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

 setTimeout(booyah, 2000);

 setTimeout(booyah(), 2000);

setTimeout(booyah, 2000): This is the actual way of giving parameters. Here after 2000 milliseconds, booyah() method is called;
setTimeout(booyah(), 2000): This is not the right way of giving parameters. Here without waiting 2000 milliseconds it directly jumps to booyah() and performs it.

### 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);

 alert(y(2,3));

alert(x) = 6
alert(y) = 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);

 Answers:

 function booyah1(){

 alert(“BOOYAH!”);

 }

 function booyah2(){

 setTimeout(booyah1, 2000);

 }

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

Answers: This is a standard and the right way of writing javascript code. This says that javascript should not be colluded with html and css. They should be writing separately in their own files. This also kind of aligns with the principle of separation of concern. One of the reasons to write Unobtrusive javascript is because adding javascript should be a constructive addition of new features. It should not destroy the existing html content when we want to update javascript. The other reason is that not all browsers support the javascript you write, so if the javascript is separated out from other components of a web page, at least the web page will work but with lower functionality i.e. with only html and css. Also, separating out javascript completely will help fellow developers who inherit the code to update and/or add features in the future later