**Javascript Timer Exercise**

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

setTimeout(booyah, 2000);

setTimeout(booyah(), 2000);

**setTimeout(booyah, 2000); It waits for 2000 ms before loading the function.**

**setTimeout(booyah(), 2000); It calls the function immediately, rather than waiting for the specified time.**

1. **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));

**it alerts 6 twice**

1. 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() {

//It gets executes after waiting 2000 milliseconds

alert(“BOOYAH);

}

function booyah2() {

//It gets immediately executes as soon as the function is invoked.

alert(“BOOYAH);

}

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

The method of separating Behavior from Content and Presentation in web designing then such Javascript is called unobtrusive Javascript.

The practical application of using Unobtrusive Javascript is:

1. Usability: An unobtrusive DOM script does not draw the attention of the user - visitors use it without thinking about it.
2. Graceful degradation: Unobtrusive DOM scripts never generate error messages, in any browser, even when they fail. If features cannot be presented properly, they silently disappear.
3. Accessibility: If any script fails, the page still delivers its core functions and information via the markup, stylesheets and/or server-side scripting.
4. Separation: For the benefit of other and future web developers, all JavaScript code is maintained separately, without impacting other files of script, markup or code.