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”); });

All three code fragments are equivalent in terms of functionality, as they all select all li elements on the page and change their font color to yellow. The differences between the code fragments lie in their use of parameters.

In code fragment 1, the function passed to .each() takes two parameters: idx and e. idx is the index of the current element in the set, and e is the current element itself. e is wrapped in a jQuery object to allow manipulation of the element's CSS properties.

In code fragment 2, the function passed to .each() takes no parameters, but uses this to refer to the current element being iterated over. Again, this is wrapped in a jQuery object to allow manipulation of the element's CSS properties.

In code fragment 3, the function passed to .each() takes one parameter: idx, which is the index of the current element in the set. Like in code fragment 2, this is used to refer to the current element, and it is wrapped in a jQuery object to allow CSS manipulation.

Therefore, all three code fragments are equivalent in terms of their effect on the li elements, but they differ in the parameters passed to the function and how the current element is referenced.

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.

 <div>no ul here </div>

 <div>

 This does contain a ul.

 <ul>

 <li>the first item</li>

 <li>the second item</li>

 </ul>

 </div>

 <script>

 $(document).ready(function(){

 $("#btn").click(function(){

 $("div ul").css("color", "blue")

 });

});

 </script>

 </body>

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

<div><h1>JQuery Core</h1></div>

HTML:

 <!DOCTYPE html>

 <html>

 <head>

 <meta charset="utf-8">

 <title>JS Bin</title>

 </head>

<body>

<script>

$(document).ready(function(){

 $("#btn").click(function(){

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

 });

});

</script>

 </body>

 </html>