Solution Objects and Prototypes

Instructions:

Solutions of the exercises are to be delivered before Thursday, the 03th of May at 10:15AM. Solutions should be placed in a separate folder with the name “**Assignment08**”.

Exercise 1 (2 points)

You should implement the `Person` prototype which has the following characteristics:

1. The `name` property which is accessible by other objects of this prototype.
2. The `password` property which cannot be accessed by other objects of this prototype.
3. The `counter` property whose value is shared among all objects of this prototype.

You need to create two objects of this prototype in order to do the following tasks:

1. Show how to access the `name` property in both objects.
2. What output do you get if you try to access the `password` property? How can you correct it?
3. How to access the `counter` property? If you change the value of the `counter` property in one object, does it affect the property value on the second object?

Answer:

```
var person = function (nm) {
this.name = nm
var secret = "Secret";
    shared_secret = 25;


this.getSecret = function () {
return secret;
}
this.setSecret = function () {
secret = "not secret";
}
    this.getName = function () {
        return this.name;
    }
this.getAge = function() {
return shared_secret;
};
this.growOlder = function () {
shared_secret++;
}

    return this;
};



var person1 = new person("mh1");
var person2 = new person("mh2");
console.log(person1.name);
console.log(person2.name);
console.log(person2.secret);
console.log(person2.secret);
console.log(person1.getSecret());
person1.setSecret();
console.log(person1.getSecret());
console.log(person2.getSecret());



console.log(person1.getAge()); // 25
console.log(person2.getAge()); // 25
person1.growOlder();
console.log(person1.shared_secret)
console.log(person1.getAge()); // 26
console.log(person2.getAge()); // 26

console.log(person1.shared_secret);
```

Exercise 2 (2 points)

A possible classification for animals is shown in Figure 1. When it comes to classify the platypus you realize that it nurses but it also spawns. So implement the class diagram shown in figure 1 including the poor platypus in Java Script. Use the poorPlatypusToSolve.html file as skeleton for your implementation.
Answer:

```
var animal = {}

var mammal = Object.create(animal);
mammal.numberofNipples = 2;
```

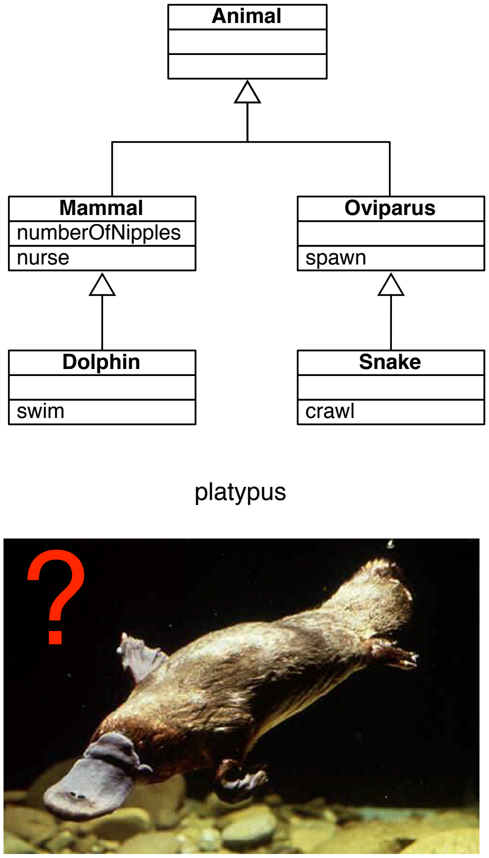

Figure 1: Animal classification

```
mammal.nurse = function () {
alert("I am nursing");
}
mammal.getNumberOfNipples = function(){
alert(this.numberofNipples);
}

dolphin = Object.create(mammal);
dolphin.numberofNipples = 4;
dolphin.swim = function() {
alert("I am swimming");
}

var oviparus = Object.create(animal);
oviparus.spawn = function() {
alert("I am spawning");
}
```

```
var snake = Object.create(oviparus);
snake.crawl = function() {
alert("I am crawling");
}

var platypus = Object.create(mammal);
platypus.numberofNipples = 6;
platypus.spawn = oviparus.spawn;

function display(text) {
document.getElementById("output").innerHTML += text + "\n";
}
```

Exercise 3 (2 points)

In this exercise, the prototype `People` is defined. You need to define two objects of this prototype. The first object must be created by the `Object.create()` method and the second object must be created by a constructor function. Finally, to check if your solution is correct, run the `checkit` method from the template. This method must return `Bern` in both objects.

Answer:

```
var People = function () {
    this.username = "Bern";
return this;
};

People.prototype.checkit = function() {
return this.username;
}


var c = new People();
d = Object.create(People.prototype);

People.apply(d);

console.log(c.checkit());
console.log(d.checkit());
```