Javascript Timer Exercises

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

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

Answer:

The first code calls the function `booyah` after 2 seconds, which is done by **function reference** as there is no parenthesis.
The second code fails with a callback error because the we are trying to execute the function immediately, by adding the parenthesis, this becomes **function evaluation**/execution, which results in a call back error.

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 output 6.
and
alert(y(2,3)) will output 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);
```

Answer:

```
const booyah1 = () => {
	alert(“BOOYAH!”);
}

setTimeout(booyah1, 2000);
setTimeout(function booyah2(){
        alert(“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 art of separating webpages into 3 different concerns:

- HTML (content)
- CSS (presentation)
- JavaScript (behavior)

The reasons for using it are for Accessibility, Maintainability, Flexibility, Compatibility.
Making sure that people with outdated browsers, or people with disabilities can access web pages, also making sure that programmers can easily test and debug their code.