Javascript Timer Exercises

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

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

- This one will call the function booyah after a delay of 2000 milliseconds.

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

- This one will immediately call the function booyah.

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));
```

- Both of them will display 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);
```

-
  ```
  function booyah1(){
       alert(‘BOOYAH!’);
  }
  ```

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

-
  ```
  function booyah2(){
       return function() { alert(‘BOOYAH!’);}
  }
  ```

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

- It is an approach that separates the behavior of a web page from its structure and presentation. It is HTML with minimal JavaScript inside it. It keeps the HTML code simple without any inline JavaScript and attaches behavior to the HTML elements using external JavaScript files.

- The practical application is to make web pages is maintainable, accessible and scalable.

- One reason can be cross-browser compatibility. Unobtrusive JS can help ensure the code works consistently across different browsers and platforms. Another reason could be scalability. Unobtrusive JavaScript allows for the creation of complex web applications with a large amount of JavaScript code, without affecting the performance of the page.