

1. What is the difference between the following 2 statements?

```
setTimeout(booyah, 2000);  
setTimeout(booyah(), 2000);
```

Answer:

The difference between the above two statements is, the first function uses the function name to establish or pass a reference with that specific function called booyah. This means the function was not invoked immediately until the setTimeout uses the referenced object to call the function. On the second statement, the caller function uses a parenthesis explicitly to invoke the function. In this case a value returned to the caller during the invocation.

2. What do the following 2 alerts display (answer without running the code)?

```
var myfunc = function(a, x) { return a * x; };  
var x = myfunc(2, 3);  
var y = myfunc;
```

```
alert(x); // Answer : 6  
alert(y(2,3)); // Answer : 6
```

3. Write functions booyah1 and booyah2 so that in both cases below, an alert box comes up after 2 seconds that says "BOOYAH!"

```
setTimeout(booyah1, 2000);  
setTimeout(booyah2(), 2000);
```

```
function booyah1(){  
    alert("BOOYAH!");  
}  
function booyah2(){  
    alert("BOOYAH!");  
}
```

4. What is "Unobtrusive Javascript"? What is the practical application of Unobtrusive Javascript (and the reasons for using it)?

Unobtrusive **JavaScript** is the way of writing JavaScript language in which we properly separate Document Content and Script Content thus allowing us to make a **clear distinction** between them. Just as we should separate our structure and presentation by putting all CSS in a separate file and minimizing the use of style attributes or other such presentational markup, we should also separate our **HTML structure** and **JavaScript behaviour**. The reasons are the same: it separates your concerns, keeps your code clean, and allows you to work on the JavaScript without touching either HTML or CSS. So it is basically separating behavior or javascript from presentation or html. This approach is useful in so many ways as it makes our code less error prone, easy to update and to debug.