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 are equivalent in the sense that they all perform the same operation: They iterate over all li elements on the page and change their color to yellow.

In the first fragment, two arguments idx and e are passed to the function. idx refers to the index of the current element in the array, and e refers to the current element itself. Then $(e) is used to create a jQuery object from the current element.

In the second fragment, no arguments are passed to the function. Instead, this is used, which in the context of a jQuery .each() function refers to the current DOM element. Therefore, $(this) also creates a jQuery object from the current element.

In the third fragment, only one argument idx is passed to the function, which refers to the index of the current element in the array. However, it is not used within the function. Again, this is used to create a jQuery object from the current element.

The fact that different parameters can be passed but the fragments remain equivalent lies in the nature of JavaScript functions. If an argument is defined but not used within the function, it has no effect on the function's output.

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(){
$("div:has(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(){
 $('body').append('<div><h1>JQuery Core</h1></div>');
});
</script>
 </body>
 </html>
```