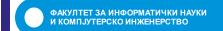
# Аудиториски вежби 6

Интернет програмирање

Иван Китановски Бојан Илијоски Влатко Спасев



## Објекти во JavaScript

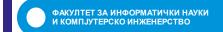
| car.name = Fiat car.start() car.model = 500 car.drive() | Објект | Атрибути           | Методи      |
|---|--------|--------------------|-------------|
| car.model = 500 car.drive()                             |        | car.name = Fiat    | car.start() |
|   |        | car.model = 500    | car.drive() |
| car.weight = 850kg car.brake()                          |        | car.weight = 850kg | car.brake() |
| car.color = white car.stop()                            |        | car.color = white  | car.stop()  |



### Објекти во JavaScript (2)

 Објектите се променливи кои чуваат повеќе вредности

```
let car = {type:"Fiat", model:"500",
color:"white"};
```



### Објекти во JavaScript (3)

■ Пристап до атрибутите на објектот

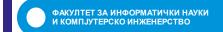
```
objectName.propertyName
или
objectName["propertyName"]
```

Пристап до методите на објектот

```
objectName.methodName()
```

 Доколку се пристапи до методот без (), тогаш се враќа дефиницијата на функцијата

```
let f = objectName.methodName;
```



### Објекти во JavaScript (4)

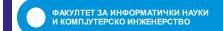
#### Конструктор функција

```
function Person(first, last, age, eyecolor) {
        this.firstName = first;
        this.lastName = last;
        this.age = age;
        this.eyeColor = eyecolor;
        this.name = function() {
            return this.firstName + " " + this.lastName;};
    }
    let person = new Person("John", "Doe", 50, "blue");
Додавање на нов атрибут/метод
person.nationality = "English";
Person.getAge = function() {return this.age;};
```



### Пример 1

```
let myObj = new Object(),
    str = 'myString',
    rand = Math.random(),
   obj = new Object();
myObj.type
                       = 'Dot syntax';
myObj['date created'] = 'String with space';
myObj[str]
            = 'String value';
myObj[rand]
                       = 'Random Number';
myObj[obj]
                       = 'Object';
myObj['']
                       = 'Even an empty string';
console.log(myObj);
//{type: "Dot syntax", date created: "String with space", myString: "String
value", 0.4700474686907987: "Random Number", [object Object]: "Object", ...}
```

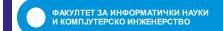


### Пример 2

```
// Animal properties and method encapsulation
let Animal = {
  type: 'Invertebrates', // Default value of properties
  displayType: function() { // Method which will display type of Animal
    console.log(this.type);
};
// Create new animal type called animal1
let animal1 = Object.create(Animal);
animal1.displayType(); // Output:Invertebrates
// Create new animal type called Fishes
let fish = Object.create(Animal);
fish.type = 'Fishes';
fish.displayType(); // Output:Fishes
```

#### Getters and Setters

```
let o = {
 a: 7,
 get b() {
    return this.a + 1;
 },
  set c(x) {
    this.a = x / 2;
console.log(o.a); // 7
console.log(o.b); // 8
0.c = 50;
console.log(o.a); // 25
```

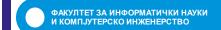


### Deleting properties

```
// Creates a new object, myobj, with two
properties, a and b.
let myobj = new Object;
myobj.a = 5;
myobj.b = 12;
// Removes the a property, leaving myobj with
only the b property.
delete myobj.a;
console.log ('a' in myobj); // yields "false"
```

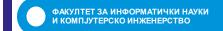
### Comparing Objects

```
// Two variables, two distinct objects with the same properties
let fruit = {name: 'apple'};
let fruitbear = {name: 'apple'};
fruit == fruitbear; // return false
fruit === fruitbear; // return false
// Two variables, a single object
let fruit = {name: 'apple'};
let fruitbear = fruit; // assign fruit object reference to fruitbear
// here fruit and fruitbear are pointing to same object
fruit == fruitbear; // return true
fruit === fruitbear; // return true
fruit.name = 'grape';
console.log(fruitbear); // yields { name: "grape" } instead of { name: "apple" }
```



