



JavaScript OOP Basics

Mariyan Apostolov

mariyan.apostolov89@gmail.com

JavaScript OOP?

- Обектно ориентирано програмиране?
- Създаване на обект в JavaScript
- Комбиниране на данни и функции в обект
- Какво представлява затварнето на функция /Closure/
 - “Модулен” (Module) шаблон използвайки Closure
 - “Разкриващ модул” (Revealing Module) шаблон използвайки Closure

Обектно ориентирано програмиране?

Типове данни:

- Булев тип
- Числов тип
- Низов тип (стрингов)
- Масив
- Обект
 - Обекта е съвкупност от възможностите които този обект притежава. Те могат да бъдат и прости типове данни (който разгледахме по горе), други обекти или функции

Създаване на обект в JavaScript

```
var student = {  
  firstName: 'Maria',  
  lastName: 'Green',  
  age: 22,  
  location: {  
    lat: 42.698,  
    lng: 23.322  
  }  
}  
  
console.log(student);  
console.log(student.location.lat);
```

Създаване на обект в JavaScript

```
var name = "Sofia";  
var population = 1325744;  
var country = "Bulgaria";  
var town = { name, population, country };
```

```
console.log(town);  
// Object {name: "Sofia", population: 1325744, country: "Bulgaria"}
```

Комбиниране на данни и функции в обект

```
var rect = {  
  width: 10,  
  height: 4,  
  grow: function(w, h) {  
    this.width += w;  
    this.height += h;  
  },  
  print: function() {  
    console.log('[' + this.width + ' x ' + this.height + ']');  
  }  
};
```

```
rect.grow(2, 3);  
rect.print();  
// [12 x 7]
```


Какво представлява затварянето на функция /Closure/

JavaScript Closure представлява затворено място във функция, което скрива нейните променливи и функции извън нея.



Какво представлява затварнето на функция /Closure/

```
function counterClosure() {  
    var counter = 0;  
    function getNextCount() {  
        console.log(++counter);  
    };  
    return getNextCount;  
}
```

```
var count = counterClosure();  
count(); // 1  
count(); // 2  
count(); // 3
```


Какво представлява затварнето на функция /Closure/

```
var counter = (function() {  
    var num = 0;  
    return function() {  
        console.log(++num);  
    };  
})();
```

```
counter(); // 1  
counter(); // 2  
counter(); // 3
```

“Модулен” (Module) шаблон използвайки Closure

```
var module = (function() {  
    var count = 0; // private  
    return {  
        increase: function (num) { count += num; },  
        decrease: function(num) { count -= num; },  
        value: function() { return count; }  
    };  
})();
```

```
console.log(module.value()); // 0  
console.log(module.increase(5)); // 5  
console.log(module.decrease(2)); // 3  
console.log(module.count); // undefined (not accessible)
```

“Разкриващ модул” (Revealing Module) шаблон използвайки Closure

```
var revModule = (function() {  
    var count = 0; // private  
    function change(amount) {return count += amount; }  
    function increase(num) { return change(num); }  
    function decrease(num) { return change(-num); }  
    function value() { return count; }  
    return {  
        increase,  
        decrease,  
        value  
    };  
})();
```

```
console.log(revModule.value()); // 0  
console.log(revModule.increase(5)); // 5  
console.log(revModule.decrease(2)); // 3  
console.log(module.count); // undefined (not accessible)
```

Въпроси?

