**1.What is a potential pitfall with using**typeof bar === "object"**to determine if**bar**is an object? How can this pitfall be avoided?**

**2.What will the code below output to the console and why?**

(**function**(){

**var** a = b = 3;

})();

console.**log**("a defined? " + (**typeof** a !== 'undefined'));

console.**log**("b defined? " + (**typeof** b !== 'undefined'));

**3.What will the code below output to the console and why?**

**var** myObject = {

foo: "bar",

func: **function**() {

**var** self = this;

console.**log**("outer func: this.foo = " + this.foo);

console.**log**("outer func: self.foo = " + self.foo);

(**function**() {

console.**log**("inner func: this.foo = " + this.foo);

console.**log**("inner func: self.foo = " + self.foo);

}());

}

};

myObject.**func**();

**4.What is the significance of, and reason for, wrapping the entire content of a JavaScript source file in a function block?**

**5.What is the significance, and what are the benefits, of including**'use strict'**at the beginning of a JavaScript source file?**

**6.Consider the two functions below. Will they both return the same thing? Why or why not?**

function **foo1**()

{

**return** {

bar: "hello"

};

}

function **foo2**()

{

**return**

{

bar: "hello"

};

}

**7.What will the code below output? Explain your answer.**

console.**log**(0.1 + 0.2);

console.**log**(0.1 + 0.2 == 0.3);

**8.In what order will the numbers 1-4 be logged to the console when the code below is executed? Why?**

(**function**() {

console.**log**(1);

setTimeout(**function**(){console.**log**(2)}, 1000);

setTimeout(**function**(){console.**log**(3)}, 0);

console.**log**(4);

})();

**9.Write a simple function (less than 160 characters) that returns a boolean indicating whether or not a string is a**[**palindrome**](http://www.palindromelist.net/)**.**

**10.Write a**sum**method which will work properly when invoked using either syntax below.**

console.log(sum(2,3)); // Outputs 5

console.log(sum(2)(3)); // Outputs 5