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

setTimeout(booyah, 2000);

setTimeout(booyah(), 2000);

answer:

On setTimeout(booyah, 2000) the content or the result on booyah function will be displayed or called after 2000 milliseconds

On setTimeout(booyah(), 2000) the content on booyah function is called immediately or with out waiting 2000 milliseconds.

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));

Answer:

Both alerts display 6

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);

Answer;

function booyah1() {

document.getElementById("output").innerHTML = "BOOYAH!";

}

Answer:

**For booyah1:**

function **booyah1**() {

        setTimeout(myalert, 2000);

        }

        function **myalert**(){

            alert("BOOYAH!")

        }

……………………………………………………………………………….

**For booyah2 :** setTimeout(booyah2(), 2000);

**Important!!!** it calls the function immediately, rather than waiting the 2000ms!

function **booyah2**() {

        setTimeout(myalert, 2000);

        }

        function **myalert**(){

            alert("BOOYAH!")

        }

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

Answer:

 a way of writing JavaScript so that your site visitors are not shut out of your site for one of these reasons—even if your JavaScript is not working correctly for them, they should still be able to use your site, albeit at a more basic level.

Separation of concerns i.e the HTML markup is now clean without any traces of javascript. Page load time is better. It is also easy to update the code as all the Javascript logic is present in a separate file