# PRO-C177

jQuery AJAX METHOD

# What is our GOAL for this MODULE?

We learned about the jQuery ajax() method and used the jQuery ajax() method call to update the score result of the Mad Libs story.

# What did we ACHIEVE in the class TODAY?

● Understood the use of AJAX techniques in web applications.

● Learned about the jQuery ajax() method.

● Showed the score result of the Mad Libs story using jQuery ajax() method call.

# Which CONCEPTS/CODING BLOCKS did we cover today?

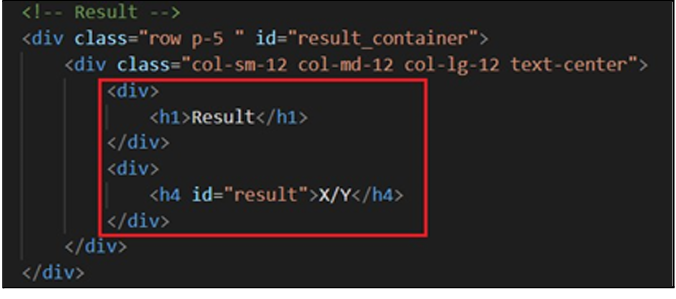
● jQuery val(), text(), attr(), html(), append(), remove()methods.

● jQuery event listeners click and keyup.

● jQuery ajax() method.

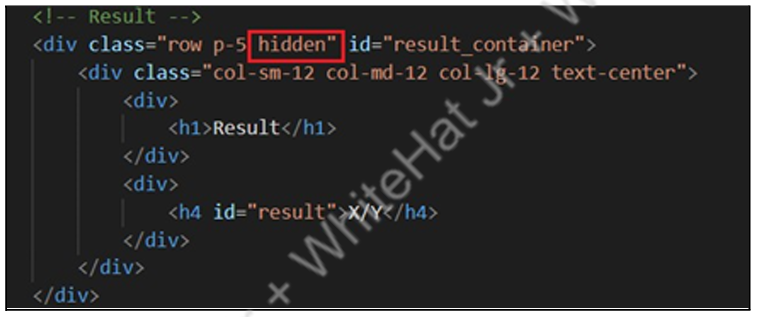
● Flask Python’s Framework to create APIs.

1. Add the<div>tags after the Word Bank div in index.html to show the result.

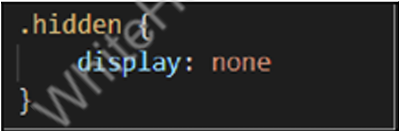


1. Change the style to hidden for the result div until the submit button is clicked:

● index.html



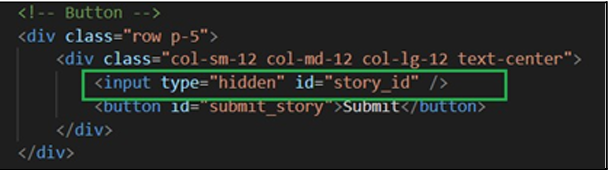
● index.css



3. To identify which story is submitted:

● Add the #story\_id<div> in the index.html file, which will be hidden

initially.

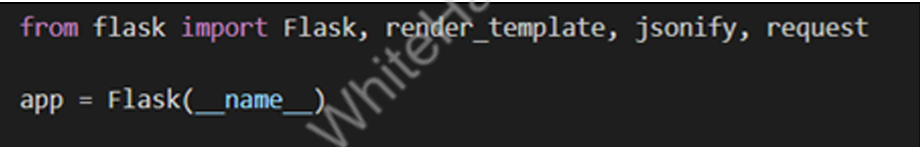


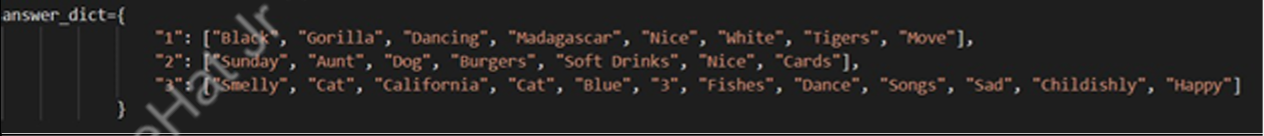
● Add the “story\_id” in the stories array (in index.js).



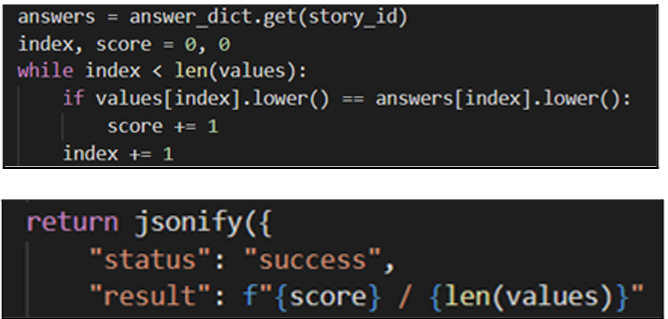
1. Write a flask POST API to get the result:

● Store answers in the variable answer\_dict in app.py and also also initialize flask.





● Add logic if the user's answer matches with the expected answer, we are increasing the score by 1 and return it to the user.



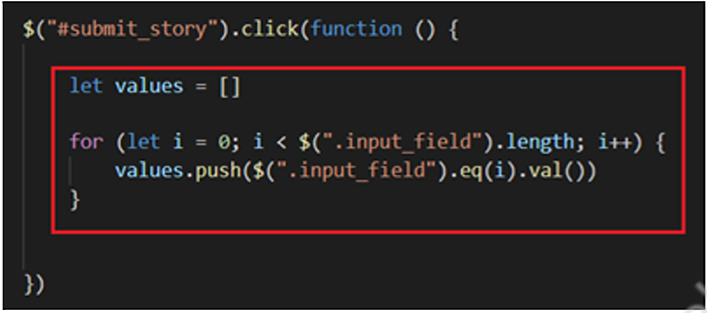
5. To Submit the story:

● Select the submit button element using #id.

● Add a click event in the index.js file.

● Loop through all the #input\_filed filled by the user.

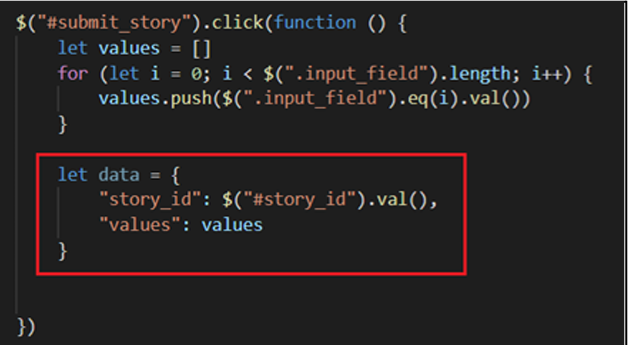
● Push all the inputs to an array ‘values’.



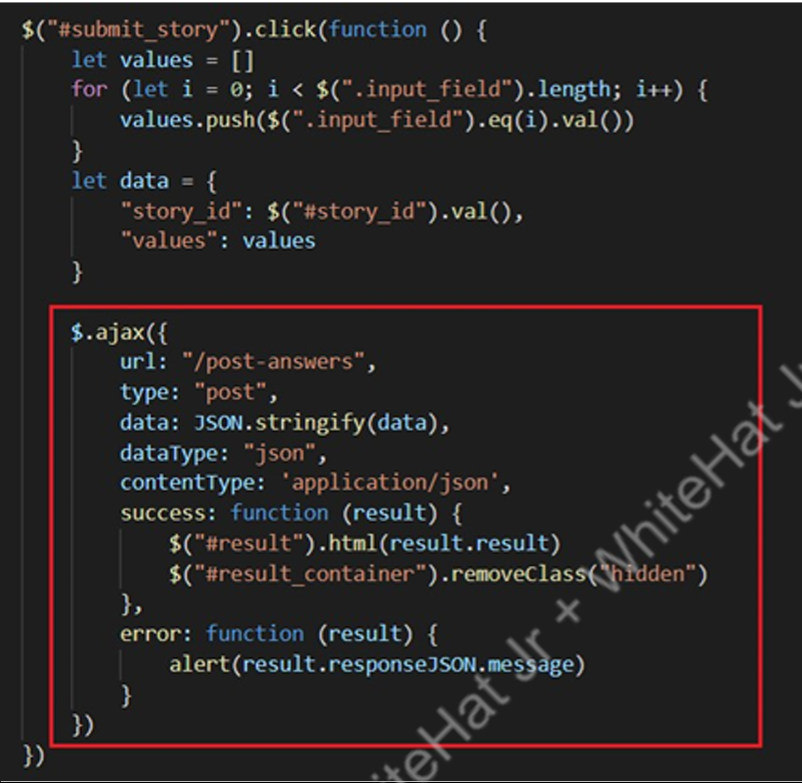
● UpdatedisplayStory()function with#story\_idin theHTML.



● Store the #story\_id and values of all the input\_field in a JSON object variable, data.

