

# JavaScript Prototype, JQuery

## Tipovi podataka

- **Primitivni**-brojevi, stringovi, boolean
- **Objekti**
  1. Ugrađeni
  2. Korisnički, definisani
- Kreiranje objekata:
  1. Korišćenjem object literala (**JSON**- *JavaScript Object Notation*)
  2. Korišćenjem ključne reči new
  3. Korišćenjem konstruktora

Prekopirati sledeći kod u primer.js

```
In [1]: var person1={
        firstName:"Mika",
        lastName:"Mikic",
        age:22
      }
      alert("Ime prvog korisnika je "+person1.firstName)

      var person2= new Object();
      person2.firstName="Luka";
      person2.lastName="Lukic";
      person2.age=23;
      alert("Ime drugog korisnika je "+person2.firstName)

      function Person(firstName, lastName, age){
        this.firstName=firstName;
        this.lastName=lastName;
        this.age=age;
      };
      var person3=new Person("Pera", "Peric", 24);
      alert("Ime treceg korisnika je "+person3.firstName)
```

Prekopirati sledeći kod u primer1.html

```
In [2]: <html>
        <body>
          <script src="primer.js"></script>
        </body>
      </html>
```

Prekopirati sledeći kod u primer2.html

```
In [3]: <html>
  <body>
    <script src="primer.js"></script>
    <script type="text/javascript">
      person1.fullName= function(){
        return this.firstName + " "+this.lastName;
      }
      console.log("Ime prvog korisnika je "+person1.firstName);
      console.log("Ime drugog korisnika je "+person2.firstName);
      console.log("Puno ime prvog korisnika je "+person1.fullName());
      console.log("Puno ime drugog korisnika je "+person2.fullName());
    </script>
  </body>
</html>
```

Prekopirati sledeći kod u primer3.html

```
In [4]: <html>
  <body>
    <script type="text/javascript">
      function Person(firstName, lastName, age){
        this.firstName=firstName;
        this.lastName=lastName;
        this.age=age;
        this.fullName= function(){
          return this.firstName+" "+this.lastName;
        }
      };
      var person1=new Person("Pera", "Peric",23);
      var person2=new Person("Mika", "Mikic",24);

      console.log("Ime prvog korisnika je "+person1.firstName);
      console.log("Ime drugog korisnika je "+person2.firstName);
      console.log("Puno ime prvog korisnika je "+person1.fullName());
      console.log("Puno ime drugog korisnika je "+person2.fullName());
    </script>
  </body>
</html>
```

## JS Prototype

JavaScript nema klase u klasičnom smislu te reči, ali ima prototipove kojima se može replicirati većina mogućnosti tradicionalne klase. Prototip sadrži spisak "nasleđenih" svojstava i funkcija koje se dodeljuju svim objektima nastalim od konstruktora koji ima posmatrani prototip. Svaki JS objekat nasleđuje svoje osobine i metode od određenih prototipova, pri čemu je Object.prototype na vrhu hijerarhije. Ako neko svojstvo ne postoji u objektu ide se kroz lanac prototipova i pretražuje se.

Polja koja se kače na **prototype** ponašaju se kao statička polja u Javi - svi objekti jedne klase (prototipa) imaju istu referencu na njih. Svaka osobina ili funkcija definisana u prototipu zauzima manje memorije, nego ona koja je definisana posebno za svaki objekat.

Prekopirati sledeći kod u primer4.html

```
In [5 ]:<html>
  <body>
    <script type="text/javascript">
      function Person(firstName, lastName, age){
        this.firstName=firstName;
        this.lastName=lastName;
        this.age=age;
      };

      Person.prototype.fullName=function(){
        return this.firstName+" "+this.lastName;
      }
      var person1=new Person("Pera", "Peric",23);
      var person2=new Person("Mika", "Mikic",24);

      console.log("Ime prvog korisnika je "+person1.firstName);
      console.log("Ime drugog korisnika je "+person2.firstName);
      console.log("Puno ime prvog korisnika je "+person1.fullName());
      console.log("Puno ime drugog korisnika je "+person2.fullName());

    </script>
  </body>
</html>
```

## Nasleđivanje

Nasleđivanje u JavaScript-u se svodi na dva koraka:

1. Pozivanje konstruktora roditeljskog prototipa(ukoliko je potrebno)
2. Podešavanje prototipa kod naslednika

Prekopirati sledeći kod u primer5.html

In [6]:

```
<html>
  <body>
    <script type="text/javascript">
      function Person(firstName, lastName, age){
        this.firstName=firstName;
        this.lastName=lastName;
        this.age=age;
      };

      Person.prototype.fullName=function(){
        return this.firstName+" "+this.lastName;
      }

      function Student(firstName, lastName, age, studentNum){
        Person.call(this, firstName, lastName, age);
        this.studentNum=studentNum;
      }

      Student.prototype=Object.create(Person.prototype);
      Student.prototype.constructor=Student;

      //Student.prototype.fullName=function(){
      //  return "Student, "+ Person.prototype.fullName.call(this)
      //}

      var person1=new Person("Pera", "Peric",23);
      var person2=new Student("Mika", "Mikic",24, "mm-11-2222");

      console.log("Ime prvog korisnika je "+person1.firstName);
      console.log("Ime drugog korisnika je "+person2.firstName);
      console.log("Broj indeksa drugog korisnika je " + person2.studentNum);
      console.log("Puno ime prvog korisnika je "+person1.fullName());
      console.log("Puno ime drugog korisnika je "+person2.fullName());
    </script>
  </body>
</html>
```

## JQuery

JavaScript biblioteka koja olakšava pisanje klijentskih aplikacija. Omogućava jednostavan pristup elementima HTML stranice i jednostavno izvršavanje akcija nad njima. Uključuje se u HTML stranice pomoću script taga.

- **\$**-aktivira biblioteku jQuery
- **(selector)** - selektuj HTML element, analogno CSS-ovim selektorima
- **akcija()** – izvrši akciju nad selektovanim elementom

Standardni JS kod:

```
document.addEventListener('ready', function() {...});
```

JS korišćenjem jQuery-ja:

```
$(document).ready(function() {...});
```

- o kod u okviru funkcije će se izvršiti samo kada je DOM stablo spremno za JS kod da se izvrši. Ovim se izbegava da se jQuery kod izvršava pre nego što se završi učitavanje stranice

## Selekcija elemenata

Standardni JS kod:

```
let elem=document.getElementById("name");  
let elem=document.querySelector("#name");
```

JS korišćenjem jQuery-ja:

```
let elem= $("name");
```

Elemente selektujemo isto kao u CSS-u. Vraćaju se jQuery wrapper objekti, koji olakšavaju korišćenje JS objekata koji predstavljaju elemente DOM stabla

## Pristup poljima wrapper objekata

Standardni JS kod:

```
let elem=document.getElementById("name");  
alert("Ukucali ste " + elem.value);
```

JS korišćenjem jQuery-ja:

```
let elem=$("#name");  
alert("Ukucali ste " +elem.val());
```

## Obrada događaja

Primer: Želimo da dodamo obrađivač click događaja na div element.

Standardni JS kod:

```
let elem=document.querySelector("div");  
elem.addEventListener("click", function(){  
    let redElem=document.getElementsByClassName("red");  
    redElem.style.display="none";  
});
```

JS korišćenjem jQuery-ja:

```
$("#div").click(function(){  
    $(".red").hide();  
});
```

Prekopirati sledeći kod u registration.html

In [7]:

```
<html>
  <head>
    <meta charset="UTF-8">
    <script src="jquery-3.3.1.js"></script>
    <script src="registration.js"></script>
  </head>
  <body>
    <form action="#" >
      <table>
        <tr>
          <td>Ime:</td>
          <td><input type="text" name="ime"/></td>
        </tr>
        <tr>
          <td>Prezime:</td>
          <td><input type="text" name="prezime"/></td>
        </tr>
        <tr>
          <td>Email:</td>
          <td><input type="email" name="email"/></td>
        </tr>
        <tr>
          <td>Lozinka:</td>
          <td><input type="password" name="lozinka"/></td>
        </tr>
        <tr>
          <td colspan="2" align="center"><input type="submit" value = "registruj"/></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

Prekopirati sledeći kod u registration.js

```
In [8 ]:$(document).ready(function(){

    let forma=$( "form" );
    forma.submit(function(event){
        let inputEl=$( "input[name=ime]" );
        console.log(inputEl.val())
        alert("Submit dogadjaj");
        event.preventDefault();
    });
});
/*
$(document).ready(function(){
    let inputEl=$( "input[name=ime]" );

    $("form").submit(function(event){
        let ime=inputEl.val();
        let allTds=$( 'td' );
        let td=allTds[1];
        if(ime==''){
            td.append("Unesite ime");
            $(allTds[0]).css("color", "red");
        }
        event.preventDefault();
    });
});
*/
```