### Потсетување

```
let num = 0;
let obj = new String('0');
let str = '0':
console.log(num == num); // true
console.log(obj == obj); // true
console.log(str == str); // true
console.log(num == obj); // true
console.log(num == str); // true
console.log(obj == str); // true
console.log(null == undefined); // true
// both false, except in rare cases
console.log(obj == null);
console.log(obj == undefined);
```



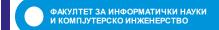
### Креирање објекти - Object Literals

```
// This is an empty object initialized using the object literal
notation
let myBooks = {};
// This is an object with 4 items, again using object literal
let mango = {
       color: "yellow",
       shape: "round",
       sweetness: 8,
       howSweetAmI: function () {
               console.log("Hmm Hmm Good");
       }
```



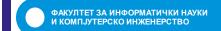
# Креирање објекти - Object Constructor

```
let mango = new Object ();
mango.color = "yellow";
mango.shape= "round";
mango.sweetness = 8;
mango.howSweetAmI = function () {
    console.log("Hmm Hmm Good");
```



#### Креирање објекти - Constructor Pattern

```
function Fruit (theColor, theSweetness, theFruitName, theNativeToLand) {
    this.color = theColor;
    this.sweetness = theSweetness;
    this.fruitName = theFruitName;
    this.nativeToLand = theNativeToLand;
    this.showName = function () {
        console.log("This is a " + this.fruitName);
    }
    this.nativeTo = function () {
        this.nativeToLand.forEach(function (eachCountry) {
                console.log("Grown in:" + eachCountry);
        });
```



#### Креирање објекти - Prototype Pattern

```
function Fruit () {
Fruit.prototype.color = "Yellow";
Fruit.prototype.sweetness = 7;
Fruit.prototype.fruitName = "Generic Fruit";
Fruit.prototype.nativeToLand = "USA";
Fruit.prototype.showName = function () {
        console.log("This is a " + this.fruitName);
Fruit.prototype.nativeTo = function () {
            console.log("Grown in:" + this.nativeToLand);
```

#### Печатење на полиња

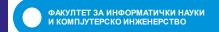


#### Печатење на својства

```
function HigherLearning () {
        this.educationLevel = "University";
  Implement inheritance with the HigherLearning constructor
let school = new HigherLearning ();
school.schoolName = "MIT";
school.schoolAccredited = true;
school.schoolLocation = "Massachusetts";
//Use of the for/in loop to access the properties in the school object
for (let eachItem in school) {
        console.log(eachItem); // Prints educationLevel, schoolName,
                               //schoolAccredited, and schoolLocation
```

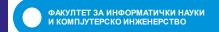
### Computed properties

```
let fruit = prompt("Which fruit to buy?", "apple");
let bag = {
  [fruit]: 5, // the name of the property is taken from
the variable fruit
};
alert( bag.apple ); // 5 if fruit="apple"
//same with
let bag = {};
// take property name from the fruit variable
bag[fruit] = 5;
```



 Напишете JavaScript програма која ќе ги прикаже сите атрибути на даден објект

```
function _keys(obj) {
    if (!isObject(obj)) return [];
    // return Object.keys(obj);
    let keys = [];
    for (let key in obj) keys.push(key);
    return keys;
}
function isObject(obj) {
    let type = typeof obj;
    return type === 'function' || type === 'object' && !!obj;
}
console.log(_keys({red: "#FF0000", green: "#00FF00", white: "#FFFFFF"}));
```



 Напишете JavaScript програма која ќе го избрише атрибутот rollno од следниот објект

```
let student = {
    name : "David Rayy",
    sclass : "VI",
    rollno : 12
};
```



```
let student = {
  name : "David Rayy",
  sclass : "VI",
  rollno : 12
};

console.log(student);

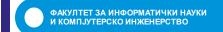
delete student.rollno;

console.log(student);
```

