

Javascript Exercises

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

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

answer :

[*setTimeout(booyah, 2000);*] will call the function of *booyah* after waiting the 2000ms.

[*setTimeout(booyah(), 2000);*] will call the function of *booyah* immediately, rather than waiting the 2000ms.

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));
```

answer:

[*alert(x);*] will pop up 6 on the window.

[*alert(y(2,3));*] will pop up 6 on the window.

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);  
answer:  
function booyah1() {  
    alert("booyah1_BOOYAH!");  
}  
function booyah2() {  
    setTimeout(function() {  
        alert("booyah2_BOOYAH!");  
    }, 2000);  
}
```

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

Answer:

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 eschewing 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.