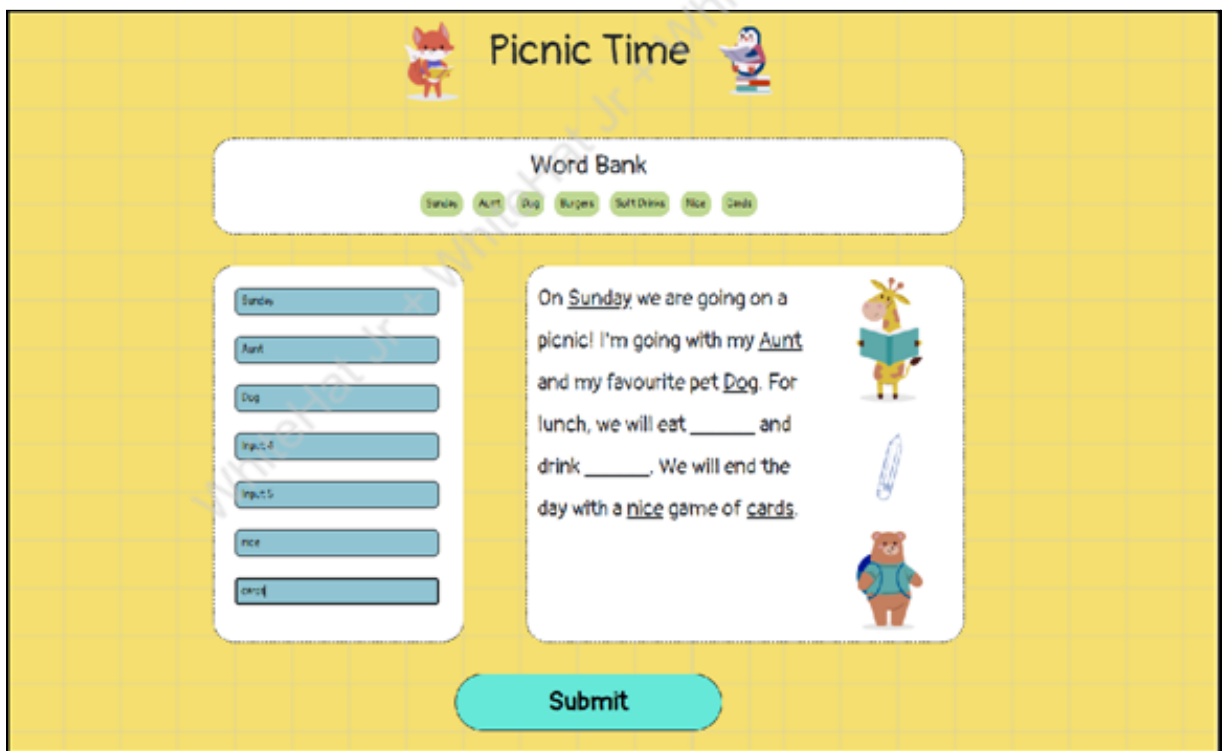
10. Add the **click** event on the button with id **#next\_story** and call the `displayStory()` function inside this listener function.

```
$(function () {
  $(".input_field").keyup(function () {
    let id = $(this).attr("id");
    let input_number = id.split("_")[1]

    $(".rep_input").eq(input_number).html($(this).val());
  })

  $("#next_story").click(function () {
    displayStory();
  })
})
```

11. Test the output by typing into different input fields and seeing how it is updating the blanks of the story in real time.



We learned to create a simple application to learn about the basics of jQuery.

### What's NEXT?

In the next class, we will learn about calling APIs using jQuery ajax() method.

### EXTEND YOUR KNOWLEDGE:

- You can refer to the link below to explore more about AFrame  
[A-Frame](#)
- You can refer to the link below to explore more about jQuery  
[jQuery](#)

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr