1. Which, if any, of the following 3 code fragments are equivalent? Explain why they are different, if they are. Explain why they can have different parameters and be equivalent, if they are equivalent.

//code fragment 1 $("li").each(function(idx, e) { $(e).css(“color”, “yellow”); });

//code fragment 2 $("li").each(function() { $(this).css(“color”, “yellow”); });

//code fragment 3 $("li").each(function(idx) { $(this).css(“color”, “yellow”); });

Answer:

All three code fragments have the same functionality and will set the color of each **li** element to yellow. However, they differ in their syntax and parameter usage.

Code fragment 1 uses the **.each()** method to loop through each **li** element, passing in two parameters to the callback function: the **idx** index and the **e** element. The **idx** index is not used in the callback function, but it can be used to access the current element's index in the collection. Inside the callback function, **$(e)** is used to wrap the current element as a jQuery object so that its CSS property can be manipulated.

Code fragment 2 is similar to code fragment 1, but it uses the **this** keyword to refer to the current element instead of passing it as a parameter. The **this** keyword is a reference to the current element in the iteration.

Code fragment 3 also uses the **this** keyword to refer to the current element, but it only passes one parameter to the callback function, which is the index of the current element in the collection. The **idx** index is not used in the callback function.

Overall, all three code fragments are functionally equivalent, but they differ in their parameter usage and syntax. The choice of which code fragment to use may depend on personal preference or the specific requirements of the situation.

2. Write a jQuery expression to find all divs on a page that include an unordered list in them, and make their text color be blue.

no ul here

This does contain a ul.

* the first item
* the second item

<!—INSERT YOUR JQUERY CODE HERE - - >

</script>

$("div:has(ul)").css("color", "blue");

</body>

3. Write jQuery code to append the following div element (and all of its contents) dynamically to the body element.

**JQuery Core**

HTML:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>JS Bin</title>

</head>

<body>

</body>

</html>

Answer:

$(document).ready(function() {

$("body").append('<div><h1>JQuery Core</h1></div>');

});

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>