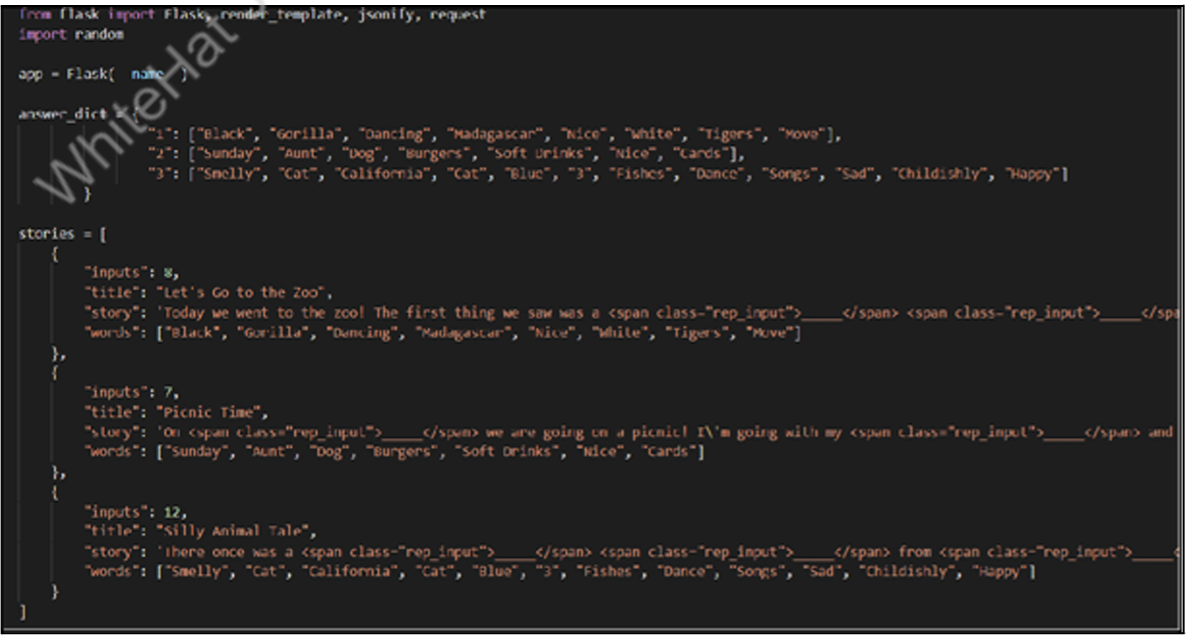


● Create an AJAX request with all the parameters.

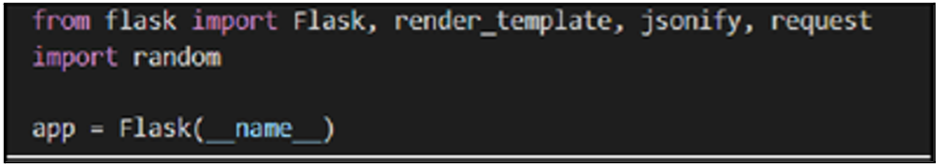


6. Write a flask GETAPI to get the stories:

● Move stories’ variables from “index.js” to “app.py”.



● Import the random module to select a random story in “app.py”.

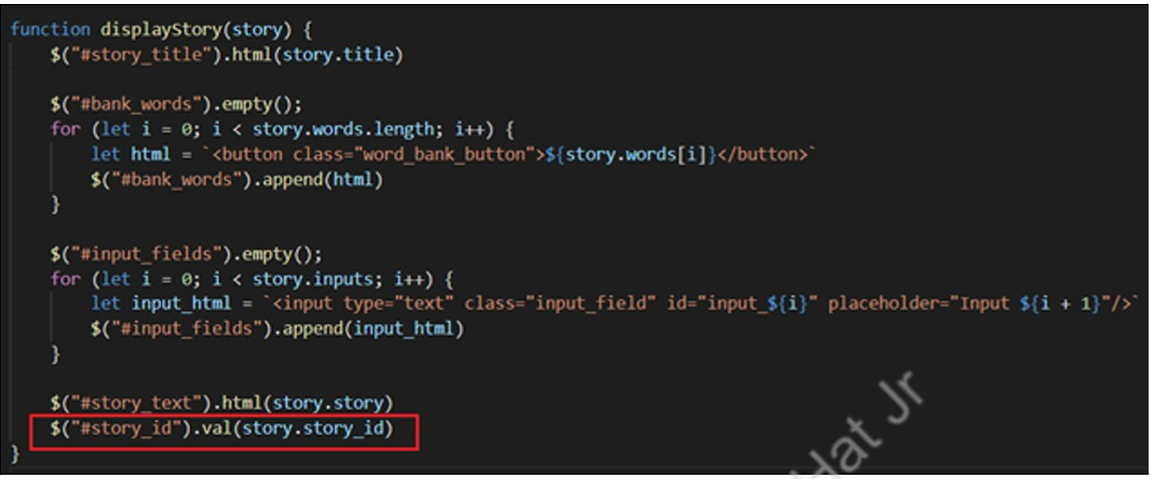




7. Write a function getStory() to call the API usingjQuery ajax() method.



8. Pass this story parameter in displayStory().



9. Test the output by typing into different input fields and seeing how it is updating the blanks of the story in real time and displaying the result after clicking on submit.



We learned to create a simple application to learn about the basics of jQuery and used jQuery ajax() method to get the result also to get a random story each time

# What’s NEXT?

In the next class, we will learn about rendering maps in a web application.

# 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.