Prototype

Samuil Gospodinov

Prototype

Every object is build by a constructor function.

Prototype

Every constructor call creates a brand new object.

Prototype

A constructor creates an object **based on** it's own prototype.

Prototype

A constructor creates an object that is **linked to** it’s own prototype.

Prototype example 4

```
1  function Chovek(ime) {
2      this.ime = ime;
3  }
4  Chovek.prototype.koisSumAz = function() {
5      return "Az sum " + this.ime;
6  };
7
8  var pesho = new Chovek('Pesho');
9  var ivan = new Chovek('Ivan');
10
11 ivan.govori = function () {
12     console.log("Privet, " + this.koisSumAz + '.');
13 };
14
15 pesho.constructor === Chovek;
16 pesho.constructor === ivan.constructor;
17 pesho.__proto__ === Chovek.prototype;
18 pesho.__proto__ === ivan.__proto__;
```

Prototype example 5

```
1  function Chovek(ime) {
2      this.ime = ime;
3  }
4  Chovek.prototype.koisSumAz = function() {
5      return "Az sum " + this.ime;
6  };
7
8  var pesho = new Chovek('Pesho');
9  var ivan = new Chovek('Ivan');
10
11 ivan.govori = function () {
12     console.log("Privet, " + this.koisSumAz + '.');
13 };
14
15 pesho.__proto__ === Object.getPrototypeOf(pesho);
16 ivan.constructor === Chovek;
17 pesho.__proto__ == ivan.__proto__;
18 ivan.__proto__ == ivan.constructor.prototype;
```

Prototype example 6

```
1  function Chovek(ime) {
2      this.ime = ime;
3  }
4  Chovek.prototype.koisSumAz = function() {
5      return "Az sum " + this.ime;
6  };
7
8  var pesho = new Chovek('Pesho');
9  pesho.koisSumAz(); // 'Az sum Pesho'
10
11 pesho.koisSumAz = function() { // <- Shadowing
12     console.log("Privet, " +
13         Chovek.prototype.koisSumAz.call(this) + '.'
14     );
15 };
16
17 pesho.koisSumAz(); // 'Privet, Az sym Pesho'
```

Prototype example 7 Super Unicorn Magic

```
1  function Chovek(ime) {
2      this.ime = ime;
3  }
4  Chovek.prototype.koisSumAz = function() {
5      return "Az sum " + this.ime;
6  };
7
8  Chovek.prototype.kazvamSe = function() { // <- Shadowing
9      console.log("Privet, " +
10         this.koisSumAz() +
11         '.'
12     );
13 };
14
15 var pesho = new Chovek('Pesho');
16 pesho.koisSumAz(); // 'Privet, Az sym Pesho'
```

Prototype example 8

```
1  function Chovek(ime) {
2      this.ime = ime;
3  }
4  Chovek.prototype.koisSumAz = function() {
5      return "Az sum " + this.ime;
6  };
7
8  function Programist(ime) {
9      Chovek.call(this, ime);
10 }
11 //Programist.prototype = new Chovek(); //ILI...
12 Programist.prototype = Object.create(Chovek.prototype);
13 // .constructor e chupen ve4e, trqbva da se opravi
14
15 Programist.prototype.kazvamSe = function () {
16     console.log('Privet! ' + this.koisSumAz() + '.');
17 };
18
19 var pesho = new Programist('Pesho');
20 var ivan = new Programist('Ivan');
21
22 pesho.kazvamSe(); // 'Privet! Az sum Pesho.'
23 ivan.kazvamSe();  // 'Privet! Az sum Ivan'.
```

Prototype Quiz

1. What is a constructor ?
2. What is .constructor ?
3. What is a [[Prototype]] and where does it comes from?
4. How does a [[Prototype]] affect an object?
5. How do we find out where an object's [[Prototype]] points to?

Reminder example 9

```
1  function Chovek(ime) {
2      this.ime = ime;
3  }
4  Chovek.prototype.koisSumAz = function() {
5      return "Az sum " + this.ime;
6  };
7
8  var pesho = new Chovek('Pesho');
9
10 $('#govori').on('click', pesho.koisSumAz);
```

Good Luck and Happy Coding