Assignment 6

Make sure to test all your answers in the console.

1. Consider the following code:

```
var o = { a: 5 };
o.foo = ......   // The different answers provide this.
var f = o.foo;
f();          // Must return 5
o.a = 2;
f();          // Must return 2
```

We want the call to `f` to return the value 5 the first time and 2 the second time. Each of the options below will set a value for `o.foo`. Put a check next to those options that would make it so that the calls to `f` behave as described.

```
// Option A
o.foo = function() { return a; };

// Option B
o.foo = function() { return this.a; };

// Option C
o.foo = function() { this.a; }

// Option D
o.foo = (function() {
    return function() { return this.a; };
}());

// Option E
o.foo = (function() {
    return function() { return o.a; };
}());

// Option F
o.foo = (function() {
    var a = o.a;
    return function() { return a; };
}());

// Option G
o.foo = (function() {
    return function() { return this.a; };
}(o));

// Option H
o.foo = (function(v) {
    return function() { return v.a; };
}(o));

// Option I
o.foo = (function() { return this.a; }).bind(o);
```

2. Explain the output of the following code, in particular the print-outs of the last 4 lines:

```
function Foo(v) { }
Foo.prototype = { f: function() { return "hi there!"; } };
var foo1 = new Foo();
Foo.prototype.g = function() { return "hi again!"; };
Foo.prototype = {
    f: function() { return "huh?"; },
    g: function() { return "double huh?"; }
};
var foo2 = new Foo();
console.log(foo1.f());
console.log(foo1.g());
console.log(foo2.f());
console.log(foo2.g());
```