 Напишете JavaScript програма која во конзола ќе ги испечати податоците за следните објекти (книга, автор и статус)

```
let library = [
       author: 'Bill Gates',
       title: 'The Road Ahead',
       readingStatus: true
   },
       author: 'Steve Jobs',
       title: 'Walter Isaacson',
       readingStatus: true
   },
       author: 'Suzanne Collins',
       title: 'Mockingjay: The Final Book of The Hunger Games',
       readingStatus: false
   }];
```

```
let library = [
    {
        title: 'Bill Gates',
        author: 'The Road Ahead',
        readingStatus: true
    },
        title: 'Steve Jobs',
        author: 'Walter Isaacson',
        readingStatus: true
    },
        title: 'Mockingjay: The Final Book of The Hunger Games',
        author: 'Suzanne Collins',
        readingStatus: false
}];
for (let i = 0; i < library.length; i++) {</pre>
    let book = "'" + library[i].title + "'" + ' by ' + library[i].author + ".";
    if (library[i].readingStatus) {
      console.log("Already read " + book);
    } else {
     console.log("You still need to read " + book);
```



 Напишете JavaScript функција која ќе трансформира објект во листа од `[key,value]` парови

```
Влез:
{red: "#FF0000", green: "#00FF00", white: "#FFFFFF"}
Излез:
[["red", "#FF0000"], ["green", "#00FF00"], ["white", "#FFFFFF"]]
```

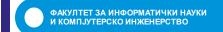
```
function key value pairs(obj) {
    let keys = keys(obj);
    let length = keys.length;
    let pairs = Array(length);
    for (let i = 0; i < length; i++) {</pre>
      pairs[i] = [keys[i], obj[keys[i]]];
    return pairs;
function keys(obj) {
    if (!isObject(obj)) return [];
   // return Object.keys(obj);
    let keys = [];
    for (let key in obj) keys.push(key);
    return keys;
function isObject(obj) {
    let type = typeof obj;
    return type === 'function' || type === 'object' && !!obj;
console.log(key_value_pairs({red: "#FF0000", green: "#00FF00", white: "#FFFFFF"}));
```



 Напишете JavaScript програма која ќе пресметува периметар и плоштина на круг.

Забелешка: Креирајте два методи за периметар и плоштина.
 Радиусот ќе биде даден како влез од корисникот

```
function circle(radius) {
    this.radius = radius;
    this.area = function () {
        return Math.PI * this.radius * this.radius;
    };
    this.perimeter = function () {
        return 2 * Math.PI * this.radius;
    };
let c = new circle(3);
console.log('Area =', c.area().toFixed(2));
console.log('perimeter =', c.perimeter().toFixed(2));
```



Напишете JavaScript програма која ќе прикаже работа на часовник.

#### Излез

"14:37:42"

"14:37:43"

"14:37:44"

"14:37:45"

"14:37:46"

"14:37:47"

```
function my Clock() {
  this.cur date = new Date();
 this.hours = this.cur date.getHours();
  this.minutes = this.cur date.getMinutes();
  this.seconds = this.cur date.getSeconds();
my Clock.prototype.run = function () {
  setInterval(this.update.bind(this), 1000);
};
my Clock.prototype.update = function () {
  this.updateTime(1);
  console.log(this.hours + ":" + this.minutes + ":" + this.seconds);
};
my Clock.prototype.updateTime = function (secs) {
 this.seconds+= secs;
  if (this.seconds >= 60) {
     this.minutes++;
     this.seconds= 0;
  if (this.minutes >= 60) {
     this.hours++;
     this.minutes=0;
  if (this.hours >= 24) {
     this.hours = 0;
};
let clock = new my_Clock();
clock.run();
```